

https://dspace.org/

An application of exponential smoothing methods to weather related data

Marera, Double-Hugh Sid-vicious

2016
http://hdl.handle.net/10539/21029

Downloaded from a DSpace repository. https://dspace.org/

An Application of Exponential Smoothing Methods to Weather Related Data

Author:
Double-Hugh Marera

Supervisor:
Prof. Frank Beichelt



UNIVERSITY OF THE
WITWATERSRAND,
JOHANNESBURG

A Research Report submitted to the Faculty of Science in partial fulfilment of the requirements for the degree of Master of Science in the School of Statistics and Actuarial Science.

26 May 2016

Declaration

I, Double-Hugh Marera, declare that this Research Report is my own, unaided work. It is being submitted for the Degree of Master of Science at the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination at any other University.



Signature:

26 May 2016

Date:

Acknowledgements

By far the greatest thanks must go to my supervisor Prof Frank Beichelt for the guidance, inspiration and support he provided. I would like to thank all the memorable lecturers who taught me at Wits (Prof Beichelt, Prof Kass, Prof Lubinsky, Prof Galpin and Mr Dowdeswell), without them, i would not have learnt much statistical theory and practice.

Thanks must also go to staff of the South African Weather Service. In particular, to Ms Elsa de Jager for providing the datasets used in the report and Dr Andries Kruger for providing valuable input to the research proposal.

Thanks should also go to postgraduate students in the School of Statistics and Actuarial Science who provided stimulating discussions in the masters lab. It would have been a lonely masters lab without them. In particular, i would like to thank Mr Penda Unandapo for his assistance with LaTeX.

Although I have not lived with them for quite a number of years, my parents deserve many thanks for their encouragement and for allowing me to pursue my interests.

I would like to give thanks to my employer Joint Education Trust (JET) and in particular the Data Unit Manager, Ms Jennifer Shindler, for support and for giving me flexible study leave.

Finally, i would like to thank my partner Ms Rani Mokere for the patience, understanding, encouragement and love.

Abstract

Exponential smoothing is a recursive time series technique whereby forecasts are updated for each new incoming data values. The technique has been widely used in forecasting, particularly in business and inventory modelling. Up until the early 2000*s*, exponential smoothing methods were often criticized by statisticians for lacking an objective statistical basis for model selection and modelling errors. Despite this, exponential smoothing methods appealed to forecasters due to their forecasting performance and relative ease of use. In this research report, we apply three commonly used exponential smoothing methods to two datasets which exhibit both trend and seasonality. We apply the method directly on the data without de-seasonalizing the data first. We also apply a seasonal naive method for benchmarking the performance of exponential smoothing methods. We compare both *in-sample* and *out-of-sample* forecasting performance of the methods. The performance of the methods is assessed using forecast accuracy measures. Results show that the Holt-Winters exponential smoothing method with additive seasonality performed best for forecasting monthly rainfall data. The simple exponential smoothing method outperformed the Holt's and Holt-Winters methods for forecasting daily temperature data.

Table of Contents

Declaration i

Acknowledgements ii

Abstract iii

1 Introduction 1

1.1 Historical Overview . . . 1
1.2 Motivation . . . 3
1.3 Aims and Objectives . . . 4
1.3.1 Aim . . . 5
1.3.2 Objectives . . . 5
1.4 Organization of the Report . . . 5

2 Theoretical Background 6

2.1 Introduction . . . 6
2.2 Stochastic Processes . . . 6
2.2.1 Random Variable . . . 7
2.2.2 Stochastic Process . . . 7
2.3 Characteristics of Stochastic Processes . . . 8
2.3.1 Sample Path . . . 8
2.3.2 Trend Function . . . 9
2.3.3 Covariance Function . . . 9
2.3.4 Correlation Function . . . 10
2.4 Classification of Stochastic Processes . . . 11
2.4.1 Stationary Processes . . . 11
2.4.2 Homogeneity and Independent Increments . . . 12
2.4.3 Gaussian Processes . . . 13
2.4.4 Markov Processes . . . 14

2.5 Univariate Time Series . . . 14
2.5.1 Time Series as a Stochastic Process . . . 14
2.5.2 Wold Decomposition Theorem . . . 16
2.5.3 Classical Decomposition . . . 17
2.5.4 Types of Smoothing . . . 19
2.6 Models for Discrete Time Series . . . 21
2.6.1 Purely Random Process . . . 21
2.6.2 Random Walk . . . 22
2.6.3 Autoregressive Process (AR) . . . 23
2.6.4 Moving Average Process (MA) . . . 24
2.6.5 Autoregressive Moving Average Process (ARMA) . . . 26
2.6.6 Integrated ARMA Process (ARIMA) . . . 26
2.6.7 General Linear Process . . . 27
2.6.8 Box-Jenkins Forecasting with ARIMA . . . 27
2.7 Forecasting in Time Series . . . 28
2.7.1 Portmanteau Tests . . . 28
2.7.2 Cross Validation . . . 30
2.7.3 Forecast Accuracy Measures . . . 31
2.7.4 Prediction Intervals . . . 33

3 Exponential Smoothing Methods (ETS) 35
3.1 Introduction . . . 35
3.2 Single Exponential Smoothing . . . 36
3.3 Double Exponential Smoothing . . . 37
3.4 Triple Exponential Smoothing . . . 38
3.5 General Exponential Smoothing . . . 40
3.6 Classification of Exponential Smoothing Methods . . . 41
3.7 Innovations State Space Models . . . 42
3.7.1 Exponential Smoothing Equivalents in State Space Form . . . 43
3.8 Forecasting with Exponential Methods . . . 44

4 Data and Methods 46
4.1 Introduction . . . 46
4.2 Data Description . . . 46
4.2.1 Rainfall Dataset . . . 46
4.2.2 Temperature Dataset . . . 47
4.2.3 Training and Test Datasets . . . 48
4.3 Methodology Description . . . 49

4.3.1 Methods . . . 49
4.3.2 Model Estimation . . . 49
4.3.3 Model Selection . . . 50
4.4 Statistical Software . . . 51

5 Analysis and Results **52**
5.1 Introduction . . . 52
5.2 Application to Rainfall Data . . . 52
5.2.1 Data Exploration . . . 52
5.2.2 Model Identification and Fitting . . . 56
5.2.3 Model Accuracy . . . 59
5.3 Application to Temperature Data . . . 61
5.3.1 Data Exploration . . . 61
5.3.2 Model Identification and Fitting . . . 65
5.3.3 Model Accuracy . . . 67

6 Discussion and Conclusion **70**
6.1 Introduction . . . 70
6.2 Summary of Findings . . . 70
6.3 Future Research . . . 71

References **73**

A Additional Results **76**

B R Code **81**

List of Figures

2.1 Example of a time series 15
2.2 Purely random process 22
2.3 Random walk process 23

4.1 Datasets Plots 48

5.1 Rainfall Box-and-whisker plot 54
5.2 Rainfall Lag Plot 55
5.3 Rainfall Autocorrelation Plot 56
5.4 Rainfall Classical Decomposition 57
5.5 Temperature Box-and-whisker plot 62
5.6 Temperature Lag Plot 63
5.7 Temperature Autocorrelation Plot 64
5.8 Temperature Classical Decomposition 65

A.1 Rainfall time series plot in multiple panels 76
A.2 Temperature time series plot in multiple panels 77
A.3 Box-and-whisker plot for Temperature, 2014 77
A.4 Average Daily Temperature: 2000-2014 78
A.5 Average Monthly Rainfall: 2000-2014 78
A.6 Residual Plots: Rainfall 79
A.7 Residual Plots: Temperature 79
A.8 Actual versus *in-sample* fit: Rainfall 80
A.9 Actual versus *in-sample* fit: Temperature 80

List of Tables

3.1 Taxonomy of exponential smoothing methods 42

5.1 Descriptive Statistics: Rainfall 53
5.2 Estimated Model Parameters: Rainfall 58
5.3 *In-Sample* Accuracy Measures: Rainfall 60
5.4 *Out-of-Sample* Accuracy Measures: Rainfall 61
5.5 Descriptive Statistics: Temperature 62
5.6 Estimated Model Parameters: Temperature 66
5.7 *In-Sample* Accuracy Measures: Temperature 68
5.8 *Out-of-sample* Accuracy Measures: Temperature 68

Chapter 1

Introduction

1.1 Historical Overview

Exponential smoothing methods (ETS) have been used for time series forecasting and prediction since the 1950s. The methods are well known for their simple formulation and relative ease of use. In previous time series forecasting competitions [30, 29], exponential smoothing methods performed better than well developed and complex statistical methods like the autoregressive integrated moving averages (ARIMA).

Exponential smoothing methods can be grouped into three basic classes: simple or single exponential smoothing, double exponential smoothing and triple exponential smoothing. A common characteristic of the methods is that the stochastic process that generated the observed data is a function of unobserved but deterministic components (e.g. local level, trend and season). The components need to be adjusted over time as the structure of the time series change.

The simple exponential smoothing method was developed independently by Brown [8] and Holt [21]. Winters [44] then extended the Holt exponential methods to

allow for forecasting complex time series including seasonality. Hence, the name Holt-Winters exponential smoothing methods to acknowledge their contribution. The original formulation of exponential smoothing methods was underpinned by *heuristics*. Thus, they did not have a statistical basis for model selection as well as for modelling the distribution of the errors. However, their ease of use meant that businesses quickly adopted them, particularly those in inventory modelling.

While the popularity of exponential smoothing methods increased from 1960 to 1970, the statisticians interest in the methods also grew. This was particularly motivated by the need to establish a statistical foundation for the methods. Two significant works laid the foundation for statistical investigations into exponential smoothing methods. First, Muth [33] investigated the statistical properties of the simple exponential methods and found that the simple exponential smoothing method is the same as a model comprising of random walk and noise (i.e. random walk plus noise model [7]). Therefore, the simple exponential smoothing model could be modelled using a stochastic model. Second, Pegels [35] developed a classification scheme of the exponential smoothing methods by considering variations in their trend and seasonal components. The classification later proved to be important as a guide to research into statistical models underlying all the exponential smoothing methods.

However, Box and Pierce [5] developed a modelling strategy for autoregressive and moving average models (ARIMA [6]). The Box-Jenkins approach quickly emerged as a powerful statistical modelling framework. Most of the research in the 1970*s* period was focused on the ARIMA. Research on exponential smoothing methods carried out in this era was biased towards showing that some of exponential smoothing methods were special cases of the ARIMA models. Thus, ARIMA models may have benefited from this research rather than the exponential smoothing methods.

Several papers in the 1980*s* were vital for the future of exponential smoothing methods. Gardner [16] provided a comprehensive review of exponential smoothing methods since they were formulated in the 1950*s*. In fact, Gardner [16] introduced new variations of exponential smoothing methods called damped exponential methods. The damped exponential smoothing methods were developed to adjust the exponential forecasts downwards in long term forecasting. Snyder [41] demonstrated that the simple exponential smoothing method could be reformulated into a model that has one source of error. This type of reformulation is termed an innovation state space model. His work went largely unnoticed at the time [14].

Ord, Koehler, and Snyder [34] extended the innovations state space models to double and triple exponential smoothing methods building on the earlier work of Snyder [41]. Finally, Hyndman, Koehler, Snyder, and Grose [25] provided a comprehensive innovations state space framework for the exponential models. They were able to show that exponential smoothing methods belong to a class that is larger than that of the ARIMA methods. Gardner [17] reviewed the new developments in the exponential smoothing methods as a way of updating their previous work. The innovations state space framework is a major breakthrough for exponential smoothing methods. They can now be considered on equal footing with the ARIMA methods.

1.2 Motivation

Historically, exponential smoothing forecasting methods were viewed as *heuristic* methods. They lacked a coherent statistical framework for modelling error distribution. Thus, they received little attention from among statisticians. Some even considered them to be special cases within the ARIMA models [33, 31]. This

situation is now different.

The development of an innovations [1] state space framework for exponential smoothing methods is a major reason for this. The significance of innovations state space models is that statistical modelling using exponential smoothing methods can be done on an objective basis. Thus, the methods that were largely ignored by the statistical community can now be tested in various application areas. Weather forecasting is one potential application area of exponential smoothing methods. It is important to assess whether or not exponential smoothing methods can be used to forecast weather related data.

A rather indirect justification for undertaking this research is that of in-filing missing weather related time series. If weather measurements are faulty at particular time points, then imputation methods can be used to fill in the missing observations. A possible imputation approach is back forecasting with an exponential forecasting method. This is, however, not the main interest in this research but is something that could be explored further.

1.3 Aims and Objectives

This research report is primarily concerned with exploring the performance of different methods for smoothing time series. More specifically, the focus is on comparing the performance of the exponential smoothing techniques on historical temperature and rainfall data. Formally, the aim and objectives are as below.

[1] a model with a single source of error

1.3.1 Aim

The aim of the project is to compare the performance of exponential smoothing methods on two weather related data sets that exhibits trend and seasonality.

1.3.2 Objectives

- To present a review of exponential smoothing methods with specific emphasis on the Holt-Winters methods,
- To perform forecasts using the exponential smoothing methods and assess their performance through *out-of-sample* forecasting validation,
- To compare the performance of exponential methods against naive smoothing methods on the two data sets, and
- To comment on the performance of exponential smoothing methods on the two data sets.

1.4 Organization of the Report

The research report is structured as follows. Chapter 2 can be divided into two parts. The first part gives a theoretical background of stochastic processes as the umbrella statistical processes under which the research falls. The second part introduces general time series concepts which are the basic building blocks of the research project. Chapter 3 narrows down time series to focus solely on exponential smoothing forecasting methods. Chapter 4 deals with the data and actual methods used in the analysis. Chapter 5 presents the results of exponential smoothing forecasts. Chapter 6 concludes with a summary and discussion of possible areas for future research.

Chapter 2

Theoretical Background

2.1 Introduction

This chapter discusses the basic theory of stochastic processes and foundational aspects of time series. Section 2.2 introduces the basics of stochastic processes. We present the general case for stochastic processes, of which stochastic processes in *discrete time* are a special case. In Section 2.5, we consider some important concepts in time series. Section 2.6 describes discrete stochastic models for univariate time series. Section 2.7 deals with evaluating the accuracy of time series forecasts.

2.2 Stochastic Processes

Stochastic processes theory is very well developed in literature. In this section we give an overview of stochastic processes highlighting the aspects that are important in time series analysis. The material presented here is based largely on Beichelt [2], Grimmett and Stirzaker [20] and Ross [39]. These three monographs offer a non-measure theoretical approach to stochastic processes. The section begins with

the definition of a random variable and extends it to the notion of the stochastic process. Further, the characteristics of stochastic processes are introduced. The section ends with a discussion on classification of stochastic processes.

2.2.1 Random Variable

A random variable X refers to the possible values of a statistical experiment when the conditions are fixed [2]. As conditions are fixed, any deviation to the value of the random variable is attributable to chance alone. A change in the experimental conditions will affect its outcome. A random variable can be redefined to take into account the case when experimental conditions vary. A random function $X(t)$ is a random variable depending on a parameter t where t is deterministic. This results in a generalization of random functions. The study of random functions is formalized next.

Let $\{X(t),\ t \in \mathbb{T}\}$ denote a random variable X depending on a time parameter t whose values are elements of a set $\mathbb{T}$. The set $\mathbb{T}$ represents the sample space of the time parameter t. The set $\mathbb{T}$ can be either one-dimensional or multidimensional. In addition, assume $\mathbb{Z}$ to be the sample space for $X(t)$ under $\mathbb{T}$.

2.2.2 Stochastic Process

A collection of random variables $\{X(t),\ t \in \mathbb{T}\}$ with parameter space $\mathbb{T}$ and state space $\mathbb{Z}$ is known as a stochastic process [2]. $\{X(t),\ t \in \mathbb{T}\}$ is called a stochastic process in *discrete-time* if the parameter space $\mathbb{T}$ is countable. If parameter space $\mathbb{T}$ is continuous, then $\{X(t),\ t \in \mathbb{T}\}$ refers to a stochastic process in *continuous-time*. Depending on the state space $\mathbb{Z}$, a stochastic process can also be discrete or continuous. Thus, if the state space $\mathbb{Z}$ is countable, then $\{X(t),\ t \in \mathbb{T}\}$ refers to a stochastic process in *discrete-space*. Similarly, if the state space $\mathbb{Z}$ is continuous,

then the stochastic process is called a stochastic process in *continuous-space*. By and large, the set $\mathbb{T}$ and the state space $\mathbb{Z}$ result in four distinct types of stochastic processes. Time series are an example of a continuous-space stochastic process in discrete-time or continuous-space stochastic process in continuous-time. Other examples include Poisson processes which are discrete-space stochastic processes in continuous-time, diffusion processes which is a continuous-space stochastic processes in continuous-time and discrete-time Markov chains which are discrete-space stochastic processes in discrete-time.

2.3 Characteristics of Stochastic Processes

A stochastic process $\{X(t),\ t \in \mathbb{T}\}$ is fully characterized by its finite-dimensional distribution. The finite-dimensional distribution of a stochastic process $\{X(t),\ t \in \mathbb{T}\}$ is the family of all joint probability distributions of $(X(t_1), X(t_2), \ldots, X(t_n))$ and is given by the n-dimensional distribution functions

$$F_{t_1,t_2,\ldots,t_n}(x_1, x_2, \ldots, x_n) = P(X_{t_1} < x_1, X_{t_2} < x_2, \ldots, X_{t_n} < x_n) \tag{2.1}$$

where $n = 1, 2, 3, \ldots$ and $t_1, \ldots, t_n \in \mathbb{T}$. Modelling the dependence between any sequence of random variables $X(t_n)$ in a generalized random experiment depends on this complete statistical or probability characterization. The study of the nature of time dependence between random variables is a subject of research in time series.

2.3.1 Sample Path

A sample path of a stochastic process $\{X(t),\ t \in \mathbb{T}\}$ is any realization $\{x(t),\ t \in \mathbb{T}\}$ of the sequentially ordered random variables $\{X(t),\ t \in \mathbb{T}\}$ observed over all t. A

sample path describes one instance of a dynamic stochastic system. Thus, sample paths model the possible developments of the stochastic system.

2.3.2 Trend Function

The trend function $m(t)$ of a stochastic process $\{X(t),\ t \in \mathbb{T}\}$ refers to the expected value of $X(t)$. It depends on the time parameter, t. In mathematical notation, it is represented as

$$m(t) = E(X(t)),\ t \in \mathbb{T}. \tag{2.2}$$

There are two possible interpretations of $m(t)$. Using theory of probability, the Law of Large Numbers [15] states that the arithmetic mean of many independent observations $x(t)$ at the same time point t would be approximately equal to $m(t)$. The trend function $m(t)$ can also be interpreted as the average of the realization $x(t)$ over time. If the probability density function $f_t(x)$ exist, then it is given by

$$m(t) = \int_{-\infty}^{+\infty} x f_t(x) dx,\ t \in \mathbb{T}. \tag{2.3}$$

2.3.3 Covariance Function

The (auto)covariance function $C(s,t)$ of a stochastic process $\{X(t),\ t \in \mathbb{T}\}$ refers to the covariance between two random variables $X(s)$ and $X(t)$ expressed in terms of s and t. Hence, the covariance function measures the degree of statistical dependence between $X(s)$ and $X(t)$. The covariance function $C(s,t)$ is given by

$$C(s,t) = Cov(X(s)X(t)) = E([X(s) - m(s)][X(t) - m(t)]);\ s,t \in \mathbb{T}. \tag{2.4}$$

Further simplification yields

$$C(s,t) = E(X(s)X(t)) - m(s)m(t);\ s,t \in \mathbb{T}. \tag{2.5}$$

In the case of $t = s$, we have the formula for the *variance* of a stochastic process $\{X(t),\ t \in \mathbb{T}\}$ as given by

$$C(t,t) = Var(X(t)) = E[X(t) - m(t)]^2;\ s,t \in \mathbb{T}. \tag{2.6}$$

2.3.4 Correlation Function

The (auto)correlation function $\rho(s,t)$ of a stochastic process $\{X(t),\ t \in \mathbb{T}\}$ refers to the correlation between two random variables $X(s)$ and $X(t)$ expressed in terms of s and t. It is important in modelling the development of a stochastic process through time. The covariance function $\rho(s,t)$ is given by

$$\rho(s,t) = \frac{Cov(X(s)X(t))}{\sqrt{Var(X(s))}\sqrt{Var(X(t))}} \tag{2.7}$$

The terms autocovariance and autocorrelation refer to the fact that we are dealing with the same stochastic process at different time points. If, however, we are interested in the covariance and correlation between two different stochastic processes, say $\{X(t),\ t \in \mathbb{T}\}$ and $\{Y(t),\ t \in \mathbb{T}\}$, then such functions will be better defined as *cross covariance function* and *cross correlation function.* The formulae remain the same as the above. Autocorrelation and autocovariance functions are very useful in time series model identification.

2.4 Classification of Stochastic Processes

Stochastic processes can be classified by making use of their n-dimensional distributions. However, this requires a simplified probability structure since it is impossible to observe all possible sample paths of the stochastic processes. Beichelt [2] argues that stochastic processes are classified according to their steady-state properties. That is, a stochastic process is characterized when its statistical properties are invariant with time. This is called a stochastic process in stationary or steady-state. Not all stochastic processes have the stationarity property. Thus, stationary stochastic processes can be viewed as a subset of the set of all stochastic processes. In a way, stationarity reduces the class of stochastic processes and allows for inference of the entire stochastic process. Stationarity is formalized next.

2.4.1 Stationary Processes

Strict Stationarity

A stochastic process $\{X(t),\ t \in \mathbb{T}\}$ is defined as strictly or strongly stationary if joint distribution functions of random vectors

$$(X_{t_1}, X_{t_2}, \ldots, X_{t_n}) \text{ and } (X_{t_1+u}, X_{t_2+u}, \ldots, X_{t_n+u}) \tag{2.8}$$

are identical for all t, possible integer n, and $t_1, t_1, \ldots, t_n \in \mathbb{T}$ satisfying the conditions $t_j + u \in \mathbb{T},\ j = 1, \ldots, n$. That is,

$$F_{t_1,t_2,\ldots,t_n}(x_1, x_2, \ldots, x_n) = F_{t_1+u,t_2+u,\ldots,t_n+u}(x_1, x_2, \ldots, x_n). \tag{2.9}$$

This means that a stochastic process is strongly stationary if joint distribution functions of a set of random vectors is not changed with a displacement in the time origin, u. A strongly stationary process is not very useful for practical applications since it is very difficult to verify that a process is strongly stationary. This gives rise to the concept of weakly stationary processes.

Weak Stationarity

A stochastic process $\{X(t),\ t \in \mathbb{T}\}$ is referred to as weakly or wide-sense stationary if it satisfies the following conditions:

- $E|X(t)|^2 < \infty$ for all $t \in \mathbb{T}$
- $m(t) = E(X(t)) = \mu$ for all $t \in \mathbb{T}$ where μ is a constant, and
- $C(s,t) = C(0, t-s)$ for all $s, t \in \mathbb{T}$ where $C(s,t) = Cov(X(s)X(t))$.

In other words, a stochastic process is weakly stationary if it has finite first and second moments and a covariance function that depends only on the interval length (i.e. covariance stationary). The (auto)covariance function for a stationary stochastic process $\{X(t),\ t \in \mathbb{T}\}$ can be written as $C(s,t) = C(0, t-s)$ for all $s, t \in \mathbb{T}$. Therefore, the (auto)covariance function of the process can be expressed in terms of just one variable τ as

$$C(\tau) \equiv C(0,\tau) = Cov(X(t+\tau)X(t)) \text{ for all } t, \tau \in \mathbb{T}. \tag{2.10}$$

2.4.2 Homogeneity and Independent Increments

Homogeneous Increments

A stochastic process $\{X(t),\ t \in \mathbb{T}\}$ is said to have homogeneous increments if, for arbitrary but fixed $t_1, t_2 \in \mathbb{T}$, the increment $X(t_2+\tau) - X(t_1+\tau)$ has the same

probability distribution for all τ with property $t_1 + \tau \in \mathbb{T}$, $t_2 + \tau \in \mathbb{T}$.

Independent Increments

A stochastic process $\{X(t),\ t \in \mathbb{T}\}$ is said to have independent increments if for all $n = 2, 3, \ldots$ and for all $n-$tuples $(t_1, t_2, \ldots, t_n) \in \mathbb{T}$ and $t_1 < t_2 < \cdots < t_n$, the increments

$$X(t_2) - X(t_1), X(t_3) - X(t_2), \ldots, X(t_n) - X(t_{n-1}) \tag{2.11}$$

are independent random variables.

2.4.3 Gaussian Processes

A stochastic process $\{X(t),\ t \in \mathbb{T}\}$ is said to be Gaussian if the random vectors $(X(t_1), X(t_2), \ldots, X(t_n))$ have multivariate normal distributions for all $n-$tuples $(t_1, t_2, \ldots, t_n) \in \mathbb{T}$ and $t_1 < t_2 < \cdots < t_n$; $n = 1, 2, \ldots$. That is, all finite-dimensional distributions of the process are joint Gaussian distributions. If $\boldsymbol{\mu} = (\mu_{t_1}, \mu_{t_2}, \ldots, \mu_{t_n})'$ be the mean vector and $\boldsymbol{\Sigma}$ denotes the $n \times n$ covariance matrix of the process, then the probability density function of the multivariate normal distribution is given by:

$$f(\mathbf{x}) = (2\pi)^{-\frac{n}{2}} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\} \tag{2.12}$$

where $|\mathbb{A}|$ is the determinant of a quadratic matrix $\mathbb{A}$. The covariance matrix must also be positive definite. An important characteristic of a Gaussian process is that its finite-dimensional distribution is fully characterized by its first and second moments. This means that it is characterized by its mean and covariance functions. In this special case, weak stationarity implies strong stationarity. The

reverse is always true if the underlying stochastic process is a second order process.

2.4.4 Markov Processes

A stochastic process $\{X(t),\ t \in \mathbb{T}\}$ in discrete state space is said to have the Markov(ian) property if for all $(n+1)-$tuples $(t_1, t_2, \ldots, t_n + 1) \in \mathbb{T}$, $t_1 < t_2 < \cdots < t_{n+1}$ and j,

$$P\{X(t_{n+1}) = j | X(t_0) = i_0, X(t_1) = i_1, \ldots, X(t_n) = i\} = P\{X(t_{n+1}) = j | X(t_n) = i\}. \tag{2.13}$$

In non-technical terms, the Markov property states that the probability of any future behaviour of a stochastic process, if the current state is exactly known, will not be changed by information concerning the process' past behaviour.

2.5 Univariate Time Series

This section outlines the general context in which our study of time series analysis takes place. We restrict our context to time series in *time domain* only. We begin by defining the elementary concepts associated with time series. Then Section 2.5.3 introduces the notion of classical decomposition, setting the scene for the rest of the report. The section ends by introducing the concept of smoothing in time series. All the concepts introduced in this section can be considered to be primary tools for time series modelling.

2.5.1 Time Series as a Stochastic Process

A **time series** is a set of realized values of a stochastic process $\{X(t),\ t \in \mathbb{T}\}$ indexed by time. Hence, a time series cannot be a stochastic process. It is the

sample path of a stochastic process if you stick to the strict definition above. Dagum [13] notes that, unlike in cross sectional studies, in time series analysis ordered observations are dependent through time. That is, observations are related through their index in time. Dagum [13] also states that the time dependence of observations is of importance in time series. A question that often arises in time series is what constitute a time series: the data or the data generating process? Brockwell and Davis [7] suggest that both the data and data generating process of which the time series is a sample path can be referred to as a time series. Figure 2.1 shows a time series plot of the Rand/Dollar exchange rates over the period 2000 and 2015. The exchange rates fluctuates between $R5$ and $R15$ per \$1. A

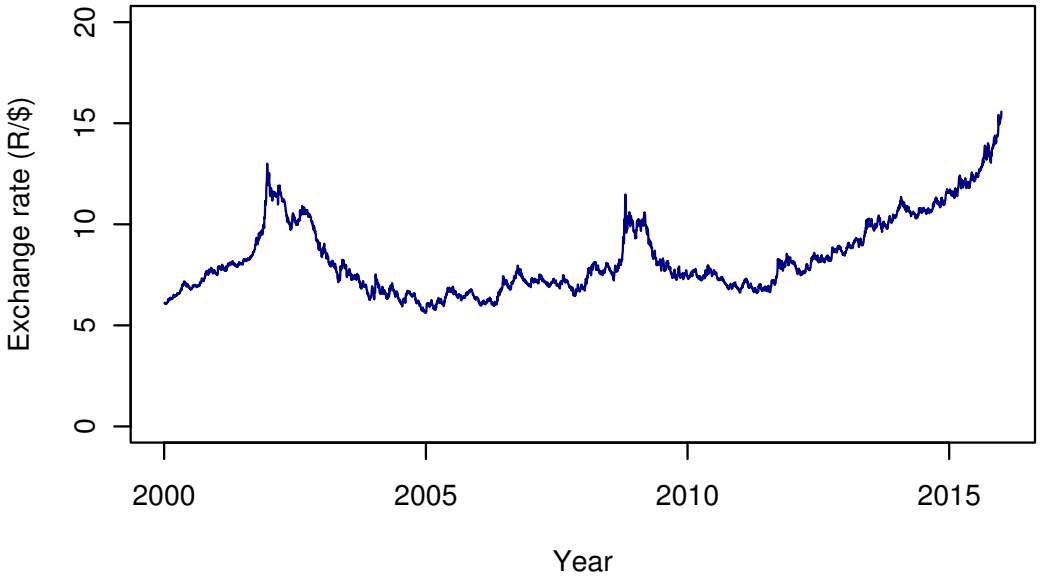


Figure 2.1: Example of a time series

stochastic process approach to time series modelling requires that observed time series data be treated as a finite component of a sample path. Then, the stochastic model serves as the theoretical model that generated the data series. This view of time series allows the use of the theoretical models developed for probability

and stochastic processes. In essence, all the properties of stochastic processes mentioned in Sections 2.2, 2.3 and 2.4 are applicable to time series. In particular, the concept of stationarity is very much central to time series modelling.

2.5.2 Wold Decomposition Theorem

The Wold Decomposition Theorem is an important theorem in time series. Put simply, the theorem states that any discrete stationary stochastic process can be expressed as a superposition of a purely deterministic process and a purely non-deterministic process [7]. A purely deterministic process, such as a function of time, is always uncorrelated to any other genuine stochastic processes. Also, a deterministic process is a general stochastic process. Any stationary process $\{X(t),\ t \in \mathbb{T}\}$ of which μ is the expected value can be decomposed into two uncorrelated components:

$$X(t) = Z(t) + V(t),\ cov(Z(t), V(t)) \equiv 0,\ \forall t \in \mathbb{T} \tag{2.14}$$

where

$$V(t) = \mu + \sum_{j=1}^{\infty}[\alpha_j sin(\lambda_j t) + \beta_j cos(\lambda_j t)],\ 0 < \lambda_j \leq \pi \tag{2.15}$$

$$Z(t) = \sum_{j=0}^{\infty} \Psi_j a_{t-j},\ \Psi_0 = 1,\ \sum_{j=1}^{\infty} \Psi^2 \leq \infty \tag{2.16}$$

where a_t is a white noise process and α_j and β_j are sequences of uncorrelated random variables with mean zero $\forall j$. An application of the Wold Decomposition Theorem is found in the classical decomposition of time series.

2.5.3 Classical Decomposition

Classical decomposition of a time series has its origins in the work of Persons [36] and Persons [37]. His work on developing business indices was among the first to explicitly define the generating processes of a time series. While there has been lots of developments in time series analysis ever since, the classical decomposition method has remained largely unchanged. The method proposes the components of the stochastic process underlying a general time series. It states that any time series $\{X(t),\ t \in \mathbb{T}\}$ can be expressed as a function of its major components. Chatfield [12] lists the components as:

- *Trend Component* $(T(t))$ which accounts for the long term increase or decrease in the mean of the series,
- *Seasonal Component* $(S(t))$ which accounts for cyclical fluctuations related to calendar time (e.g. hour, day, month or quarter),
- *Cycle Component* $(C(t))$ which accounts for other cyclical fluctuations (e.g. business cycles) which may or may not be permanent, and
- *Random Noise Component* $(R(t))$ which measures other random fluctuations and is assumed to be stationary.

In practice, trend and cycle components of a time series can be combined into trend-cycle. Components $T(t)$, $S(t)$ and $C(t)$ are deterministic while $R(t)$ is non-deterministic thereby satisfying the Wold Decomposition Theorem. Brockwell and Davis [7] argue that the aim of classical decomposition is to estimate the deterministic components and if the model is correct the random noise component would be a stationary random process. In mathematical notation, the stochastic process underlying a time series is given by:

$$X(t) = f(T(t), S(t), C(t), R(t)). \tag{2.17}$$

Traditionally, the sources of variations in a time series have been assumed to be deterministic and therefore independent of each other. Hence, the time series can be specified by means of an additive superposition model as:

$$X(t) = T(t) + S(t) + C(t) + R(t) \tag{2.18}$$

where $X(t)$ denotes the observed series at time t, $T(t)$ the trend component at time t, $S(t)$ seasonal component at time t, $C(t)$ is the cycle component at time t and $R(t)$ the random noise or irregular component at time t. The additive decomposition model is appropriate in situations where the amplitude of seasonal component $S(t)$ does not change with changes in time series level.

There are cases when the additive model is not suitable. For example, when there is dependence among the underlying components of the stochastic process. That is, the amplitude of the seasonal component $S(t)$ varies with the level of the time series. In such cases, decomposition is specified through a multiplicative model as follows

$$X(t) = T(t)S(t)C(t)R(t) \tag{2.19}$$

where now $S(t)$ and $R(t)$ are expressed in proportion to the trend $T(t)$ and $C(t)$. A logarithmic transformation makes this model an additive model (i.e. $\log(X(t)) = \log(T(t)) + \log(S(t)) + \log(C(t)) + \log(R(t))$). A third group of decomposition models makes use of a combination of the additive and multiplicative models. An example of such models is the pseudo-additive model given below

$$Y(t) = [T(t) + S(t) + C(t)]R(t) \tag{2.20}$$

where the variables are as mentioned above. Seasonally adjusted time series can be formed by removing the seasonal component $S(t)$ from observed data. This results in a time series $Y^*(t)$ which is a function of trend and random error components

only. The seasonal adjusted time series are $Y^*(t) = Y(t) - S(t)$ and $Y^*(t) = Y(t)/S(t)$ for the additive and multiplicative models respectively.

2.5.4 Types of Smoothing

Smoothing techniques are efficient methods to identify important and non-important fluctuations within an observed time series. The techniques allow for easy identification of the trend component $T(t)$ of the series. Smoothing is an important element of time series forecasting. Smoothing theory presented here is based on Chatfield [12] and Shumway and Stoffer [40]. Smoothing has its origins in the theory of linear systems. A linear filter maps a given time series $\{x_t,\ t \in \mathbb{T}\}$ into a new time series $\{y_t,\ t \in \mathbb{T}\}$ as follows:

$$y_t = \sum_{j=-\infty}^{\infty} w_j x_{t-j},\ t \in \mathbb{T}. \tag{2.21}$$

where w_j are the filter weights assigned to the values x_j of a time series. For a given interval, say $[-a, b]$, the weights satisfy the following normalizing condition:

$$\sum_{j=-a}^{b} w_j = 1 \tag{2.22}$$

The interval $[-a, b]$ determines the bandwidth of the filter.

Moving Average Smoothing

The moving average is a type of a linear filter with the interval $[-a, a]$ with $2a+1$ time points. The weights assigned by a moving average smoother are all the same and equal to

$$w_j = \begin{cases} \frac{1}{2a+1} & \text{for } j = -a, -a+1, \ldots, a-1, a \\ 0 & \text{Otherwise.} \end{cases} \tag{2.23}$$

Kernel Smoothing

Kernel smoothers are sometimes referred to as density estimators. Examples include the Gaussian and Epanechnikov kernels. The Gaussian kernel uses the Gaussian (normal) distribution function as the weights. The weights for the Epanechnikov kernel using the bandwidth $[-a, a]$ are given by

$$w_j = c\left[1 - \frac{j^2}{(a+1)^2}\right] \text{ for } j = 0, \pm 1, \ldots, \pm a. \tag{2.24}$$

Ideally, the wider the bandwidth the smoother the kernel. In order for $\sum_{j=-a}^{a} w_j = 1$ to be satisfied, it follows that the factor c is given by

$$c = \left[\frac{a(4a+5)}{3(a+1)}\right]^{-1} \tag{2.25}$$

Exponential Smoothing

In exponential smoothing, the past and present values of a time series $\{x_t,\ t \in 1 \ldots n\}$ are used to calculate y_k from the observations $x_k, x_{k-1}, \ldots, x_0$ according to the following rule

$$y_k = \lambda l(k) x_k + \lambda(1-\lambda) l(k) x_{k-1} + \cdots + \lambda(1-\lambda)^k l(k) x_0,\ k = 0, 1, \ldots, n. \tag{2.26}$$

where the parameter λ satisfies $0 < \lambda < 1$. The weights under exponential smoothing are given by

$$w_{-j} = \lambda(1-\lambda)^j l(k), \text{ for } j = k, k-1, \ldots, 1, 0. \tag{2.27}$$

The bandwidth is such that $a = a(k) = k+1$ and $b = 0$. In order for $\sum_{j=-a}^{b} w_j = 1$ to be satisfied, it follows that the multiplier $c(k)$ is given by

$$c(k) = \frac{1}{1-(1-\lambda)^{k+1}} \tag{2.28}$$

If the parameter λ is small, this implies a strong smoothing on x_j since observations that are far in time will have a non-negligible effect on smoothing. Exponential smoothing methods for forecasting are dealt with in depth in Chapter 3.

2.6 Models for Discrete Time Series

This section describes stochastic processes *in discrete time* that are useful for time series modelling. A more detailed discussion of the probability models described here can be found in Beichelt and Fatti [3] and Chatfield [12]. Unless stated otherwise, the information presented in this section comes from these two books.

2.6.1 Purely Random Process

A stochastic process in *discrete time* is defined as a *purely random or white noise* process if it is made up of a sequence of independent and identically distributed random variables $\{\epsilon_t\}$. The random variables $\{\epsilon_t\}$ are assumed to be from a normal distribution with mean zero and variance σ_ϵ^2. Thus, a *purely random* process has constant mean and variance. The covariance function of a *purely random* process is given by

$$\gamma(k) = Cov(\epsilon_t, \epsilon_{t+k}) = \begin{cases} \sigma_\epsilon^2 & k = 0 \\ 0 & k = \pm 1, \pm 2, \ldots \end{cases} \tag{2.29}$$

The mean and autocovariance function of a purely random process are independent of time. This means that a *purely random* process is covariance stationary. The

assumption of independence guarantees that the process is also strictly stationary. The *purely random* process is not particularly interesting in itself but can be used to build more complex time series (e.g. autoregressive moving averages). Figure 2.2 below shows an example of a purely random process simulated from a standard normal distribution.

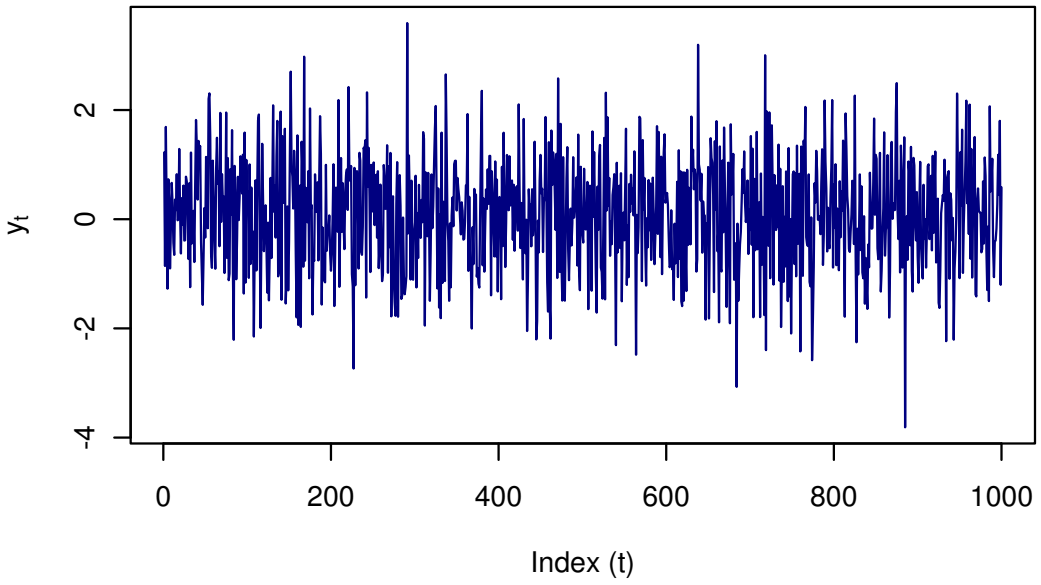


Figure 2.2: Purely random process

2.6.2 Random Walk

Let ϵ_t denote a discrete-time, *purely random* process having mean μ and variance σ_ϵ^2. The process $\{Y_t\}$ is called a *random walk* if

$$Y_t = Y_{t-1} + \epsilon_t. \tag{2.30}$$

By convention, the process starts at zero when $t = 0$ so that $Y_1 = \epsilon_1$ and $Y_t = \sum_{i=1}^{t} \epsilon_i$. The mean, $E(Y_t) = t\mu$, and variance, $Var(Y_t) = t\sigma_\epsilon^2$, increase with time

thus the process is not stationary. A first difference of the *random walk* process, $\bigtriangledown Y_t = \epsilon_t$, results in a *purely random* process and therefore the first difference of a *random walk* is stationary. This insight is important in modelling time series whose behaviour is a *random walk* such as is the movement in share prices in successive time periods. An example of random walk process is shown in Figure 2.3 below.

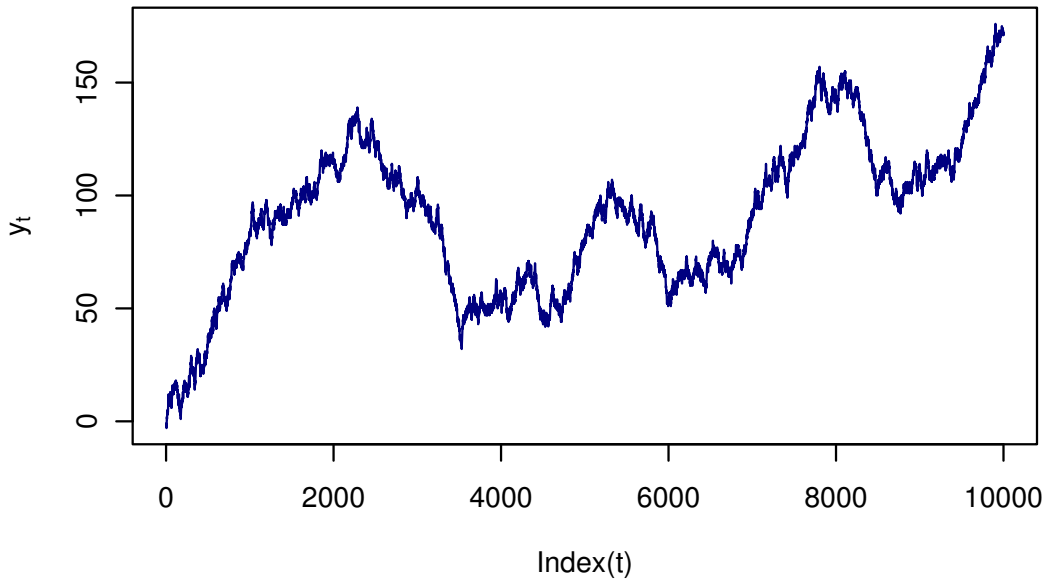


Figure 2.3: Random walk process

2.6.3 Autoregressive Process (AR)

A stochastic process $\{Y_t\}$ is called an *autoregressive* process of order p, denoted by AR(p), if it is defined by

$$Y_t = \sum_{i=1}^{p} \phi_i Y_{t-i} + \epsilon_t \tag{2.31}$$

where p is an integer, $\phi_1, \ldots, \phi_i$ are fixed constants of the autoregressive terms and $\{\epsilon_t\}$ denotes a sequence of independent random variables with mean 0 and variance σ^2. The model is similar to multiple linear regression with the exception that Y_t is regressed on historical values of Y_t rather than on separate predictors. If $p = 1$, the AR(1) process is called a *first-order AR* process and is defined by $Y_t = \phi_1 Y_{t-1} + \epsilon$. When $p > 1$, the process is a *general order AR* process. Upon applying the backward shift operator $BY_t = Y_{t-1}$, $AR(p)$ becomes

$$(1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p) Y_t = \epsilon_t \tag{2.32}$$

which can be written compactly as $\phi(B) Y_t = \epsilon_t$ where

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p \tag{2.33}$$

is called the autoregressive operator and $Y_t = \phi(B)^{-1} \epsilon_t$ is an infinite series. $\{Y_t\}$ is covariance stationary if all the roots of the equation $\phi(B) = 0$ lie outside the unit circle. A necessary condition for stationarity is that $\sum_{i=1}^{p} \phi < 1$. Autoregressive processes are useful for modelling time series whereby the current value is linearly related to its preceding value plus some random noise.

2.6.4 Moving Average Process (MA)

A stochastic process $\{Y_t\}$ is called a *moving average* process of order q, denoted by MA(q), if it satisfies

$$Y_t = \sum_{j=0}^{q} \theta_j \epsilon_{t-j} \tag{2.34}$$

where q is an integer, $\theta_1, \ldots, \theta_q$ are fixed constants of moving average process, $\theta_0 = 1$ and $\{\epsilon_t\}$ is a sequence of independent random variables such that mean is 0 and variance σ^2. If $q = 1$, the MA(1) process is a *first order* process and is defined

by $Y_t = \theta_0\epsilon_t + \theta_1\epsilon_{t-1}$. When $q > 1$, the process is a *general order MA* process. It is clear that $E(Y_t) = 0$ and $Var(Y_t) = \sigma_\epsilon^2 \sum_{i=0}^{q} \beta_i^2$, because $\epsilon_j' s$ are independent. The covariance function of the MA(q) is given by

$$\gamma(k) = Cov(Y_t, Y_{t+k}) = \begin{cases} 0 & k > q \\ \sigma_\epsilon^2 \sum_{i=0}^{q-k} \theta_i\theta_{i+k} & k = 0, 1, \ldots, q \\ \gamma(-k) & k < 0. \end{cases} \tag{2.35}$$

The mean and covariance of MA(q) are independent of t so the process is covariance stationary for all values of θ. If the ϵ_t are normally distributed then the MA(q) will be strictly stationary. Using the backward shift operator,B, the MA equation can be expressed in the form

$$Y_t = \theta(B)\epsilon_t \tag{2.36}$$

where $\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_p B^q$. Unlike the autoregressive processes, there is no restriction on the θ_i for stationarity. However, for the process to be invertible, the roots of the polynomial $\theta(B) = 0$ must lie outside of the unit circle. This means that a moving average process can be expressed in terms of an autoregressive process whose coefficients satisfy $\theta(B) = 0$.

The moving average process MA(q) is a finite stochastic process because the order q is an integer. When the order is unbounded, we have a *moving average* process of unbounded denoted by MA(∞). A stochastic process is an MA(∞) if it can be expressed as $Y_t = \sum_{j=0}^{\infty} \theta_j \epsilon_{t-j}$ and provided that $\sum_{j=0}^{\infty} \theta_j^2$ converges. The importance of an MA(∞) is that any AR(p) can be expressed as MA(∞). Similarly, subject to being an invertible process, an MA(q) can be expressed as AR(∞).

2.6.5 Autoregressive Moving Average Process (ARMA)

A new class of time series models is formed by combining AR and MA processes. An *autoregressive moving average* process, ARMA(p, q) of a time series Y_t is defined by

$$Y_t = \sum_{i=1}^{p} \phi_i Y_{t-i} + \sum_{j=0}^{q} \theta_j \epsilon_{t-j} \tag{2.37}$$

where p is the number of AR terms, q is the number of MA terms, ϕ refers to the coefficients of the AR terms, θ refers to the coefficients of the MA terms and constant and $\{\epsilon_t\}$ is a purely random process. Using the backward shift operator,B, the equation above can be expressed in the form

$$\phi(B)Y_t = \theta(B)\epsilon_t \tag{2.38}$$

where $\phi(B) = 1-\phi_1 B-\phi_2 B^2-\cdots-\phi_p B^p$ and $\theta(B) = 1-\theta_1 B-\theta_2 B^2-\cdots-\theta_p B^q$. The values of $\{\phi_i\}$ and $\{\theta_i\}$ which make the process stationary are the roots of $\phi(B) = 0$ and $\theta(B) = 0$ respectively. All the roots must lie outside of the unit circle. The ARMA(p, q) is very flexible in that stationary time series can be modelled using an ARMA with fewer parameters than a pure AR or MA on its own.

2.6.6 Integrated ARMA Process (ARIMA)

An *autoregressive integrated moving average* process, ARIMA(p, d, q), is an extension of the ARMA process to data that are not stationary. If data are non-stationary, then it is necessary to make it stationary before an ARMA can be applied to the data. One common method of making a time series stationary is by differencing the series, expressed as $\bigtriangledown^d Y_t$. d denotes the number of differences to the original data Y_t required before the series is stationary. When Y_t is replaced by

$\bigtriangledown^d Y_t$ in the ARMA model then the model is referred to as an *integrated* model. The term *integrated* is due to the fact that a stationary model has been fit to differenced data have to be summed to provide a model for the original data (i.e. non-stationary). Integration is a form of summing, hence the term. A time series Y_t is said to be of the ARIMA(p, d, q) if $\bigtriangledown^d Y_t$ is a stationary ARMA process and the parameters as stated previously. The general ARIMA model is of the form

$$\phi(B)(1 - B)^d Y_t = \theta(B)\epsilon_t \tag{2.39}$$

where B, $\phi(B)$ and $\theta(B)$ are as in ARMA. A generalization of the ARIMA model to deal with non-seasonal data is found in Box and Pierce [5]. Such a model is referred to as a SARIMA.

2.6.7 General Linear Process

A general class of linear processes may be written as a moving average process, of possibly infinite order, in the form

$$Y_t = \sum_{i=0}^{\infty} \phi X_{t-i}. \tag{2.40}$$

A sufficient condition for the sum to converge and hence the process to be stationary is that $\sum_{i=0}^{\infty} |\phi| < \infty$. Stationary AR and ARMA processes can also be expressed in terms of the general linear process.

2.6.8 Box-Jenkins Forecasting with ARIMA

The Box-Jenkins procedure [6] is a forecasting strategy based on the autoregressive integrated moving average (ARIMA) models. It emphasizes an iterative process to forecasting model building as follows:

1. *Model identification* Explore the data to select a suitable model from the ARIMA family,
2. *Estimation* Estimate model parameters for the selected model,
3. *Diagnostic checking* Assess residuals from the fitted model to see if it does not violate model assumptions, and
4. *Consideration for alternative models* If the chosen model is not adequate, repeat the above process until a suitable model can be found.

2.7 Forecasting in Time Series

Forecasting refers to the process of estimating the future trajectory of a sequence of observations (i.e. how the time series will continue into the future).

2.7.1 Portmanteau Tests

When fitting a time series model the distribution of residuals is of importance to the modeller. We are interested in finding whether or not the residuals of the estimated model are *white noise*. A plot of the autocorrelation function (ACF) can show if the residuals are uncorrelated, have mean zero and constant variance. However, it is often not sufficient. A *portmanteau test* is a formal test for the independence of residuals up to a lag m. The most commonly used *portmanteau tests* are described below.

Box-Pierce Test

This classical test was proposed by Box and Pierce [5]. The test statistic for Box-Pierce test is given by

$$Q = n \sum_{k=1}^{m} r_k^2 \tag{2.41}$$

where r_k is sample autocorrelation of order k, m is the maximum lag considered and n is the number of observations. Under the null hypothesis that the model is adequate (i.e. residuals are *white noise*), Q follows the χ^2 distribution with $(m - P)$ degrees of freedom. The constant P is a count of model parameters. If the $r_k's$ are very small (i.e. close to zero), then the value of Q will be small. If some $r_k's$ are large, then Q will be large. The Box-Pierce test performs poorly in small samples.

Ljung-Box Test

Ljung and Box [27] developed an alternative test to deal with the perceived shortcomings of the Box-Pierce test. Their test is designed to work well with small samples. The test statistic for the Ljung-Box test is given by

$$Q^* = n(n+2) \sum_{k=1}^{m} \frac{r_k^2}{n-k} \tag{2.42}$$

where r_k is sample autocorrelation of order k, m denotes the maximum lag and n is the total observations. Under the null hypothesis that the residuals are *white noise*, Q^* follows the χ^2 distribution with $(m-P)$ degrees of freedom where P is a count of parameters in the model. If the $r_k's$ are small (i.e. in the neighbourhood of zero), then Q^* will be small. Conversely, if some $r_k's$ are large, then Q^* will be large.

2.7.2 Cross Validation

Measuring how well a statistical algorithm (method) performs on a given dataset is an integral part of statistical modelling. A common approach is to train (fit) an algorithm to a dataset and then use the fitted model to predict the observed values. The extent to which the predicted values deviates from the observed values gives a measure of performance of the method. This kind of approach evaluates the method on the full dataset and may yield downward biased estimates. The *cross validation* approach was developed to deal with perceived weaknesses of evaluating an algorithm on same data it was trained on [26, 32, 42, 18]. In its simplest form, *cross validation* involves splitting a dataset into two sub-datasets. The split ratio depends on the features and size of the dataset, though a ratio of $70:30$ is usually recommended. One of the datasets is used to train the algorithm and the other is used to evaluate its statistical performance. If a split is performed only once, it gives a *validation* estimate while averaging over multiple splits it is known as a *cross validation* estimate. Often an assumption is made that the sub-datasets are independent and identically distributed.

The approach to *cross validation* described above fails when data are dependent, such as is time series data. This is because observations left out during training of the algorithm are necessarily correlated with those used in the training of it. Associated information therefore remained in the evaluation data making it difficult to assess the performance of the algorithm (method). However, Burman, Chow, and Nolan [10] demonstrate that *cross validation* can be extended to the dependent data, where the data form a stationary sequence. They suggest an $h-$block *cross validation* obtained by removing h observations in the neighbourhood of the test observation. Arlot and Celisse [1] provide a comprehensive survey of research into *cross validation* for both independent and dependent data.

For time series forecasting, Hyndman and Athanasopoulos [22] propose the fol-

lowing method to *cross validation*:

- Fit a time series model to the data $Y_1, \ldots, Y_t$,
- Let $\hat{Y}_{t+1}$ be the forecast value of the next time point,
- Calculate the forecast error ($e^*_{t+1} = Y_{t+1} - \hat{Y}_{t+1}$) for the forecast value,
- Redo the process for $t = m, \ldots, n-1$ where m is the smallest number of observations needed to fit the model and
- Calculate forecast accuracy measures (e.g. mean square error) from $e^*_{m+1}, \ldots, e^*_n$.

2.7.3 Forecast Accuracy Measures

The performance of an exponential smoothing method is assessed by considering its forecasting accuracy. There are a number of evaluation criteria which are used to estimate forecast accuracy [24]. Hyndman and Athanasopoulos [22] provide a useful grouping for forecasting accuracy measures. They classify forecasting accuracy measures as scale-dependent, percentage and scaled. We present six accuracy measures used in this report, according to the grouping.

Scale Dependent Measures

Scale dependent accuracy measures are measures which are based on the same scale as the data values. They are, therefore, sensitive to the data values. The forecast error itself is based on the data values and thus is scale dependent. Any accuracy measure that is a function of the forecast error only and not the data values is scale dependent. This means such accuracy measures can not be used for comparing time series measured on different scale. The most common scale dependent measures are defined below. *Mean Error (ME)* is calculated as the

average of the forecast errors over n observations.

$$\text{ME} = \frac{1}{n}\sum_{t=1}^{n} e_t \tag{2.43}$$

Mean Absolute Error (MAE) is calculated as the average of the absolute forecast error values over n data points.

$$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n} |e_t| \tag{2.44}$$

Mean Square Error (MSE) is calculated as the average of the total squared forecast errors over n data points.

$$\text{MSE} = \frac{1}{n}\sum_{t=1}^{n} e_t^2 \tag{2.45}$$

Root Mean Square Error (RMSE) is calculated by taking the square root of MSE (i.e. $\sqrt{(MSE)}$).

Percentage Measures

Percentage accuracy measures are independent of measurement scale. Thus, they are suitable for comparing the performance of time series that are measured on different scales. The common ones include the mean percentage error and mean absolute percentage error. They are undefined on time series with zero values and give extreme estimates with data values in the neighbourhood of zero. *Mean Percentage Error (MPE)* is the average of forecast errors with respect to the actual values over n data points.

$$\text{MPE} = \frac{1}{n}\sum_{t=1}^{n} \frac{e_t}{y_t} \tag{2.46}$$

Mean Absolute Percentage Error (MAPE) is calculated by taking the average of forecast errors with respect to the true values over n data points.

$$\text{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\frac{|e_t|}{y_t} \tag{2.47}$$

Scaled Measures

The scaled measures deal with the weaknesses of the percentage errors [24]. They do not become undefined or produce extreme values. A common scaled measure is the mean absolute scale error defined below. *Mean Absolute Scaled Error (MASE)* is the average absolute error produced by the actual forecast over n data points.

$$\text{MASE} = \frac{1}{n}\sum_{t=1}^{n}\left(\frac{|e_t|}{\frac{1}{n-1}\sum_{i=2}^{n-1}|y_i - y_{i-1}|}\right) \tag{2.48}$$

By and large, all the accuracy measures are based on how close the forecasts are to the eventual (or observed) outcomes. In practical applications, however, the measures are known to give completely different results. As such, evaluating the accuracy of a time series require that one use more than one accuracy measure.

2.7.4 Prediction Intervals

A *prediction interval (PI)* measures the extent of uncertainty around forecast or predicted values. It consists of lower and upper bounds which a forecast or predicted value is expected to be bounded with a specified confidence level. The PI is dependent on the forecasting method used. The $100(1-\alpha)\%$ *prediction interval* for y_{t+h} is given by

$$\hat{y}_t(h) \ \pm \ z_{\alpha/2}\sqrt{Var[\epsilon_t(h)]} \tag{2.49}$$

where $\epsilon_t(h) = y_{t+h} - \hat{y}_t(h)$ is the forecast error made at time t when forecasting h steps ahead, $z_{\alpha/2}$ is a multiplier from the standard normal distribution at $\alpha/2$ and Var is the variance of the errors.

Chapter 3

Exponential Smoothing Methods (ETS)

3.1 Introduction

The main idea behind the exponential smoothing method is to smooth a time series by allocating more importance to recent observations and less importance to observations that are located far in time. It does so by allocating unequal weights to the observations. The largest weight is given to the current observation, less weight to the immediately preceding observation, even less weight to the observation before that, and so on. The weights assigned to time series observations older in time decrease exponentially. By and large, exponential smoothing depends more on recent observations than old observations to forecast. Exponential smoothing methods have three main forms: single exponential smoothing, double exponential smoothing and triple exponential smoothing methods. General exponential smoothing (GES) methods are less well known but are also discussed at the end. Differences between Holt-Winters and GES methods are highlighted.

3.2 Single Exponential Smoothing

This method is commonly known as simple exponential smoothing (SES). The terms are used interchangeably in this report. Simple exponential smoothing is mainly used for short-term forecasting, usually for periods not longer than one month [12]. The method assumes that the time series fluctuates around a stable mean (i.e. the time series is stationary). Thus, there is no trend or seasonality in the time series. The general formula for simple exponential smoothing is:

$$Y_t = \alpha X_t + (1-\alpha)Y_{t-1} \tag{3.1}$$

where Y_t is the smoothed current smoothed value, X_t is the current observation and α is a smoothing parameter such that $0 < \alpha < 1$. In the SES, setting of the initial value of Y_t is critical. A recursive application of the method shows that the weights decay geometrically.

$$\begin{aligned} Y_t &= \alpha X_t + (1-\alpha)Y_{t-1} && (3.2)\\ &= \alpha X_t + (1-\alpha)[\alpha X_{t-1} + (1-\alpha)Y_{t-2}] && (3.3)\\ &= \alpha X_t + \alpha(1-\alpha)X_{t-1} + (1-\alpha)^2 Y_{t-2} && (3.4)\\ &= \alpha X_t + \alpha(1-\alpha)X_{t-1} + \alpha(1-\alpha)^2 X_{t-2} + \cdots + \alpha(1-\alpha)^{t-1}X_1 + (1-\alpha)^t Y_0 && (3.5) \end{aligned}$$

The weights assigned to past observations are proportionate to terms in the geometric series $\{1,\ (1-\alpha),\ (1-\alpha)^2,\ (1-\alpha)^3, \dots\}$. The geometric series is the discrete case of the exponential function, hence the name exponential smoothing. The forecast equation for the SES is given by:

$$F_{t+h|t} = Y_t,\ h = 1, 2, 3, \dots \tag{3.6}$$

where h is the number of forecast steps into future and its associated forecast error at time t is

$$e_{t+h|t} = X_{t+h} - F_{t+h|t} = X_{t+h} - Y_t. \tag{3.7}$$

Simple exponential smoothing smoothing can also be seen as updating the local mean level of a time series, L_t. This simply means replacing Y_t by L_t in the above equations. This notation and interpretation is used in the next method.

3.3 Double Exponential Smoothing

Double exponential smoothing is a generalization of exponential smoothing to time series showing both changing local level and trend. The method is commonly known as Holt's method. It involves exponentially updating (or adjusting) the level and trend of the series at the end of each period. The level (L_t) is estimated by the smoothed data value at the end of each period. The trend (T_t) is estimated by the smoothed average increase at the end of the period. The equations for updating the level and trend of the series are:

$$L_t = \alpha X_t + (1-\alpha)(L_{t-1} + T_{t-1}),\ 0 < \alpha < 1 \tag{3.8}$$

$$T_t = \gamma(L_t - L_{t-1}) + (1-\gamma)T_{t-1},\ 0 < \gamma < 1 \tag{3.9}$$

where α and γ are smoothing parameters for level and trend respectively. The values of α and γ need not be the same. The forecast equation of the Holt's method at time t is given by:

$$F_{t+h} = L_t + hT_t,\ h = 1, 2, 3, \ldots \tag{3.10}$$

where h is the number of forecast steps into future and its associated forecast error at time t is given by

$$e_{t+h|t} = X_{t+h} - F_{t+h} = X_{t+h} - L_t - hT_t. \tag{3.11}$$

3.4 Triple Exponential Smoothing

Triple exponential smoothing extends the double exponential smoothing to model time series with seasonality. The method is also known as the Holt-Winters in recognition of the name of the inventors. Winters improved the Holt's method by adding a third parameter to deal with seasonality. Thus, the method allows for smoothing time series when the level, trend and seasonality can vary. There are two main variations of the triple exponential model and they depend on the type of seasonality. Section 2.5.3 has alluded to the seasonality models as additive or multiplicative model. If the seasonality is multiplicative (i.e. non-linear), then the three smoothing equations pertaining to level, trend and seasonality of p-period cycles are given by:

$$L_t = \alpha(X_t/I_{t-p}) + (1-\alpha)(L_{t-1} + T_{t-1}) \tag{3.12}$$

$$T_t = \gamma(L_t - L_{t-1}) + (1-\gamma)T_{t-1} \tag{3.13}$$

$$I_t = \delta(X_t/L_t) + (1-\delta)I_{t-p}) \tag{3.14}$$

where I_t is seasonality adjusting equation, p is the number of period in seasonal cycle and δ is its smoothing parameter such that $0 < \delta < 1$. The forecast equation h-steps ahead at time t from the Holt-Winters with multiplicative seasonality is given by

$$F_{t+h} = (L_t + hT_t)I_{t-p+h}, \ h = 1, 2, 3, \ldots \tag{3.15}$$

and its associated forecast error at time t is

$$e_{t+h} = X_{t+h} - F_{t+h} = X_{t+h} - (L_t + hT_t)I_{t-p+h}. \tag{3.16}$$

If the seasonality is additive (i.e. linear), the smoothing equations are given by

$$L_t = \alpha(X_t - I_{t-12}) + (1-\alpha)(L_{t-1} + T_{t-1}) \tag{3.17}$$

$$T_t = \gamma(L_t - L_{t-1}) + (1-\gamma)T_{t-1} \tag{3.18}$$

$$I_t = \delta(X_t - L_t) + (1-\delta)I_{t-p} \tag{3.19}$$

where I_t is seasonality adjusting equation and δ is its smoothing parameter such that $0 < \delta < 1$. Hyndman and Athanasopoulos [22] note that the additive model is seldom used in practice. The forecast equation at time t from the Holt-Winters method with additive seasonality is given by

$$F_{t+h} = L_t + hT_t + I_{t-p+h}, \ h = 1, 2, 3, \ldots \tag{3.20}$$

and its associated forecast error at time t is

$$e_{t+h} = X_{t+h} - F_{x+h} = X_{t+h} - L_t - hT_t - I_{t-p+h}. \tag{3.21}$$

All the three main exponential methods are relatively simple to apply, with the SES being the simplest. However, it can be seen as the least realistic for application to real world time series. The exponential methods have been extended to include a "damping" factor which allows for a more conservative forecasting horizon [16, 22]. The exponential smoothing methods are based on recurrence relations. They require initial values of the series of the recurrence equation to be set first. That is, the initial value of Y_t in the SES, the initial values of L_t and T_t in Holt's method and the initial values of L_t, T_t and I_t in the Holt-Winters method. It

would appear that the smaller the values of α, β and γ, the more important is the choice of initial values. The study by Makridakis and Hibon [28] comparing the effect of five choices of starting values (i.e. least squares, backcasting, training set, convenient, and zero starting values) on post-sample forecasting accuracy did not find significant effect caused by the choice of starting values. Recent work by Hyndman et al. [25] use optimization methods to derive starting values.

3.5 General Exponential Smoothing

Brown [9] introduced the idea of general exponential smoothing (GES) or adaptive smoothing. The method uses discounted least squares to fit certain functions of time to a time series. Examples of fitting functions include polynomials, exponentials, sinusoids as well as combinations of their sums or products. In particular, the seasonal component of a time series is modelled by using Fourier functions of time [16]. This use of functions of time is in contrast with the Holt-Winters methods where weights are attached to the data itself. The GES is, perhaps, not as popular as the Holt-Winters methods as a result of mathematical complexity introduced by the use of Fourier functions. The model formulation is described below.

Let $f_i(h)$ be local fitting functions of time. The forecast at time t is a linear combination of locally fitted functions $f_i(h)$ and is given by

$$F_{t+h} = \sum_{i=1}^{n} a_i(t) f_i(h) = \mathbf{a}_t' \mathbf{f}(h) \tag{3.22}$$

where $\mathbf{a}_t$ is a vector of coefficients. $\mathbf{a}_t$ is obtained by minimizing

$$\sum_{j=0}^{t} \beta^j [X_{t-j} - \mathbf{a}_t' \mathbf{f}(-j)]^2. \tag{3.23}$$

For large t, Gardner [16] suggests that the solution to the above equation is of the form,

$$\mathbf{F} = \sum_{j=0}^{\infty} \beta^j \mathbf{f}(-j)\mathbf{f}'(-j). \tag{3.24}$$

Unlike Holt-Winters where the time series components are smoothed, the GES smoothes the model parameters. The solution, in an error correction form, is given by

$$\mathbf{a}_t = \mathbf{L}'\mathbf{a}_{t-1} + \mathbf{h}\mathbf{e}_t \tag{3.25}$$

where $\mathbf{e}$ is an error vector and $\mathbf{L}$ is a square matrix which depends on functions of time only and satisfying $\mathbf{f} = \mathbf{L}\mathbf{f}_{t-1}$ and $\mathbf{h} = \mathbf{F}^{-1}\mathbf{f}(0)$ denotes a smoothing vector dependent on functions of time and parameter β. The smoothing vector is as important as the smoothing parameters in the Holt-Winters methods. The GES is useful only in cases where the data are seasonal. The GES method has some strengths over the Holt-Winters. First, forecast errors for GES have a finite variance owing to the use of a single parameter, β, $0 < \beta < 1$. Second, the GES should adapt to changing seasonal patterns better than Holt-Winters methods since the seasonal terms are adjusted with each data point [16]. The main weakness of the GES over the Holt-Winters methods is that it has a limited scope of application (i.e. where data are seasonal).

3.6 Classification of Exponential Smoothing Methods

The most common exponential smoothing methods described in Section 2.5.4 differ due to trend and seasonality. Pegels [35] developed a classification scheme of the methods by considering variations in the trend and seasonal components. The classification is known as a taxonomy. It has been extended further by Gardner

[16] and Taylor [43] to allow for damped trends in time series forecasting. Table 3.1 presents an comprehensive list of all variants of the exponential smoothing methods. Each cell represent an exponential smoothing methods. The most common ones are the Simple Exponential Smoothing denoted by cell (N,N), Double Exponential Smoothing denoted by cell (A,N) as well as Holt-Winters Methods in cells (A,A) and (AM).

Table 3.1: Taxonomy of exponential smoothing methods

		Seasonal		
		N (None)	A (Additive)	M (Multiplicative)
Trend	N (None)	N,N	N,A	N,M
	A (Additive)	A,N	A,A	A,M
	A_d (Additive damped)	A_d,N	A_d,A	A_d,M
	M (Multiplicative)	M,N	M,A	M,M
	M_d (Multiplicative damped)	M_d,N	M_d,A	M_d,M

3.7 Innovations State Space Models

Statistical models that form the basis for exponential smoothing methods are known as innovations state space models. The innovations models provide a general framework for statistical modelling with exponential smoothing methods. Thus, they allow for an objective model selection. The state space models allow for the estimation of both point forecasts as well as prediction intervals. They can explicitly model the forecast error distribution, which was not possible in the past and had seen them labelled as *ad-hoc*. A detailed exposition of the innovations state space models is found in Ord et al. [34], Hyndman et al. [25] and Hyndman and Athanasopoulos [22].

In the state space models, for each exponential method there exist two equivalent exponential models: one whose error distribution is additive and another whose error distribution is multiplicative. Hyndman and Athanasopoulos [22] note that

there are 30 variations of the exponential smoothing methods in total which correspond to the ones in Table 3.1 with either additive or multiplicative errors. The point forecasts generated by the two models are the same but they do not necessarily generate the same the prediction intervals. This apparent lack of uniqueness make them more suitable for real world applications.

3.7.1 Exponential Smoothing Equivalents in State Space Form

The exponential smoothing methods as described in Sections 3.2, 3.3 and 3.4 can be expressed in their error correction forms. Unless stated otherwise, the formulation presented here is summarized from Hyndman and Athanasopoulos [22]. Simple exponential smoothing (SES) is expressed as local level model governed by the recursive relationship below:

$$y_t = l_{t-1} + \epsilon_t \tag{3.26}$$

$$l_t = l_{t-1} + \alpha\epsilon_t \tag{3.27}$$

where y_t is a smoothed value, α is a smoothing parameter such that $0 \leq \alpha \leq 1$ and ϵ is a random error component. The random error follows the normal distribution and is identically distributed with mean zero and constant variance. Similarly, Double exponential smoothing (Holt's) is formulated as a local trend model with the following recurrence relation:

$$y_t = l_{t-1} + b_{t-1} + \epsilon_t \tag{3.28}$$

$$l_t = l_{t-1} + b_{t-1} + \alpha\epsilon_t \tag{3.29}$$

$$b_t = b_{t-1} + \beta\epsilon_t \tag{3.30}$$

where l_t adjusts local level, b_t adjust for local growth, $0 \leq \alpha \leq 1$, $0 \leq \beta \leq 1$ and ϵ is a random error also assumed to be white noise as with the SES. The Triple exponential smoothing (Holt-Winters) with additive is formulated as below. As seasonality can be additive or multiplicative, we present the additive seasonality model as follows

$$y_t = l_{t-1} + b_{t-1} + s_{t-m} + \epsilon_t \tag{3.31}$$

$$l_t = l_{t-1} + b_{t-1} + \alpha\epsilon_t \tag{3.32}$$

$$b_t = b_{t-1} + \beta\epsilon_t \tag{3.33}$$

$$s_t = s_{t-m} + \gamma\epsilon_t \tag{3.34}$$

where l_t adjusts local level, b_t adjust for local growth, s_t adjusts for seasonality, m is the number of seasons in period (e.g. in a year), $0 \leq \alpha \leq 1$, $0 \leq \beta \leq 1$ and ϵ is a random error assumed to be white noise. The multiplicative seasonal model is not presented here but is very much similar to the additive model. Note that, in all the three models, the equations together with distribution of the error terms result in a fully specified statistical model. Also since the source of randomness in these models is only via the random error term, ϵ, they are referred to as *innovations* models. Hyndman and Athanasopoulos [22] argue that this is to distinguish *innovations* state space models from models that have multiple sources of errors.

3.8 Forecasting with Exponential Methods

The main steps in setting up a Holt-Winters exponential forecasting model according to Chatfield [12] are as follows:

1. Examine the data to select an appropriate model,

2. Initialize starting values for L_1, T_1 and seasonal indices, say $I_1, I_2, \ldots, I_n$ using the first n observations in the series,

3. Estimate values for smoothing parameters α, γ and σ by minimizing the error sum of squares or other suitable measure,

4. Normalize the seasonal indices at regular intervals if appropriate,

5. Estimate model using a fully automatic approach or non-automatic approach, and

6. Estimate forecast accuracy of the estimated model.

Chapter 4

Data and Methods

4.1 Introduction

In this chapter we discuss the data and methods used in this research. Datasets are introduced in Section 4.2. In Section 4.3, we describe the exponential smoothing methods used in the study as well as the seasonal naive method. Section 4.4 deals with the statistical software used in the analysis.

4.2 Data Description

4.2.1 Rainfall Dataset

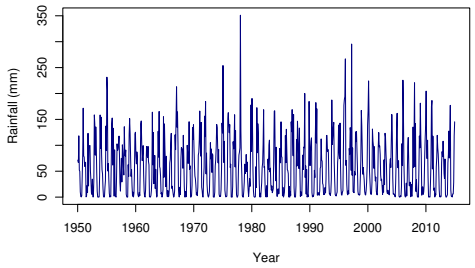
The dataset comprises 780 time series of rainfall measurements taken from the nine provinces of South Africa. The data are monthly averages for each of the nine provinces. The rainfall data are characterized by strong seasonality. The data were collected over the period 1950 to 2014. As rainfall is not very frequent all the seasons in South Africa, daily rainfall time series would have provided data with lots of zero values in some seasons. Thus, average monthly data were considered

to be more useful. The methodology of averaging the data is not provided but has been assumed to be simple averages of daily rainfall. Not all provinces data will be used for the report, only data from Gauteng will be used.

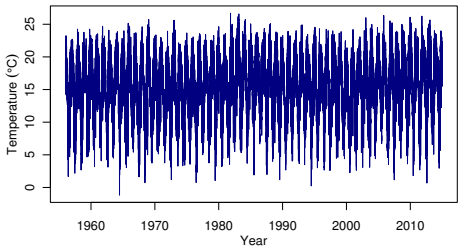
4.2.2 Temperature Dataset

The dataset consists of 21535 empirical time series of temperature recorded at the Johannesburg Weather Station (formerly Jan Smuts) over the period from 1956 to 2014. The South African Weather Service provided the data. The time series is made up of hourly readings from the station. The majority of the time series are positive values, with handful of negative values. The data exhibit clear seasonal pattern along as well as evidence of a trend. The time series are nevertheless characterized by levels of noise. However, this is not uncommon in time series datasets. Less than 1% of the time series are missing data. The missing observations were replaced by values obtained from fitting a local spline function to neighbouring non-missing values. This is to facilitate exponential smoothing forecasting.

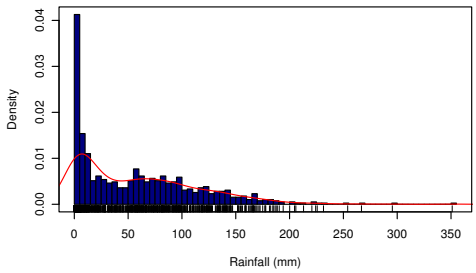
The two datasets contain seasonal time series that are of a univariate nature. No covariates were measured together with the time series. For example, atmospheric pressure and humidity could be important predictors of temperature and humidity. However, the purpose of this research is to compare the performance of the univariate exponential smoothing models. Figure 4.1 below shows time series plots as well as histograms obtained from the two data sets. Figure 4.1(a) describes the average monthly rainfall in Gauteng for the period 1950 to 2014. We see that there is a seasonal pattern to the series with high fluctuations. Most of the rainfall amounts received in Gauteng fall below 100 mm. There is evidence of outlying values with rainfall amounts exceeding 200 mm. Figure 4.1(b) describes the hourly temperature in Johannesburg over the period 1956 to 2014. The time series is characterized by strong seasonality, with temperatures ranging between $5°C$ and



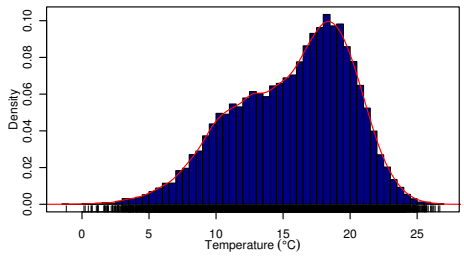
(a) Rainfall Series



(b) Temperature Series



(c) Rainfall Histogram



(d) Temperature Histogram

Figure 4.1: Datasets Plots

$25°C$. Figures 4.1(c) and 4.1(d) depicts the distribution of rain and temperature within each year respectively.

4.2.3 Training and Test Datasets

Both datasets are split into training (*in-sample*) and test (*out-of-sample*) datasets using a ratio of 80 : 20. That is, 80% of the data are used to fit the exponential smoothing models while 20% of the data are used to evaluate the forecasting performance of the models.

4.3 Methodology Description

4.3.1 Methods

Exponential smoothing methods as described in 3.6 are applied to rainfall and temperature data sets. Both data sets have level, stable trend and seasonality. A seasonal naive method is fitted to the same data sets for comparative purposes. The algorithm for the seasonal naive method can be described as follows. The forecast value is the last observed value from the time series at same time and season of the year. For monthly data, that would be last observed data point from the same month of the same season. The forecast for the seasonal naive for time $T + h$ is written as

$$Y_{T+h-km} \tag{4.1}$$

where m = seasonal period, $k = \lfloor (h-1)/m \rfloor + 1$, and $\lfloor u \rfloor$ denotes the integer part of u [22]. An exponential smoothing method that performs worse than this method is not useful for forecasting purposes. The seasonal naive methods therefore provides a benchmark for the performance of exponential smoothing methods.

4.3.2 Model Estimation

Exponential smoothing methods require starting values as well as smoothing parameters, both of which are estimated from the data. The estimation procedure is as follows. First, apply all the models to the data using parameters obtained via maximum likelihood estimation (MLE). Second, select the best model. Details of likelihood theory can be found in Brockwell and Davis [7]. The *forecast* package developed by Hyndman and Khandakar [23] provides automated procedures to estimate the smoothing parameters.

4.3.3 Model Selection

The statistical models that underlie the exponential smoothing methods allow the use of information criteria for model selection. Information criteria measures how well an estimated model fits with data in comparison with other models. According to Hyndman and Athanasopoulos [22], the following information criteria are appropriate for time series data.

The Akaike's Information Criterion (AIC) for time series is defined as

$$\text{AIC} = -2\log(L) + 2k, \tag{4.2}$$

where L is the likelihood function of the model and k is the total number of estimated parameters as well as estimated initial states.

The Akaike's Information Criterion corrected for small sample bias (AICc) is defined as

$$\text{AICc} = \text{AIC} + \frac{2(k+1)(k+2)}{T-k}, \tag{4.3}$$

where T is the number of observations used to estimate the model and k is the total number of estimated parameters in the model and

The Bayesian Information Criterion (BIC) for time series is defined as

$$\text{BIC} = \text{AIC} + k[\log(T) - 2] \tag{4.4}$$

where T is the number of observations used to estimate the model and k is the total number of estimated parameters in the model.

The AIC, AICc and BIC are used here to determine which of the exponential smoothing models is most appropriate for a given time series. In automatic forecasting, the statistical software handles the model selection by comparing various models and choosing the one that minimizes the AIC, AICc and BIC. This is consistent with work by Billah, King, Snyder, and Koehler [4] which argue that

information criteria perform best for model selection purposes compared to other approaches. The accuracy of the selected forecasting model is evaluated using an *out-of-sample* evaluation.

4.4 Statistical Software

The statistical software used in this research project is the R Language [38] and is free to download from `www.R-project.org`. The *forecast* package [23] contains implementations of most of the univariate time series algorithms and modelling frameworks. It includes automatic forecasting for both exponential smoothing and ARIMA methods. The algorithms have been shown to compare favourably to many proprietary software [23]. The base R time series package `ts` also contains the exponential smoothing functions but is limited for complex and large time series analyses.

Chapter 5

Analysis and Results

5.1 Introduction

The purpose of this research is to apply exponential smoothing forecasting methods to two weather related datasets and to assess their forecasting performance using time series *out-of-sample* validation. In addition, we sought to compare the performance of exponential methods against the naive seasonal forecasting method. The chapter is structured as follows. Section 5.2 presents the findings of applying exponential smoothing methods and the seasonal naive method to the monthly rainfall dataset. Section 5.3 presents the results of applying the same methods to the daily temperature dataset.

5.2 Application to Rainfall Data

5.2.1 Data Exploration

Exploratory data analysis (EDA) is crucial to any successful statistical modelling [11]. It involves an in-depth description and assessment of the quality of the data.

Descriptive measures such as summary statistics and graphical methods are used to gain insight into the data. Missing data, outlying and influential data values are checked for to prevent distortions of the data generating process(es). By and large, the initial examination of the data helps in model selection and fitting. Results from the exploratory data analysis follow below.

We begin by looking at the average monthly rainfall recorded over the period 1950 to 2014. Table 5.1 presents summary statistics for the rainfall quantities received over the period. The time series comprises 780 monthly observations. The estimated average monthly rainfall received over the period is 58mm. It ranges over zero (no rain received at all) to 351mm As mentioned in the data description, the distribution of rainfall amounts is skewed.

Table 5.1: Descriptive Statistics: Rainfall

Statistic	N	Mean	St. Dev.	Min	Max
Rainfall	780	58.239	55.705	0.000	351.100

The summary statistics, whilst useful, do not shed much insight into the rainfall pattern over time. We looked at the distribution of rainfall within each year. Figure 5.1 presents a box-and-whisker plot of the rainfall quantities received each year. It shows that the mean rainfall was relatively stable at about 58mm. Most of the rainfall recorded is between 0mm and 100mm. There appear to be a few outlying observations in the rainfall time series whose quantities exceed 150mm. A review of the data and checking with the data providers suggests that the data are indeed correct and it is not a case of faulty recording. The outlying observations are therefore left in the data for analysis as they are a feature of the data rather than an artefact.

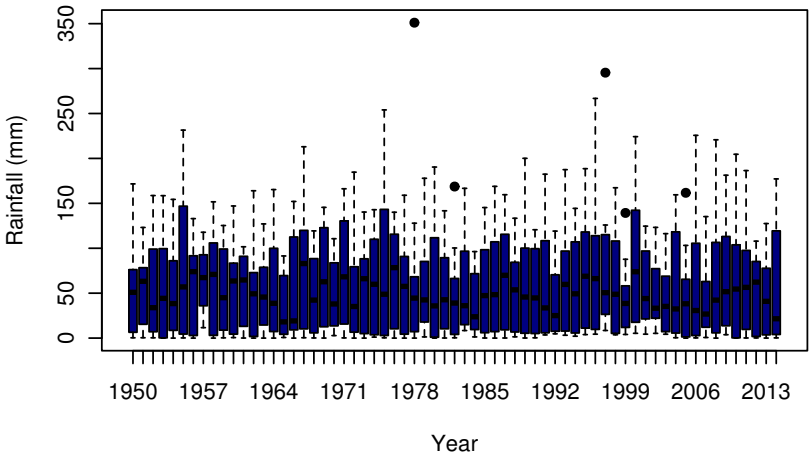


Figure 5.1: Rainfall Box-and-whisker plot

Time series observations are correlated through time. Therefore, it is important to assess the nature of the autocorrelation and indeed check if autocorrelation is present. Figure 5.2 shows a lagged scatter plot (lag plot) with nine lags. In this case, the lag represents the preceding months. If the points lie on the diagonal, this implies a linear relationship between the time series observation and its lag. There is evidence of strong linearity between an observation and its lag (see top left plot). The relationship with observations relatively far in time is that of a structured curvature which suggests a non-linear dependence.

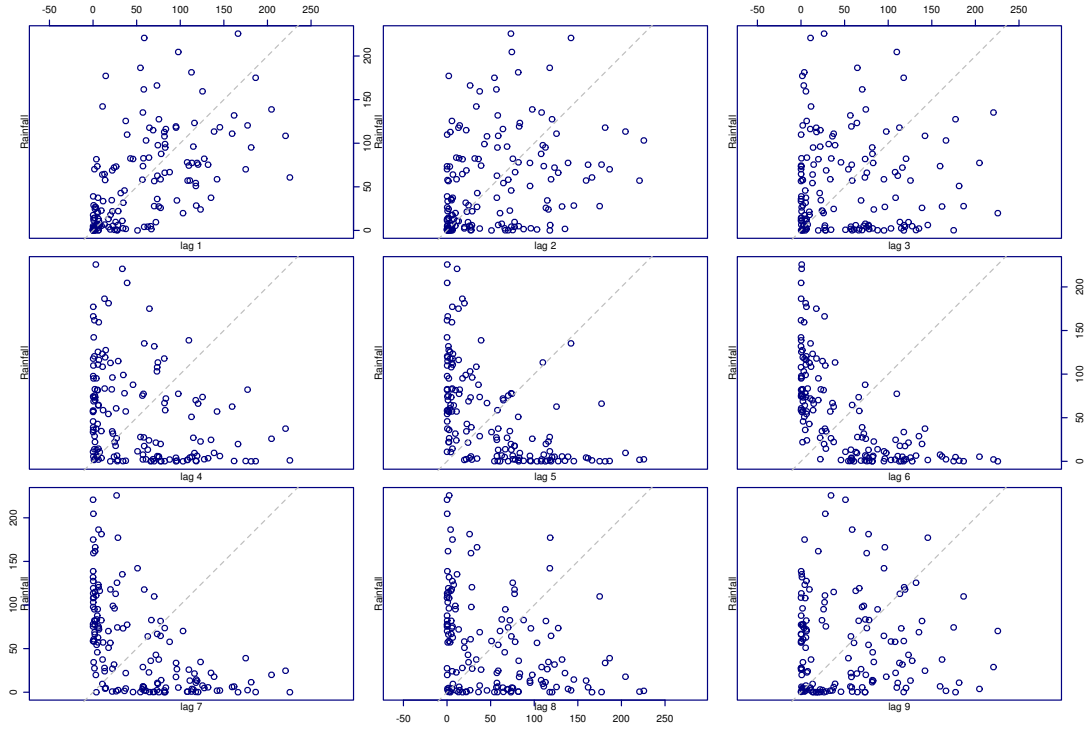


Figure 5.2: Rainfall Lag Plot

The lag plot does not show whether or not the non-linear dependence is seasonal. The autocorrelation (ACF) plot can be used to identify seasonal time series. Figure 5.3 presents the autocorrelation plot of the rainfall time series. The time series is clearly non-linear and the ACF plot shows that the relationship has a strong seasonality. The season has a period of 12 months.

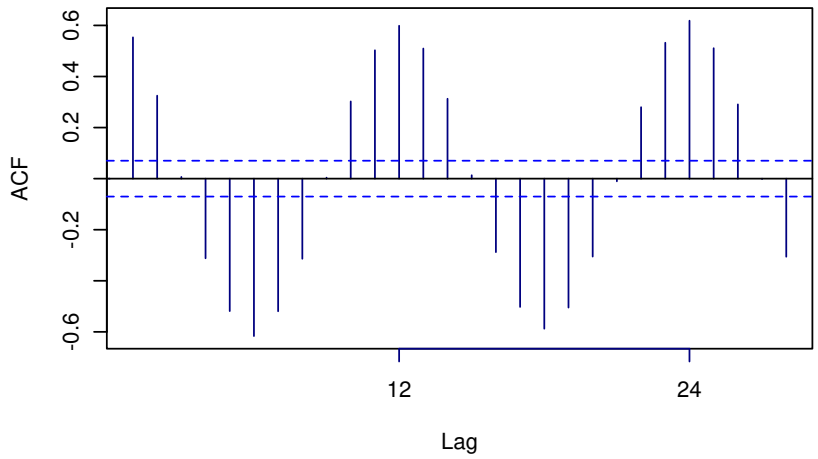


Figure 5.3: Rainfall Autocorrelation Plot

Now that we have more insight into the rainfall time series, we investigate the components of the time series. We make use of classical decomposition of the time series. Figure 5.4 depicts the separate components of the rainfall time series. This suggests that if the nature of the series is additive, the time series can be expressed as an additive function of level, trend and season. In summary, the exploratory data analysis has shown that the average rainfall received in Gauteng can be modelled by a non-linear time series. The time series has a seasonal pattern but the mean is relatively stable at $58mm$. The strength of linearity of observation dependence diminish as the lag increases.

5.2.2 Model Identification and Fitting

The rainfall dataset was divided into two: 80% *in-sample* and 20% *out-of-sample*. As mentioned previously, the *in-sample* dataset was used for model fitting and the *out-of-sample* was used to validate the fitted models. The rainfall observations were displaced upwards with a small constant (i.e. $y_t^* = y_t + 0.00005$) to remove zero values. Accuracy measures such as percentage errors are undefined in the

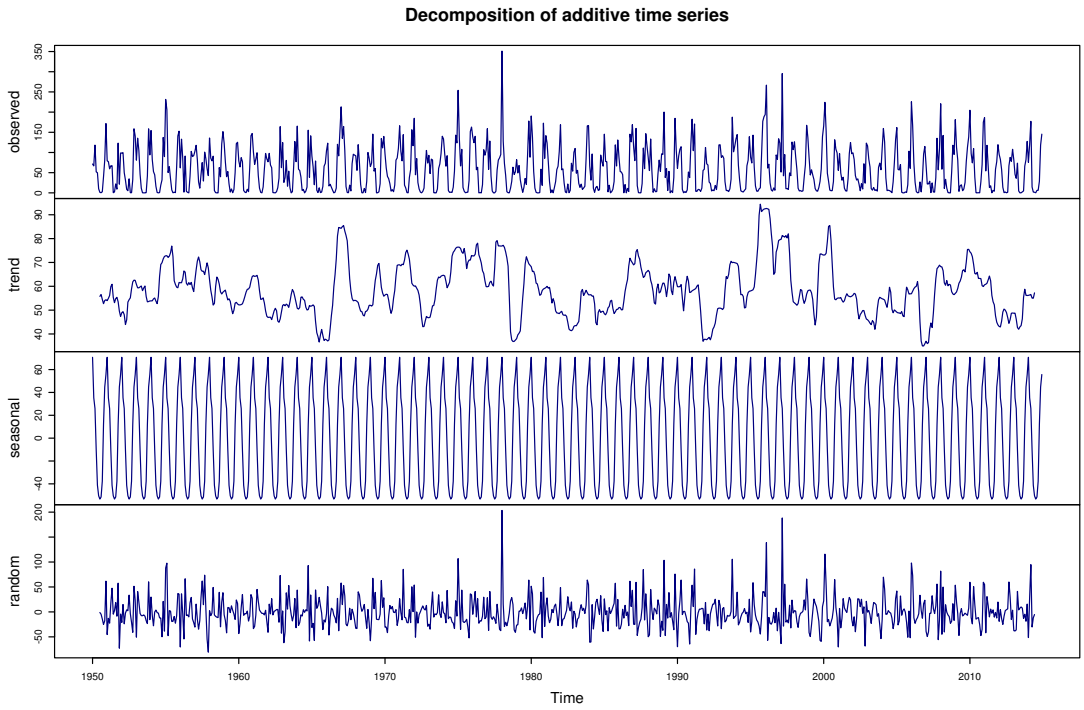


Figure 5.4: Rainfall Classical Decomposition

presence of zero values. The adjustment itself will not affect the analysis all the observations are adjusted by the same constant. It is worth noting that, while the adjustment prevents measures such as MAPE from being undefined, it still yields very large values of MAPE. Thus, caution need to be exercised when interpreting measures sensitive to near zero values. A total of four time series models were fit to the dataset. They include seasonal naive method, simple exponential smoothing (SES), Holt's method and Holt-Winters method. The inclusion of the seasonal naive method is to see how it compares to the exponential smoothing methods. It is standard practice in time series modelling to compare the methods under review to naive methods. It is assumed the other methods will outperform naive methods. Thus, if naive methods outperform the methods being tested, then it means the methods being tested are not worth being used for forecasting or prediction. The process of model fitting uses an automated procedure provided for in the *forecast* package. They use information criteria to select the best model parameters.

Table 5.2: Estimated Model Parameters: Rainfall

Parameter	SES	Holt	H-W Add	H-W Mult
Alpha (α)	0.7689	0.7692	0.0145	0.0319
Beta (β)		0.0001	0.0001	0.0001
Gamma (γ)			0.0001	0.0408

When fitting exponential smoothing models the values of smoothing parameters are of interest. Table 5.2 presents the estimated parameters for the SES, Holt and Holt-Winters methods. SES depends on the smoothing parameter α which adjusts for the local level of the series. The local level α of the rainfall series is estimated to be 0.7689. This is implies that more weight is placed on recent observations than observations far in time in our estimated SES model. A large value of the smoothing parameter α is indicative of a time series with a level that is changing rapidly. This is consistent with monthly rainfall fluctuations as depicted in the time series plots earlier.

Holt's exponential smoothing method is controlled by two parameters: level of the series α and trend of the series β. The estimated value of α is 0.7692 and that of β is 0.0001. The value of α is high telling us that more weight is placed on recent observations than older observations to adjust for the local level of the rainfall series. The value of β is close to zero meaning that the trend of the rainfall time series is changing rather slowly. Relatively little weight is applied to recent observations when forecasting rainfall with estimated Holt's model. It is worth noting that the estimated level parameters between SES and Holt's method are almost the same. Thus, these two models are likely to give similar forecasting accuracy.

Three parameters need to be estimated in the Holt-Winters exponential smoothing method. These are α for the level of the series, β for the trend of the series and γ for the season component. We considered the Holt-Winters method with additive

seasonality and with multiplicative seasonality. First, we present the estimated smoothing parameters for the Holt-Winters method with additive seasonality. The estimated value for the level α of the series is 0.0145, the trend β of the series is 0.0001 and the seasonal component γ is 0.0001. All the smoothing parameters are close to zero. This implies that the fitted model has slowly changing level, trend and season. The level of the rainfall series α is changing relatively faster than the trend and season. The estimated values in this model suggest that relatively little weight is placed on recent observations compared to less recent ones. Second, we now look at the Holt-Winters method with multiplicative seasonality. The estimated parameters are 0.0319, 0.0001 and 0.0408 for α, β and γ respectively. The level parameter α is very low suggesting a low weighting on recent observations than less recent observations. The smoothing values for the trend and seasonal components are also close to zero. This means that the trend and seasonal components are updating slowly and that little weight is placed on recent observations as well as observations far in time.

We also check model assumptions. An assumption underlying the use of state space exponential smoothing methods is that the model residuals are normally and independently distributed with mean zero and constant variance. Histograms of residuals in Figure A.6 suggests that assumption is not violated for all the exponential smoothing models fitted in this research. Both Ljung-Box and Box-Pierce portmanteau tests returned large $p-$values indicating that the model residuals were indeed *white noise*. Thus, the models are a good fit for the rainfall data.

5.2.3 Model Accuracy

We evaluate how well the exponential time series models fit the rainfall data by computing a number of *in-sample* accuracy measures. This is to check how well the model fits the data on which it is modelled. In a way, it gives a sense of

internal consistency of the fitted model. The accuracy measures obtained from fitting the four rainfall time series models are presented in Table 5.3. Across the accuracy measures, the seasonal naive method outperforms both the SES and Holt's methods. The Holt-Winters exponential smoothing methods performed best. In fact, the Holt-Winters model with additive seasonality performed best across the measures for *in-sample* accuracy. It performed better than the Holt-Winters exponential model with multiplicative seasonality.

Table 5.3: *In-Sample* Accuracy Measures: Rainfall

METHOD	ME	RMSE	MAE	MPE	MAPE	MASE
Naive	-9.072	33.221	24.091	-83.730	112.605	0.737
SES	-58.843	66.167	58.843	-411.018	411.018	1.801
Holt	-59.192	66.522	59.192	-413.175	413.175	1.812
H-W Add	-19.920	30.749	22.528	-104.448	114.303	0.689
H-W Mult	-23.270	35.314	26.699	-99.977	114.299	0.817

While *in-sample* evaluation of the models is important, it does not imply that the models will necessarily forecast well. Using the models estimated, we performed an *out-of-sample* evaluation. This evaluation shows which of the models captures well the future evolution of the rainfall time series. Table 5.4 shows the results of the evaluation. The results are similar to the *in-sample* ones with one exception. The Holt-Winters exponential model with multiplicative errors performed very poorly, even worse than the other methods. Still the Holt-Winters method with additive seasonality performed best followed by seasonal naive method. Since the data clearly has seasonality, it is to be expected that the Holt-Winters method (i.e. the model allows for seasonality) performs best. It suggests that ignoring seasonality, when it is presented, does not improve rainfall forecasting ability of a model.

Table 5.4: *Out-of-Sample* Accuracy Measures: Rainfall

METHOD	ME	RMSE	MAE	MPE	MAPE	MASE
Naive	-9.072	33.221	24.091	-83.730	112.605	0.737
SES	-58.843	66.167	58.843	-411.018	411.018	1.801
Holt	-59.192	66.522	59.192	-413.175	413.175	1.812
H-W Add	-19.920	30.749	22.528	-104.448	114.303	0.689
H-W Mult	-23.270	35.314	26.699	-99.977	114.299	0.817

5.3 Application to Temperature Data

5.3.1 Data Exploration

We follow the same approach with temperature as we did with rainfall time series. We use descriptive statistics and exploratory plots to better understand the data. We check for the presence of missing data, outlying and influential data values. The insight gained from the exploratory analysis is useful for model selection and fitting. The temperature dataset comprises daily average temperatures recorded in Johannesburg from 1956 to 2014.

Table 5.5 presents the overall descriptive statistics of the temperature time series dataset. The dataset comprises 21535 observations, which is approximately 365 observations per year. The size of the data means we will able to capture the essential underlying structure of the time series, better than in the rainfall dataset. The temperatures recorded at the weather station ranged between -2 and 27 Degrees Celsius. Thus, the coldest day, on average, had a temperature below zero. Similarly, the warmest day, on average, exceeded 25 degrees Celsius. The mean temperature over the period was 15.8 Degrees Celsius, with a spread of about 4 Degrees Celsius.

Table 5.5: Descriptive Statistics: Temperature

Statistic	N	Mean	St. Dev.	Min	Max
Temperature	21,535	15.670	4.314	−1.171	26.692

We performed a graphical analysis to understand the distribution of time series over this period. Figure 5.5 shows an annual box-and-whisker plot for the temperature time series. The plot suggest a stable distribution of temperatures within the years. As can be seen, the median temperature is approximately 15 Degree Celsius. There is substantial shift in the overall distributions between the year, perhaps only wider distributions in years. Most of temperature lie in a band between 10 and 20 Degrees Celsius. There is evidence of outlying observations. In this case, we consider outliers the observations below 5 Degrees Celsius. It appears that most years have some days which are much colder than normal. Most of the coldest days are Winter days and this is not unexpected. The box-and-whisker plot, on the whole, suggests that the temperatures do not change significantly over time.

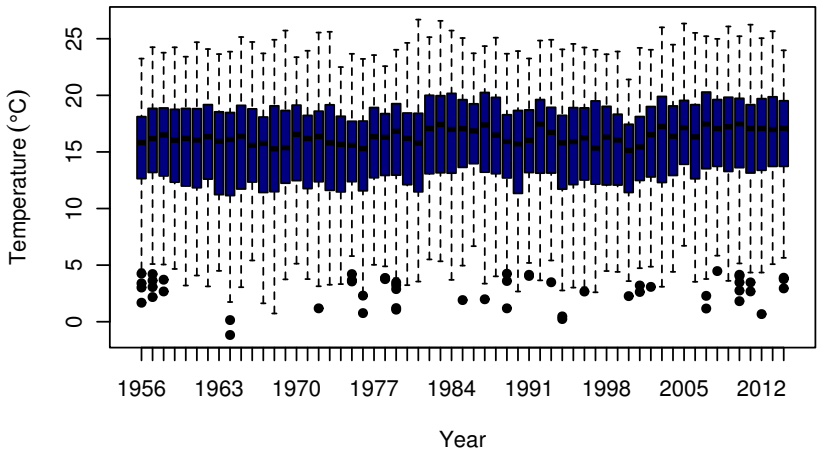


Figure 5.5: Temperature Box-and-whisker plot

Figure 5.6 shows the lagged scatter plot up to nine lags. Here we are looking at

the correlation between the current and preceding observations. The plot shows a very strong linear relationship between the observations up to lag nine. Since this is a high frequency dataset, any non-linearity if present will show in lags far removed from the current observation. Therefore, the figure does not contain a definitive trend. Further autocorrelation checks were performed.

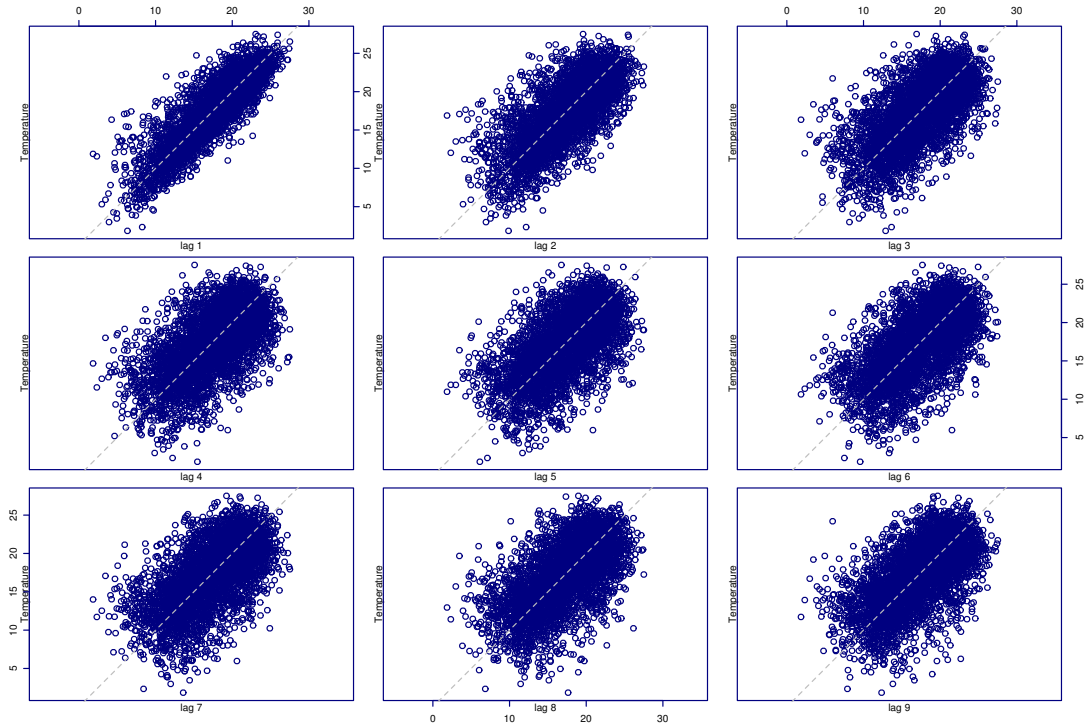


Figure 5.6: Temperature Lag Plot

The autocorrelation plot is able to pick up a non-linear relationship even in high frequency data. Figure 5.7 depicts the relationship between successive observations in the temperature time series. Unlike the lagged scatter plot, the ACF shows strong non-linear relationship as well as evidence of seasonality. This is what we expected. The lagged plot might have been slightly misleading.

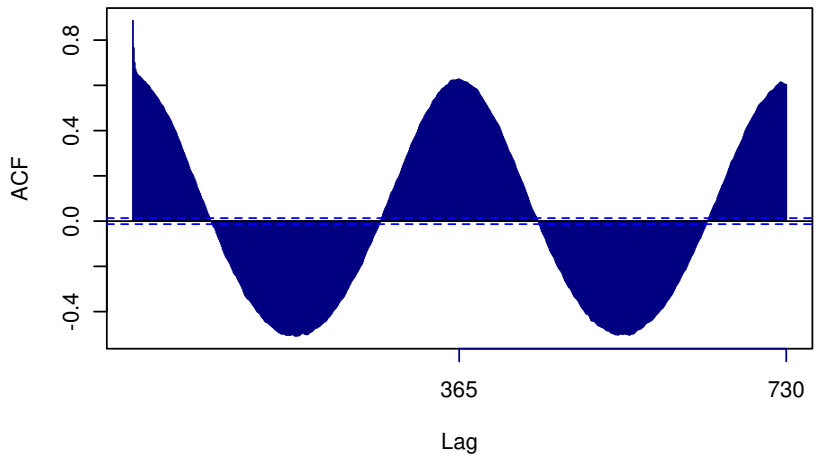


Figure 5.7: Temperature Autocorrelation Plot

We end the exploratory data analysis by performing classical decomposition as a means to understand the underlying structure. We used the additive decomposition seeing it performed best in the rainfall data set. Again, if the additive seasonality is the correct type of seasonality, we can separate the noise from the features that deterministic components of level, trend and seasonality. Figure 5.8 suggest that temperature has been steadily rising, albeit over a very long period.

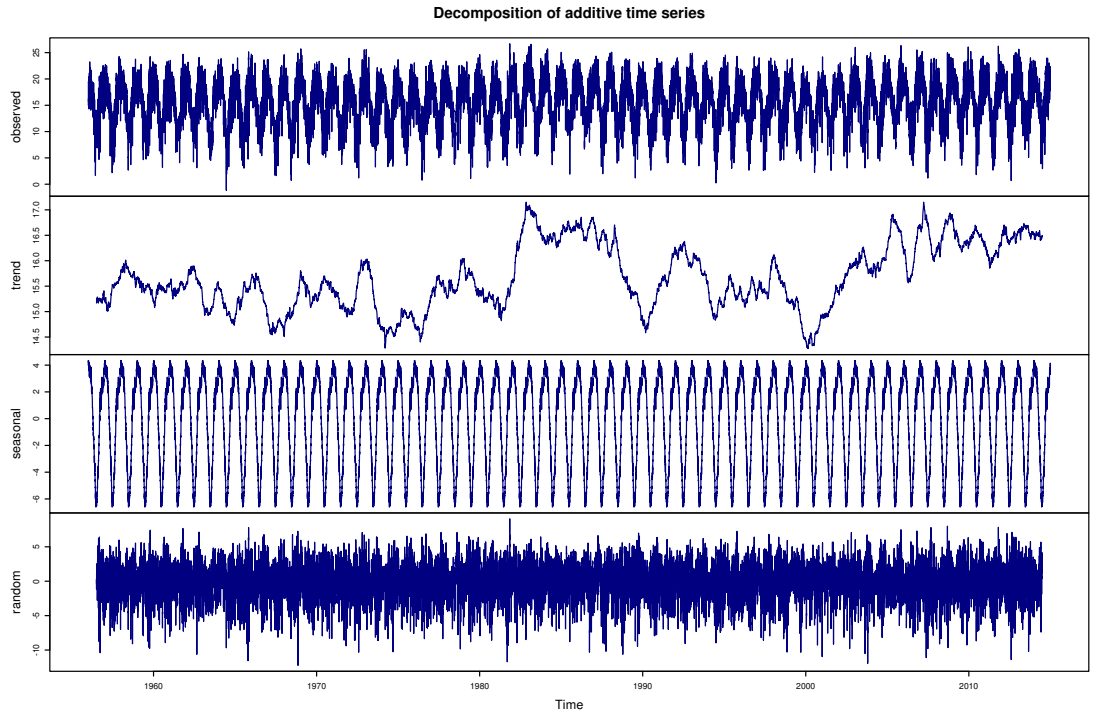


Figure 5.8: Temperature Classical Decomposition

In summing up the exploratory analysis, we note that the temperature time series has strong seasonality and a possible trend. Next, we fit the exponential smoothing models to the data.

5.3.2 Model Identification and Fitting

As with rainfall dataset, the temperature dataset was divided into 80% *in-sample* and 20% *out-of-sample*. The observations were also adjusted with a constant to make the series positive (i.e. $y_t^* = y_t - min(y_t) + 0.00005$). The forecasts can be transformed back to the original by subtracting/adding the adjustment. The accuracy measures are affected in the presence of zeros and negative observations. Four models were fit to the data. They are seasonal naive, simple exponential smoothing (SES), Holt's (double exponential smoothing) and Holt-Winters (triple exponential smoothing). Holt-Winters models with additive seasonality and multiplicative seasonality were fit separately. The models were selected via the in-

formation criteria and parameters estimated via maximum likelihood estimation. The estimated model smoothing parameters are as shown in Table 5.6.

Table 5.6: Estimated Model Parameters: Temperature

Parameter	SES	Holt	H-W Add	H-W Mult
Alpha (α)	0.9999	0.9999	0.8756	0.7241
Beta (β)		0.0001	0	0
Gamma (γ)			1	0.4419

We describe the estimated parameters from the SES, Holt's and Holt-Winters exponential smoothing models. The estimated value of the level parameter α for the SES model is 0.9999. The value is very high indicating that the local level of the temperature time series is based on very recent observations. The level of the time series is changing quickly.

The Holt's method requires the estimation of the local level parameter α and trend parameter β. The estimated smoothing values for the Holt's method are 0.9999 and 0.0001 for α and β respectively. The very high value of α implies that more weight is placed on recent observations than older observations. This is a sign of rapidly changing local level of the temperature. The trend parameter is close to zero meaning that there is little updating of the trend over time. Thus, the temperature time series is slowly changing over time. This is consistent with the finding of the exploratory analysis.

Two Holt-Winters exponential smoothing models were fit to the temperature dataset. One with additive seasonality and another with multiplicative seasonality. Each model requires estimation of three parameters: α, β and γ. The estimated values of the smoothing parameters for the Holt-Winters model with additive seasonality are 0.8756, 0 and 1 for α, β and γ respectively. The estimated α is high which means the local level is updating rapidly and more weight is placed on recent observations than older ones. The smoothing parameter for the trend β

is zero meaning the trend of the temperature time series is not adjusted over time. Its current value remains the initial trend value. The trend of the temperature time series is unchanging over time. The smoothing parameter for the season is 1 which is very high indicating that it is based on the most recent observation. The estimated values for the Holt-Winters model with multiplicative seasonality are 0.7241, 0 and 0.4419 for α, β and γ respectively. The multiplicative model suggests a high value of α indicating that the local level is based on recent observations. As with the additive model, the value of β suggests no updating of the trend in the temperature time series. The value of the seasonal component γ is relatively low indicating that both recent and less recent observations are used to estimate the seasonal component. Overall, all the three models indicate that temperature time series has a rapidly changing local level. In addition, the Holt and Holt-Winters models indicate that the temperature time series has stable trend (i.e. not changing over time). We check the assumption of normally distributed model residuals for the temperature exponential smoothing models. Figure A.7 suggests that residuals are normally and independently distributed with constant variance. The formal portmanteau tests, Ljung-Box and Box-Pierce, confirmed with large $p-$values that the residuals were *white noise*. We assess the model fit and forecasting performance using *in-sample* and *out-of-sample* accuracy measures respectively.

5.3.3 Model Accuracy

The results of model *in-sample* performance are presented in Table 5.7. All the models outperformed the seasonal naive method. In fact, all the models perform very well with very high accuracy in mean error (ME), root mean square error (RMSE), mean absolute error (MAE). They performed worse on mean percentage error (MPE) and mean absolute percentage error (MAPE). SES and Holt

performed better than the Holt-Winters models in most measures. In terms of the Holt-Winters, the additive model outperform the multiplicative model. The performance of all the models is indication that we have selected the appropriate models. As stated before, *in-sample* performance is not necessarily an indicator of forecasting performance.

Table 5.7: *In-Sample* Accuracy Measures: Temperature

METHOD	ME	RMSE	MAE	MPE	MAPE	MASE
Naive	0.012	3.653	2.838	-11,672.380	11,688.510	1
SES	0.0001	2.036	1.535	-3,397.768	3,407.303	0.541
Holt	0.001	2.036	1.535	-3,395.517	3,405.058	0.541
H-W Add	0.0001	2.331	1.768	5,115.279	5,126.968	0.622
H-W Mult	-0.075	2.467	1.863	6,381.416	6,394.349	0.655

We proceed to investigate the *out-of-sample* performance of the chosen models. Table 5.8 shows the accuracy measures based on the *out-of-sample* data, i.e., data which has not been used in the model fitting. As expected, the seasonal naive method performed worst across most of the measures. Although all the exponential smoothing methods produced good *out-of-sample* forecasts, the simple exponential smoothing methods performed best. This was followed by the Holt's exponential smoothing method. Interestingly, the Holt-Winters with multiplicative seasonality performed better than the Holt-Winters with additive seasonality. This is unlike the rainfall dataset. Also, all the accuracy measures in the *out-of-sample* evaluation were worse than in *in-sample* evaluation. This adds to the view

Table 5.8: *Out-of-sample* Accuracy Measures: Temperature

METHOD	ME	RMSE	MAE	MPE	MAPE	MASE
Naive	0.301	3.641	2.784	−1.829	18.748	0.981
SES	−1.461	3.107	1.863	−9.287	10.982	0.656
Holt	−1.463	3.109	1.864	−9.300	10.987	0.657
H-W Add	−1.491	3.596	2.788	−13.548	19.970	0.982
H-W Mult	−0.326	3.466	2.674	−4.668	17.915	0.942

that the predictive ability of a fitted forecasting model is best done using the data the model has not seen before. Since the model has been fitted using *in-sample* data, it is has already been tuned to this data hence higher predictive power on the *in-sample* data than the *out-of-sample* data.

Chapter 6

Discussion and Conclusion

6.1 Introduction

This chapter looks at the extent to which aims and objectives of the research report have been met. Section 6.2 presents a summary and discussion of the exponential smoothing forecasting results. Section 6.3 outlines areas in which the exponential smoothing methods can be furthered. The chapter ends with conclusions.

6.2 Summary of Findings

This section presents a summary of the results of the applying exponential smoothing methods to two weather related datasets. In total, four methods were applied to the rainfall and temperature datasets. They include the three exponential smoothing methods and a seasonal naive method. The Holt-Winters method with additive seasonality was the best method for forecasting rainfall time series. The SES and Holt's methods performed worse than the seasonal naive so as to be disregarded for modelling monthly rainfall time series. The Holt-Winters model is, therefore, the most likely data generating process for the average rainfall received

in Gauteng Province.

The simple exponential smoothing method (SES) performed best for forecasting daily temperature time series. The three exponential time series models considered generally performed well across all the accuracy measures. All the exponential smoothing methods performed better than the seasonal naive method in forecasting daily temperature time series. The multiplicative Holt-Winters methods performed better than the additive model. From a theoretical point of view, we were expecting the Holt-Winters method to emerge as the most accurate. However, the temperature time series showed that simplest of them was most accurate. Since the rainfall data was already modified, the exponential smoothing methods performed relatively poorly. The temperature data had minimal modification and the exponential smoothing methods performed well. This suggests real life data contain information more information than manipulated ones. The SES is the most likely process that generated the daily temperature data recorded at the Johannesburg Weather Station.

A limitation of the research is the lack of a Monte-Carlo simulation. This means that the study did not investigate the properties of the smoothing parameters α, β and γ. Such a study would have resulted in more certainty around the study results.

6.3 Future Research

The statistical properties of exponential smoothing methods are still to be fully understood. There are areas for which this research could be extended to. For instance, the exploratory analysis highlighted the potential issue of outlying values. It would be a worthwhile statistical pursuit to assess the robustness of the methods in the presence of outliers or influential points. A paper by Gelper, Fried, and

Croux [19] found that the Holt-Winters exponential smoothing methods can be affected by outlying observations.

There are more directions into which the research could be taken:

- Performing Monte-Carlo simulations on the exponential smoothing methods
- Comparison of Holt-Winters exponential smoothing methods with General exponential smoothing (GES) methods and
- Comparison of Holt-Winters exponential methods with the ARIMA models on these data.

Overall, the empirical investigation showed that the performance of exponential smoothing methods vary depending on the quality of data available for modelling. The result show that not one of exponential smoothing methods can be used on its own. While the Holt-Winters model was the theoretically correct model to apply on these data, the SES outperformed it. Forecasters will have to assess the performance of the three exponential models before they can decide which one to use for forecasting.

References

[1] S. Arlot and A. Celisse. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40–79, 2010.

[2] F. Beichelt. *Stochastic Processes in Science, Engineering and Finance*. CRC Press, 2006.

[3] F. Beichelt and L. Fatti. *Stochastic Processes and Their Applications*. CRC Press, 2002.

[4] B. Billah, M. L. King, R. D. Snyder, and A. B. Koehler. Exponential smoothing model selection for forecasting. *International Journal of Forecasting*, 22 (2):239–247, 2006.

[5] G. E. Box and D. A. Pierce. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American Statistical Association*, 65(332):1509–1526, 1970.

[6] G. E. Box, G. M. Jenkins, and G. C. Reinsel. *Time Series Analysis: Forecasting and Control*, volume 734. John Wiley & Sons, 2011.

[7] P. J. Brockwell and R. A. Davis. *Time Series: Theory and Methods*. Springer Science & Business Media, 2009.

[8] R. G. Brown. Statistical forecasting for inventory control. 1959.

[9] R. G. Brown. *Smoothing, forecasting and prediction of discrete time series*. Courier Corporation, 1962.

[10] P. Burman, E. Chow, and D. Nolan. A cross-validatory method for dependent data. *Biometrika*, 81(2):351–358, 1994.

[11] C. Chatfield. *Problem Solving: A Statistician's Guide*. CRC Press, 1995.

[12] C. Chatfield. *The Analysis of Time Series: An Introduction*. CRC press, 2013.

[13] E. B. Dagum. Time series modelling and decomposition. *Statistica*, 70(4): 433–457, 2010.

[14] J. G. De Gooijer and R. J. Hyndman. 25 years of time series forecasting. *International Journal of Forecasting*, 22(3):443–473, 2006.

[15] W. Feller. *An Introduction to Probability Theory and Its Applications. Volume I.* John Wiley & Sons London-New York-Sydney-Toronto, 1968.

[16] E. S. Gardner. Exponential smoothing: The state of the art. *Journal of Forecasting*, 4(1):1–28, 1985.

[17] E. S. Gardner. Exponential smoothing: The state of the art part ii. *International Journal of Forecasting*, 22(4):637–666, 2006.

[18] S. Geisser. The predictive sample reuse method with applications. *Journal of the American Statistical Association*, 70(350):320–328, 1975.

[19] S. Gelper, R. Fried, and C. Croux. Robust forecasting with exponential and holt–winters smoothing. *Journal of Forecasting*, 29(3):285–300, 2010.

[20] G. Grimmett and D. Stirzaker. *Probability and Random Processes*, volume 2. Oxford Univ Press, 1992.

[21] C. C. Holt. Forecasting trends and seasonal by exponentially weighted averages. *International Journal of Forecasting*, 20(1):5–10, 1957.

[22] R. J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2014.

[23] R. J. Hyndman and Y. Khandakar. Automatic time series forecasting: the forecast package for r. *Journal of Statistical Software*, 27(Issue 3), 2008. URL https://www.jstatsoft.org/article/view/v027i03.

[24] R. J. Hyndman and A. B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688, 2006.

[25] R. J. Hyndman, A. B. Koehler, R. D. Snyder, and S. Grose. A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of Forecasting*, 18(3):439–454, 2002.

[26] S. C. Larson. The shrinkage of the coefficient of multiple correlation. *Journal of Educational Psychology*, 22(1):45, 1931.

[27] G. M. Ljung and G. E. Box. On a measure of lack of fit in time series models. *Biometrika*, 65(2):297–303, 1978.

[28] S. Makridakis and M. Hibon. Exponential smoothing: The effect of initial values and loss functions on post-sample forecasting accuracy. *International Journal of Forecasting*, 7(3):317–330, 1991.

[29] S. Makridakis and M. Hibon. The m3-competition: results, conclusions and implications. *International Journal of Forecasting*, 16(4):451–476, 2000.

[30] S. Makridakis, C. Chatfield, M. Hibon, M. Lawrence, T. Mills, K. Ord, and L. F. Simmons. The m2-competition: A real-time judgmentally based forecasting study. *International Journal of Forecasting*, 9(1):5–22, 1993.

[31] E. McKenzie. General exponential smoothing and the equivalent arma process. *Journal of Forecasting*, 3(3):333–344, 1984.

[32] F. Mosteller and J. W. Tukey. Data analysis, including statistics. 1968.

[33] J. F. Muth. Optimal properties of exponentially weighted forecasts. *Journal of the American Statistical Association*, 55(290):299–306, 1960.

[34] J. K. Ord, A. Koehler, and R. D. Snyder. Estimation and prediction for a class of dynamic nonlinear statistical models. *Journal of the American Statistical Association*, 92(440):1621–1629, 1997.

[35] C. C. Pegels. Exponential forecasting: some new variations. *Management Science*, 15:311–315, 1969.

[36] W. M. Persons. An index of business conditions. *Review of Economic Statistics*, 1(2):112–113, 1919.

[37] W. M. Persons. A non-technical explanation of the index of general business conditions. *The Review of Economic Statistics*, pages 39–48, 1920.

[38] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2015. URL http://www.R-project.org/.

[39] S. M. Ross. *Introduction to probability models*. Academic press, 2014.

[40] R. H. Shumway and D. S. Stoffer. *Time Series Analysis and its Applications: with R examples*. Springer Science & Business Media, 2010.

[41] R. Snyder. Recursive estimation of dynamic linear models. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 272–276, 1985.

[42] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 111–147, 1974.

[43] J. W. Taylor. Exponential smoothing with a damped multiplicative trend. *International Journal of Forecasting*, 19(4):715–725, 2003.

[44] P. R. Winters. Forecasting sales by exponentially weighted moving averages. *Management Science*, 6(3):324–342, 1960.

Appendix A

Additional Results

This section presents some of the tables and graphs not included in the main chapters.

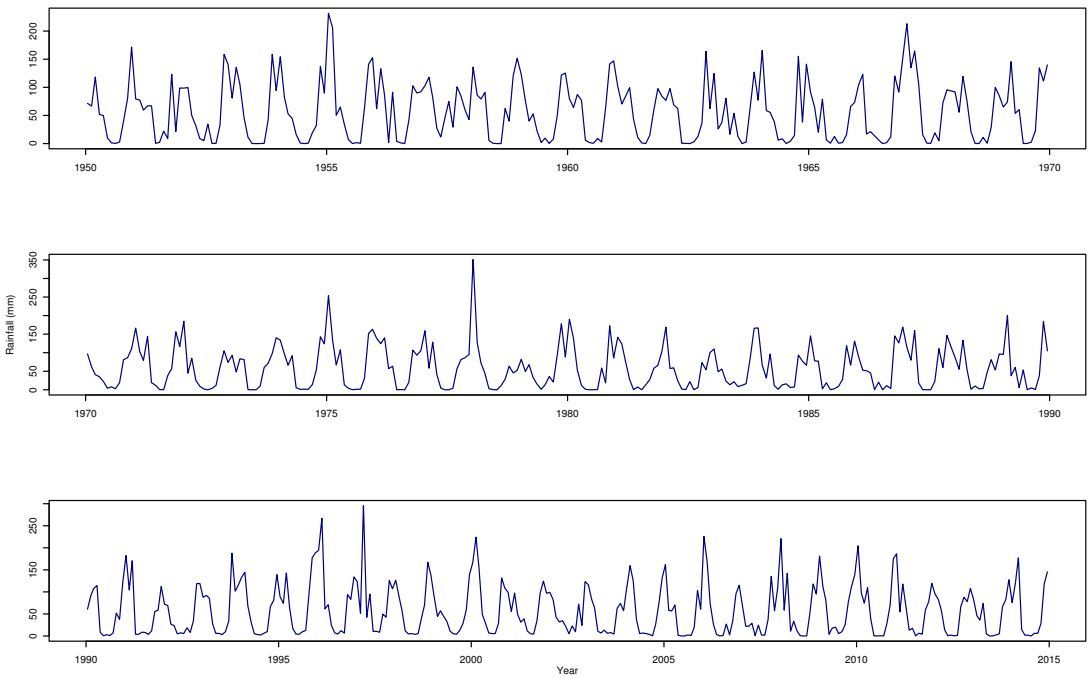


Figure A.1: Rainfall time series plot in multiple panels

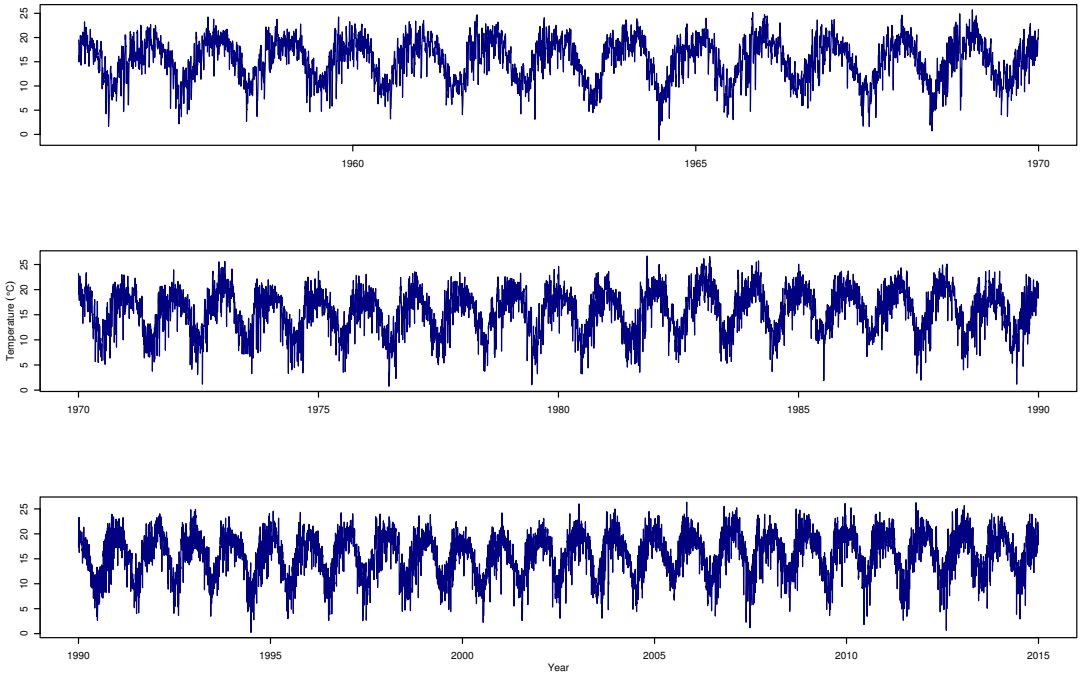


Figure A.2: Temperature time series plot in multiple panels

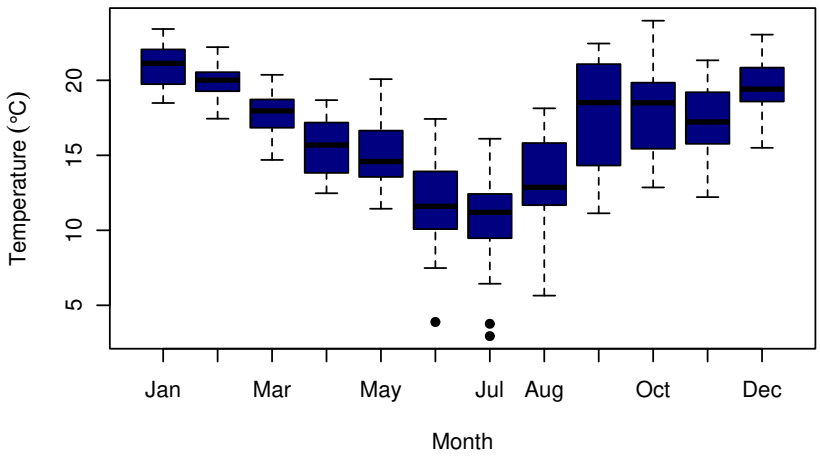


Figure A.3: Box-and-whisker plot for Temperature, 2014

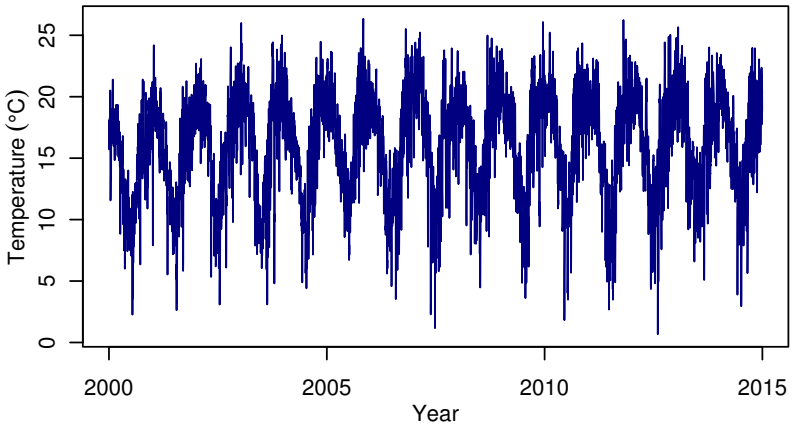


Figure A.4: Average Daily Temperature: 2000-2014

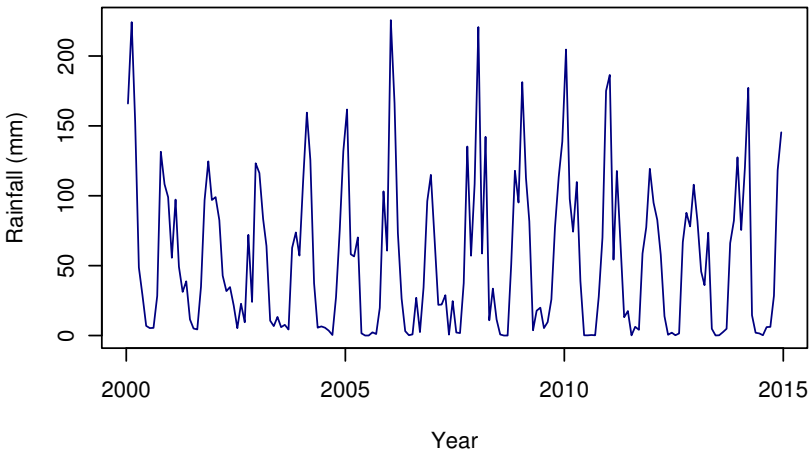
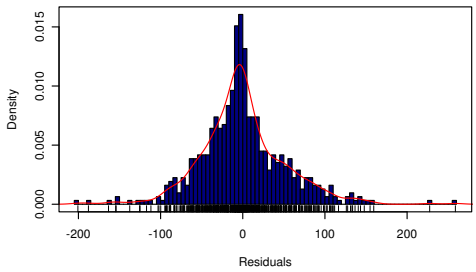
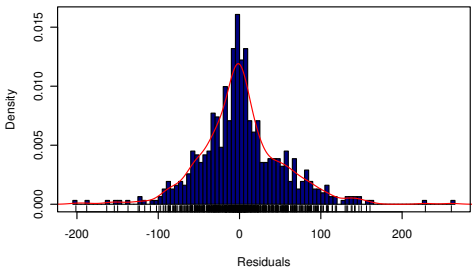


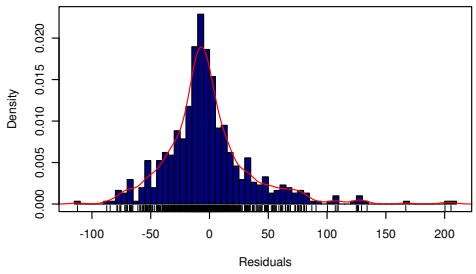
Figure A.5: Average Monthly Rainfall: 2000-2014



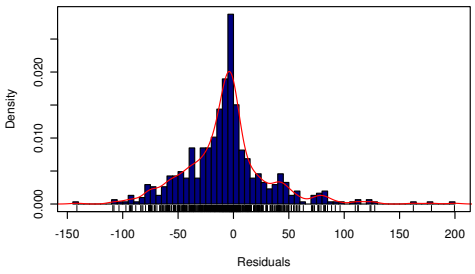
(a) SES



(b) Holt

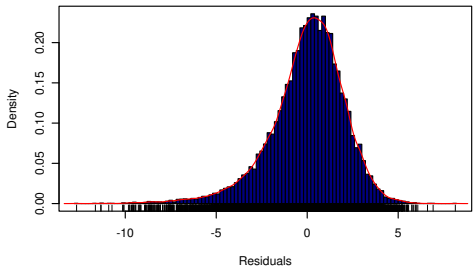


(c) HW-Add

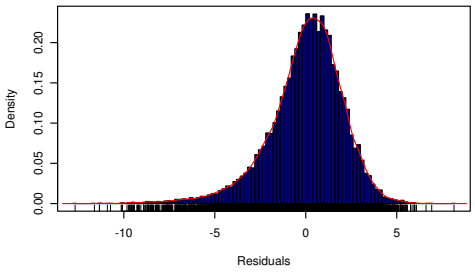


(d) HW-Mult

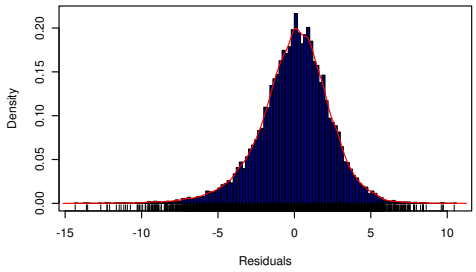
Figure A.6: Residual Plots: Rainfall



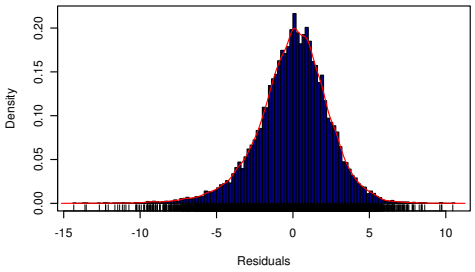
(a) SES



(b) Holt



(c) HW-Add



(d) HW-Mult

Figure A.7: Residual Plots: Temperature

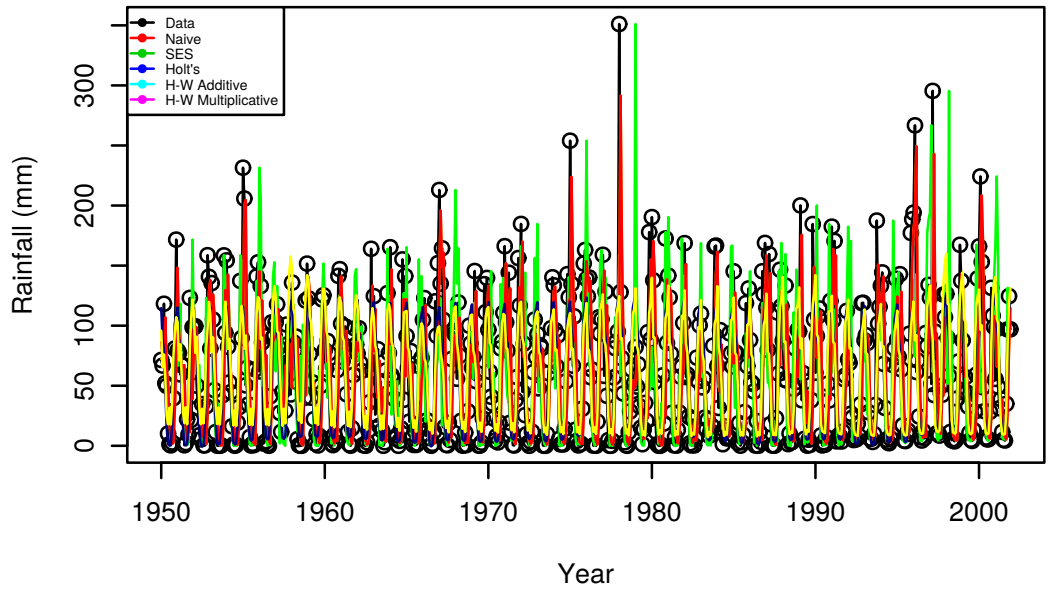


Figure A.8: Actual versus *in-sample* fit: Rainfall

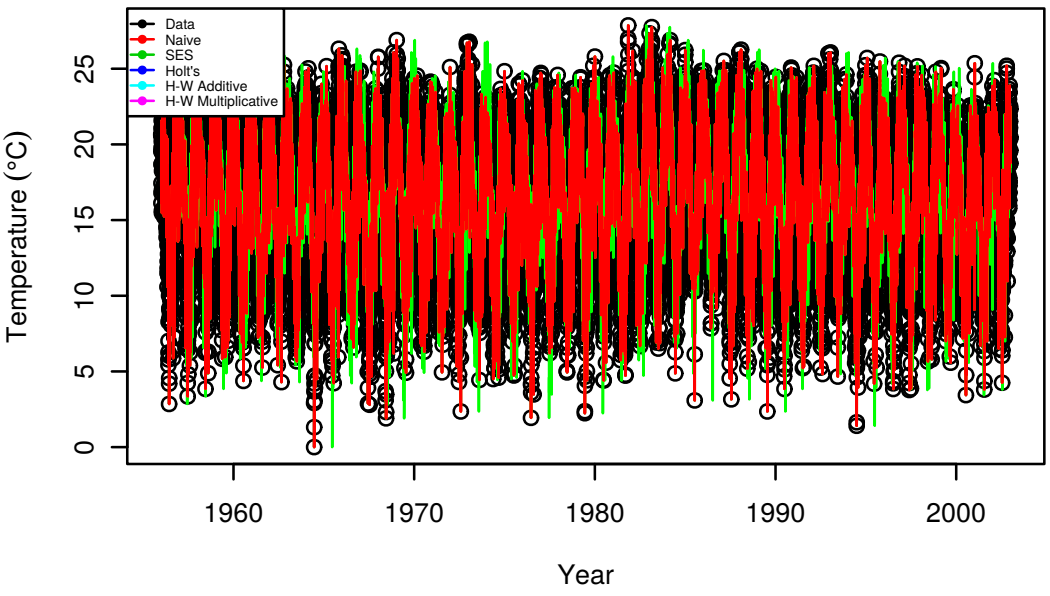


Figure A.9: Actual versus *in-sample* fit: Temperature

Appendix B

R Code

This section contains the R-language code that was used to generate the output results in the research report.

```
setwd("~/Dropbox/WitsAll/analysis2015/")
library("Cairo")
library("gridExtra")
library("ggplot2")
library("ggfortify")
library("lubridate")
library("lattice")
library("reshape")
library("scales")
library("dplyr")
library("stargazer")
library("forecast")

#suppress scientific notation
options(scipen=999)

##################RAINFALL################

#open data set
rainfall<-readRDS("./rainfalldata/rainfalldata.RDS")

#check for missing data
sum(is.na(rainfall$Rainfall))

#descriptive
stargazer(rainfall["Rainfall"], type="latex")

#slices
rainfall$Period[rainfall$Year>=1950 & rainfall$Year
   <1970]<-"[1950-1970)"
rainfall$Period[rainfall$Year>=1970 & rainfall$Year
   <1990]<-"[1970-1990)"
rainfall$Period[rainfall$Year>=1990 & rainfall$Year
   <2015]<-"[1990-2014]"
table(rainfall$Period)
```

```
#exploratory data analysis
CairoPDF("preplot1a.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
plot(Rainfall~Date, data=rainfall, type="l", col="navy", xlab="
   Year", ylab="Rainfall (mm)")
dev.off()

CairoPDF("preplot1b.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
plot(Rainfall~Date, data=rainfall, type="o", col="navy", xlab="
   Year", ylab="Rainfall (mm)", pch=20)
dev.off()

CairoPDF("preplot1c.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
boxplot(Rainfall~Year, data=rainfall, xlab="Year", ylab="Rainfall
   (mm)", col="navy", pch=20)
dev.off()

CairoPDF("preplot1d.pdf", height=4, width=6, pointsize=10, family
   ="serif")
hist(rainfall$Rainfall, xlab="Rainfall (mm)", col="navy", breaks
   =99, main=NULL, prob=TRUE)
lines(density(rainfall$Rainfall), col="red", lwd=1)
rug(rainfall$Rainfall)
box()
dev.off()

CairoPDF("preplot1e.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
trainfall<-ts(rainfall$Rainfall, start=c(1950,1), frequency=12)
Acf(trainfall, col="navy", main="")
dev.off()

CairoPDF("preplot1f.pdf", height=8.5, width=12.75, pointsize=12,
   family="serif")
trainfall<-ts(rainfall$Rainfall, start=c(1950,1), frequency=12)
plot(decompose(trainfall), col="navy")
dev.off()

CairoPDF("preplot1g.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
plot(Rainfall~Date, data=rainfall[rainfall$Year>1999,], type="l",
   col="navy", xlab="Year", ylab="Rainfall (mm)")
dev.off()

CairoPDF("preplot1h.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
plot(Rainfall~Date, data=rainfall[rainfall$Year>2009,], type="l",
   col="navy", xlab="Year", ylab="Rainfall (mm)")
dev.off()

CairoPDF("preplot1i.pdf", height=8.5, width=12.75, pointsize=12,
   family="serif")
```

```
par1<-par(mfrow=c(3,1))
plot(Rainfall~Date, data=rainfall[rainfall$Period=="[1950-1970)
   ",], type="l", col="navy", xlab="", ylab="")
plot(Rainfall~Date, data=rainfall[rainfall$Period=="[1970-1990)
   ",], type="l", col="navy", xlab="", ylab="Rainfall (mm)")
plot(Rainfall~Date, data=rainfall[rainfall$Period
   =="[1990-2014]",], type="l", col="navy", xlab="Year", ylab="",
     mgp = c(2, 1, 0))
par(par1)
dev.off()

##################ANALYSIS.1################

# added a small quantity to prevent zero values.
epsi <- min(rainfall$Rainfall)+0.000005
epsi
rainfall$Rainfall <- rainfall$Rainfall+epsi
min(rainfall$Rainfall)

#split into train/testing datasets
index<-1:(0.8*nrow(rainfall))
trainRainfall <- rainfall[index,]
testRainfall <- rainfall[-index,]

#convert data into time series object
alldata <- ts(rainfall$Rainfall, start=c(1950,1), frequency=12)
traindata1 <- ts(trainRainfall["Rainfall"], frequency=12, start
   =1950)
testdata1 <- ts(testRainfall["Rainfall"], frequency=12, start
   =2002)

#Fit Models
naivefit <- snaive(traindata1)
sesfit <- ses(traindata1)
holtfit <- holt(traindata1)
hwfit_add <- hw(traindata1,seasonal="additive")
hwfit_mult <- hw(traindata1,seasonal="multiplicative")

naivefit$model
sesfit$model
holtfit$model
hwfit_add$model
hwfit_add$model

naivefit <- snaive(traindata1)
sesfit <- HoltWinters(traindata1, beta=FALSE, gamma=FALSE)
holtfit <- HoltWinters(traindata1, gamma=FALSE)
hwfit_add <- HoltWinters(traindata1,seasonal="additive")
hwfit_mult <- HoltWinters(traindata1,seasonal="multiplicative")

hwfit_add$alpha
hwfit_add$beta
hwfit_add$gamma

hwfit_mult$alpha
```

```
hwfit_mult$beta
hwfit_mult$gamma

#In-sample accuracy
naiveacc <- accuracy(naivefit)[1,]
sesacc <- accuracy(sesfit)[1,]
holtacc <- accuracy(holtfit)[1,]
hwacc_add <- accuracy(hwfit_add)[1,]
hwacc_mult <- accuracy(hwfit_mult)[1,]

accuracy1 <- data.frame(rbind(naiveacc,sesacc,holtacc,hwacc_add,
    hwacc_mult))
methods <- data.frame(METHOD=c("Naive","SES", "Holt", "H-W Add", "
    H-W Mult"))
accuracy1 <- cbind(methods,accuracy1)
accuracy1<-accuracy1[,-length(accuracy1)]
row.names(accuracy1)<-NULL
accuracy1
stargazer(accuracy1, type="latex", summary=FALSE)

#Out of sample accuracy
naiveacc <- accuracy(naivefit, testdata1)[2,]
sesacc <- accuracy(sesfit, testdata1)[2,]
holtacc <- accuracy(holtfit, testdata1)[2,]
hwacc_add <- accuracy(hwfit_add, testdata1)[2,]
hwacc_mult <- accuracy(hwfit_mult, testdata1)[2,]

accuracy1 <- data.frame(rbind(naiveacc,sesacc,holtacc,hwacc_add,
    hwacc_mult))
methods <- data.frame(METHOD=c("Naive","SES", "Holt", "H-W Add", "
    H-W Mult"))
accuracy1 <- cbind(methods,accuracy1)
accuracy1<-accuracy1[,c(-(length(accuracy1)-1),-length(accuracy1))
    ]
row.names(accuracy1)<-NULL
accuracy1
stargazer(accuracy1, type="latex", summary=FALSE)

# diagnostic checks
res1 <- residuals(naivefit)
res2 <- residuals(sesfit)
res3 <- residuals(holtfit)
res4 <- residuals(hwfit_add)
res5 <- residuals(hwfit_mult)

# plot of residuals
CairoPDF("preplotr1a1.pdf", height=4.5, width=6.75, pointsize=12,
    family="serif")
plot(res1, ylab="Rainfall (mm)", xlab="Year", col="navy")
dev.off()
CairoPDF("preplotr1a2.pdf", height=4.5, width=6.75, pointsize=12,
    family="serif")
plot(res2, ylab="Rainfall (mm)", xlab="Year", col="navy")
dev.off()
```

```
CairoPDF("preplotr1a3.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
plot(res3, ylab="Rainfall (mm)", xlab="Year", col="navy")
dev.off()
CairoPDF("preplotr1a4.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
plot(res4, ylab="Rainfall (mm)", xlab="Year", col="navy")
dev.off()
CairoPDF("preplotr1a5.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
plot(res5, ylab="Rainfall (mm)", xlab="Year", col="navy")
dev.off()

#ACF plots
CairoPDF("preplotr1b1.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
Acf(res1, main="", col="navy")
dev.off()
CairoPDF("preplotr1b2.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
Acf(res2, main="", col="navy")
dev.off()
CairoPDF("preplotr1b3.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
Acf(res3, main="", col="navy")
dev.off()
CairoPDF("preplotr1b4.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
Acf(res4, main="", col="navy")
dev.off()
CairoPDF("preplotr1b5.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
Acf(res5, main="", col="navy")
dev.off()

#histograms of residuals
CairoPDF("preplotr1c1.pdf", height=4, width=6, pointsize=10,
   family="serif")
hist(res1, main=NULL, col="navy", breaks=99, xlab="Residuals",
   prob=TRUE)
lines(density(res1, na.rm=T), col="red", lwd=1)
rug(res1)
box()
dev.off()
CairoPDF("preplotr1c2.pdf", height=4, width=6, pointsize=10,
   family="serif")
hist(res2, main=NULL, col="navy", breaks=99, xlab="Residuals",
   prob=TRUE)
lines(density(res2, na.rm=T), col="red", lwd=1)
rug(res2)
box()
dev.off()
CairoPDF("preplotr1c3.pdf", height=4, width=6, pointsize=10,
   family="serif")
```

```
hist(res3, main=NULL, col="navy", breaks=99, xlab="Residuals",
   prob=TRUE)
lines(density(res3, na.rm=T), col="red", lwd=1)
rug(res3)
box()
dev.off()
CairoPDF("preplotr1c4.pdf", height=4, width=6, pointsize=10,
   family="serif")
hist(res4, main=NULL, col="navy", breaks=99, xlab="Residuals",
   prob=TRUE)
lines(density(res4, na.rm=T), col="red", lwd=1)
rug(res4)
box()
dev.off()
CairoPDF("preplotr1c5.pdf", height=4, width=6, pointsize=10,
   family="serif")
hist(res5, main=NULL, col="navy", breaks=99, xlab="Residuals",
   prob=TRUE)
lines(density(res5, na.rm=T), col="red", lwd=1)
rug(res5)
box()
dev.off()

# lag=h and fitdf=K
Box.test(res1, lag=12, fitdf=0)
Box.test(res2, lag=12, fitdf=0)
Box.test(res3, lag=12, fitdf=0)
Box.test(res4, lag=12, fitdf=0)
Box.test(res5, lag=12, fitdf=0)

Box.test(res1, lag=12, fitdf=0, type="Lj")
Box.test(res2, lag=12, fitdf=0, type="Lj")
Box.test(res3, lag=12, fitdf=0, type="Lj")
Box.test(res4, lag=12, fitdf=0, type="Lj")
Box.test(res5, lag=12, fitdf=0, type="Lj")

# more descriptives
CairoPDF("preplotd1a.pdf", height=8.5, width=12.75, pointsize=12,
   family="serif")
lag.plot(testdata1, lags=9, do.lines=FALSE, col="navy")
dev.off()

CairoPDF("preplotd1b.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
Acf(testdata1, col="navy", main="")
dev.off()

##################TEMPERATURE################

#open data set
temperature<-readRDS("./temperaturedata/temperaturedata.RDS")

temp2<-temperature

#check for missing data
```

```
sum(is.na(temperature$Temperature))

#excluding leap days
temperature<-temperature[!(temperature$month=="02" &
   temperature$day=="29"),]

#create  data subsets
tdataset1<-temperature %>% group_by(Date) %>% summarise(MaxTemp=
   max(Temperature, na.rm=TRUE), MinTemp=min(Temperature, na.rm=
   TRUE), AverageTemp=mean(Temperature, na.rm=TRUE))

tdataset2<-temperature %>% mutate(week=week(Date)) %>% group_by(
   year, week) %>% summarise(MaxTemp=max(Temperature, na.rm=TRUE)
   , MinTemp=min(Temperature, na.rm=TRUE), AverageTemp=mean(
   Temperature, na.rm=TRUE))

tdataset3<-temperature %>% group_by(year, month) %>% summarise(
   MaxTemp=max(Temperature, na.rm=TRUE), MinTemp=min(Temperature,
    na.rm=TRUE), AverageTemp=mean(Temperature, na.rm=TRUE))

tdataset4<-temperature %>% mutate(quarter=quarter(Date)) %>%
   group_by(year, quarter) %>% summarise(MaxTemp=max(Temperature,
    na.rm=TRUE), MinTemp=min(Temperature, na.rm=TRUE),
   AverageTemp=mean(Temperature, na.rm=TRUE))

tdataset5<-temperature %>% group_by(year) %>% summarise(MaxTemp=
   max(Temperature, na.rm=TRUE), MinTemp=min(Temperature, na.rm=
   TRUE), AverageTemp=mean(Temperature, na.rm=TRUE))

temperature<-tdataset1 %>% mutate(Temperature=AverageTemp, year=
   year(Date))

#slices
temperature$Period[year(temperature$Date)>=1956 & year(
   temperature$Date)<1970]<-"[1956-1970)"
temperature$Period[year(temperature$Date)>=1970 & year(
   temperature$Date)<1990]<-"[1970-1990)"
temperature$Period[year(temperature$Date)>=1990 & year(
   temperature$Date)<2015]<-"[1990-2014]"
table(temperature$Period)

#impute missing values
table(temperature$Date[is.na(temperature$Temperature)])
temperature$Temperature<-zoo::na.spline(temperature$Temperature)

#summary
stargazer(data.frame(temperature[,"Temperature"]), type="latex")

CairoPDF("preplot2a.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
plot(Temperature~Date, data=temperature, type="l", col="navy",
   xlab="Year", ylab=expression(Temperature~(degree*C)),  mgp = c
   (2, 1, 0))
dev.off()
```

```
CairoPDF("preplot2b.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
plot(Temperature~Date, data=temperature, type="o", col="navy",
   xlab="Year", ylab=expression(Temperature~(degree*C)), pch=20,
    mgp = c(2, 1, 0))
dev.off()

CairoPDF("preplot2c.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
boxplot(Temperature~year, data=temperature, xlab="Year", ylab=
   expression(Temperature~(degree*C)), pch=20, col="navy",  mgp =
    c(2, 1, 0))
dev.off()

CairoPDF("preplot2c2.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
boxplot(Temperature~month(Date, label=TRUE), data=temperature[
   temperature$year==2014,], type="l", col="navy", pch=20, xlab="
   Month", ylab=expression(Temperature~(degree*C)), mgp = c(2, 1,
    0))
dev.off()

CairoPDF("preplot2c3.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
boxplot(Temperature~Hour, data=temp2, type="l", col="navy", pch
   =20, xlab="Hour of Day", ylab=expression(Temperature~(degree*C
   )), mgp = c(2, 1, 0))
dev.off()

CairoPDF("preplot2d.pdf", height=4, width=6, pointsize=10, family
   ="serif")
hist(temperature$Temperature, xlab=expression(Temperature~(degree*
   C)), col="navy", breaks=99, main=NULL, prob=TRUE,  mgp = c(2,
   1, 0))
lines(density(temperature$Temperature), col="red", lwd=1)
rug(temperature$Temperature)
box()
dev.off()

CairoPDF("preplot2e.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
ttemperature<-ts(temperature$Temperature, start=c(1956,1,1),
   frequency=365)
Acf(ttemperature, col="navy", main="")
dev.off()

CairoPDF("preplot2f.pdf", height=8.5, width=12.75, pointsize=10,
   family="serif")
ttemperature<-ts(temperature$Temperature, start=c(1956,1,1),
   frequency=365)
plot(decompose(ttemperature), col="navy",  mgp = c(2, 1, 0))
dev.off()

CairoPDF("preplot2g.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
```

```
plot(Temperature~Date, data=temperature[temperature$year>1999,],
   type="l", col="navy", xlab="Year", ylab=expression(Temperature
   ~(degree*C)),  mgp = c(2, 1, 0))
dev.off()

CairoPDF("preplot2h.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
plot(Temperature~Date, data=temperature[temperature$year>2009,],
   type="l", col="navy", xlab="Year", ylab=expression(Temperature
   ~(degree*C)),  mgp = c(2, 1, 0))
dev.off()

CairoPDF("preplot2h2.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
plot(Temperature~Date, data=temperature[temperature$year==2014,],
   type="l", col="navy", xlab="Year", ylab=expression(Temperature
   ~(degree*C)),  mgp = c(2, 1, 0))
dev.off()

CairoPDF("preplot2i.pdf", height=8.5, width=12.75, pointsize=12,
   family="serif")
par1<-par(mfrow=c(3,1))
plot(Temperature~Date, data=temperature[temperature$Period
   =="[1956-1970)",], type="l", col="navy", xlab="", ylab="", mgp
    = c(2, 1, 0))
plot(Temperature~Date, data=temperature[temperature$Period
   =="[1970-1990)",], type="l", col="navy", xlab="", ylab=
   expression(Temperature~(degree*C)), mgp = c(2, 1, 0))
plot(Temperature~Date, data=temperature[temperature$Period
   =="[1990-2014]",], type="l", col="navy", xlab="Year", ylab="",
    mgp = c(2, 1, 0))
par(par1)
dev.off()

##################ANALYSIS.2#################

# added a small quantity to prevent zero values.
epsi <- -1*min(temperature$Temperature)+0.000005
epsi
temperature$Temperature <- temperature$Temperature +epsi
min(temperature$Temperature)

#split into train/testing datasets
index <- 1:(0.8*nrow(temperature))

trainTemperature <- temperature[index,]
testTemperature <- temperature[-index,]

trainTemperature <- temperature[temperature$year<=2002,]
testTemperature <- temperature[temperature$year>=2003,]

#convert data into time series object
traindata2 <- ts(trainTemperature["Temperature"], frequency=365,
   start=1956)
```

```
testdata2 <- ts(testTemperature["Temperature"], frequency=365,
   start=2003)

#Fit Models
#naivefit <- snaive(traindata2)
sesfit <- ses(traindata2)
holtfit <- holt(traindata2)
hwfit_add <- HoltWinters(traindata2, seasonal="additive")
hwfit_mult <- HoltWinters(traindata2, seasonal="multiplicative")

naivefit$model
sesfit$model
holtfit$model

#naivefit <- snaive(traindata2)
sesfit <- HoltWinters(traindata2, beta=FALSE, gamma=FALSE)
holtfit <- HoltWinters(traindata2, gamma=FALSE)
hwfit_add <- HoltWinters(traindata2,seasonal="additive")
hwfit_mult <- HoltWinters(traindata2,seasonal="multiplicative")
hwfit_mult <- HoltWinters(traindata2)

hwfit_add$alpha
hwfit_add$beta
hwfit_add$gamma

hwfit_mult$alpha
hwfit_mult$beta
hwfit_mult$gamma

#In-sample accuracy
naiveacc <- accuracy(naivefit)[1,]
sesacc <- accuracy(sesfit)[1,]
holtacc <- accuracy(holtfit)[1,]
hwacc_add <- accuracy(forecast(hwfit_add))[1,]
hwacc_mult <- accuracy(forecast(hwfit_mult))[1,]

accuracy2 <- data.frame(rbind(naiveacc,sesacc,holtacc,hwacc_add,
   hwacc_mult))
methods <- data.frame(METHOD=c("Naive","SES", "Holt", "H-W Add", "
   H-W Mult"))
accuracy2 <- cbind(methods,accuracy2)
accuracy2<-accuracy2[,-length(accuracy2)]
row.names(accuracy2)<-NULL
accuracy2
stargazer(accuracy2, type="latex", summary=FALSE)

#Out of sample accuracy
naiveacc <- accuracy(naivefit, testdata2)[2,]
sesacc <- accuracy(sesfit, testdata2)[2,]
holtacc <- accuracy(holtfit, testdata2)[2,]
hwacc_add <- accuracy(forecast(hwfit_add), testdata2)[2,]
hwacc_mult <- accuracy(forecast(hwfit_mult), testdata2)[2,]

accuracy2 <- data.frame(rbind(naiveacc,sesacc,holtacc,hwacc_add,
   hwacc_mult))
```

```
methods <- data.frame(METHOD=c("Naive","SES", "Holt", "H-W Add", "
   H-W Mult"))
accuracy2 <- cbind(methods,accuracy2)
accuracy2<-accuracy2[,c(-(length(accuracy2)-1),-length(accuracy2))
   ]
row.names(accuracy2)<-NULL
accuracy2
stargazer(accuracy2, type="latex", summary=FALSE)

# diagnostic checks
res1 <- residuals(naivefit)
res2 <- residuals(sesfit)
res3 <- residuals(holtfit)
res4 <- residuals(hwfit_add)
res5 <- residuals(hwfit_mult)

# plot of residuals
CairoPDF("preplotr2a1.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
plot(res1, ylab=expression(Temperature~(degree*C)), xlab="Year",
   col="navy")
dev.off()
CairoPDF("preplotr2a2.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
plot(res2, ylab=expression(Temperature~(degree*C)), xlab="Year",
   col="navy")
dev.off()
CairoPDF("preplotr2a3.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
plot(res3, ylab=expression(Temperature~(degree*C)), xlab="Year",
   col="navy")
dev.off()
CairoPDF("preplotr2a4.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
plot(res4, ylab=expression(Temperature~(degree*C)), xlab="Year",
   col="navy")
dev.off()
CairoPDF("preplotr2a5.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
plot(res5, ylab=expression(Temperature~(degree*C)), xlab="Year",
   col="navy")
dev.off()

#ACF plots
CairoPDF("preplotr2b1.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
Acf(res1, main="", col="navy")
dev.off()
CairoPDF("preplotr2b2.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
Acf(res2, main="", col="navy")
dev.off()
CairoPDF("preplotr2b3.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
Acf(res3, main="", col="navy")
```

```
dev.off()
CairoPDF("preplotr2b4.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
Acf(res4, main="", col="navy")
dev.off()
CairoPDF("preplotr2b5.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
Acf(res5, main="", col="navy")
dev.off()

#histograms of residuals
CairoPDF("preplotr2c1.pdf", height=4, width=6, pointsize=10,
   family="serif")
hist(res1, main=NULL, col="navy", breaks=99, xlab="Residuals",
   prob=TRUE)
lines(density(res1, na.rm=T), col="red", lwd=1)
rug(res1)
box()
dev.off()
CairoPDF("preplotr2c2.pdf", height=4, width=6, pointsize=10,
   family="serif")
hist(res2, main=NULL, col="navy", breaks=99, xlab="Residuals",
   prob=TRUE)
lines(density(res2, na.rm=T), col="red", lwd=1)
rug(res2)
box()
dev.off()
CairoPDF("preplotr2c3.pdf", height=4, width=6, pointsize=10,
   family="serif")
hist(res3, main=NULL, col="navy", breaks=99, xlab="Residuals",
   prob=TRUE)
lines(density(res3, na.rm=T), col="red", lwd=1)
rug(res3)
box()
dev.off()
CairoPDF("preplotr2c4.pdf", height=4, width=6, pointsize=10,
   family="serif")
hist(res4, main=NULL, col="navy", breaks=99, xlab="Residuals",
   prob=TRUE)
lines(density(res4, na.rm=T), col="red", lwd=1)
rug(res4)
box()
dev.off()
CairoPDF("preplotr2c5.pdf", height=4, width=6, pointsize=10,
   family="serif")
hist(res5, main=NULL, col="navy", breaks=99, xlab="Residuals",
   prob=TRUE)
lines(density(res5, na.rm=T), col="red", lwd=1)
rug(res5)
box()
dev.off()

# lag=h and fitdf=K
Box.test(res1, lag=365, fitdf=0)
Box.test(res2, lag=365, fitdf=0)
```

```
Box.test(res3, lag=365, fitdf=0)
Box.test(res4, lag=365, fitdf=0)
Box.test(res5, lag=365, fitdf=0)

Box.test(res1, lag=365, fitdf=0, type="Lj")
Box.test(res2, lag=365, fitdf=0, type="Lj")
Box.test(res3, lag=365, fitdf=0, type="Lj")
Box.test(res4, lag=365, fitdf=0, type="Lj")
Box.test(res5, lag=365, fitdf=0, type="Lj")

# more descriptives
CairoPDF("preplotd2a.pdf", height=8.5, width=12.75, pointsize=12,
   family="serif")
lag.plot(testdata2, lags=9, do.lines=FALSE, col="navy")
dev.off()

CairoPDF("preplotd2b.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
Acf(testdata2, col="navy", main="")
dev.off()

#open data set
exchangerate<-read.csv("./exchangerate/ExchangeRateDetail.csv", as
   .is = TRUE, header = TRUE, skip = 3, sep=",")

#housekeeping
exchangerate <- exchangerate[1:4000,]
names(exchangerate) <- c("Date", "exchangerate")
exchangerate$Date <- ymd(exchangerate$Date)
exchangerate <- exchangerate[order(exchangerate$Date),]

#time series plot
CairoPDF("preplot0a.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
plot(exchangerate~Date, data=exchangerate, type="l", col="navy",
   xlab="Year", ylab="Exchange rate (R/$)", ylim=c(0,20), xlim=c(
   ymd("2000-01-01"), ymd("2016-01-01")))
dev.off()

set.seed(23012016)
n <- 10000
x <- cumsum(sample(c(-1, 1), n, TRUE))

CairoPDF("preplot0b.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
plot(x, type="l", col="navy", xlab="Index(t)", ylab=expression(y[t
   ]))
dev.off()

y <- rnorm(1000,0,1)
CairoPDF("preplot0c.pdf", height=4.5, width=6.75, pointsize=12,
   family="serif")
plot(y, type="l", col="navy", xlab="Index (t)", ylab=expression(y[
   t]))
dev.off()
```

```
# create parameter tables
mydata1 <- read.csv("parameters1.csv", header = TRUE)
names(mydata1) <-gsub("\\.","-", names(mydata1))
stargazer(mydata1, type = "latex", summary = FALSE, rownames =
   FALSE)

mydata2 <- read.csv("parameters2.csv", header = TRUE)
names(mydata2) <-gsub("\\.","-", names(mydata2))
stargazer(mydata2, type = "latex", summary = FALSE, rownames =
   FALSE)
```