
By HAKATA JONATHAN
B1336501

A DISSERTATION SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS OF THE BACHELOR OF SCIENCE HONOURS DEGREE IN APPLIED MATHEMATICS

SUPERVISOR: MR. K. BASIRA

JUNE 2017
APPROVAL FORM

I Hakata Jonathan, do hereby declare that this submission is my own work apart from the references of other people’s work which has duly been acknowledged. I hereby declare that this work has neither been presented in whole nor in part for any degree at this university or elsewhere.

HAKATA JONATHAN ......................................................... ................................................
B1336501 Signature Date

Certified by
Mr. K. BASIRA ......................................................... ................................................
Supervisor. Signature Date

Mr. K. BASIRA ......................................................... ................................................
Head of Department. Signature Date
DEDICATION

I dedicate this piece of work to the Almighty God who by his grace and mercy equipped me with knowledge and strength to undertake this academic path successfully. I again dedicate this dissertation to my beloved wife Caroline and mother Mrs. J.E. Hakata.
ACKNOWLEDGEMENTS

I wish to express my deepest appreciation to my supervisor Mr K. Basira for his patience, support, and knowledge. The completion of this dissertation would not have been possible without his guidance throughout.

I would also like to acknowledge Dr. P. Chiurunge of Zimbabwe Centre For High Performance Computing for assisting me with statistical software plus appropriate computing facilities for data analysis. More gratitude goes to Mr. D. Murekachiro from the Faculty of Commerce and all the lecturers in the Department of Mathematics and Physics at Bindura University of Science Education for the immense academic knowledge imparted. I am forever indebted to all my colleagues for the experience and inspiration we shared together during this programme.
ABSTRACT

An attempt to fit a time series model that can be used to forecast patterns, trends and volatility of five stocks listed on the Zimbabwe Stock Exchange (ZSE) for the period 2009 to 2015 was made. Daily closing stock price data for selected companies trading on the ZSE was used. Using Python 3 and Compendium of statistics, tentative Generalized Additive Model (GAM) and Auto Regressive Integrated Moving Average (ARIMA) models were fitted. In order to assess the investment value at risk stock price volatility was modelled. The results show that the GAM model can be used for forecasting, since the forecasts exhibited excellent properties of being best linear unbiased estimates (blue) with least Mean Squared Error (MSE) compared to the traditional ARIMA (p, d,q) models. The most appropriate volatility model was found to be the GARCH (1,1) model which exhibited statistical significance based on its MSE at 0.05 significance level. In general, the GAM models outperformed their ARIMA counter parts and they are highly recommended in this research. It was also found that there is no one single GAM model for all the stocks, different stocks may require different GAM models. Further studies in this area should include more counters listed on the ZSE.

Keywords: Generalised Additive Models; Best Linear Unbiased Estimates; Value at Risk; Volatility; Forecasts.
TABLE OF CONTENTS

Contents
APPROVAL FORM ........................................................................................................... ii
DEDICATION .................................................................................................................... iii
ACKNOWLEDGEMENTS ............................................................................................... iv
ABSTRACT ....................................................................................................................... v
TABLE OF CONTENTS .................................................................................................... vi
LIST OF TABLES ........................................................................................................... ix
LIST OF FIGURES ......................................................................................................... ix
ACRONYMS .................................................................................................................... x
Chapter 1 ......................................................................................................................... 11
Introduction ...................................................................................................................... 11
  1.1 Background of the study ......................................................................................... 11
  1.2 Objectives of the study ......................................................................................... 11
  1.3 Research Questions .............................................................................................. 12
  1.4 Statement of the problem ..................................................................................... 12
  1.5 Justification of the study ..................................................................................... 12
  1.6 Assumptions ......................................................................................................... 13
  1.7 Limitations ............................................................................................................ 13
  1.8 Delimitations ........................................................................................................ 13
  1.9 Definition of key terms ....................................................................................... 14
  1.10 Summary .............................................................................................................. 14
Chapter 2 ......................................................................................................................... 15
Literature Review ........................................................................................................... 15
  2.1 Introduction .......................................................................................................... 15
  2.2 Empirical Literature Review ............................................................................... 15
  2.3 Theoretical Literature Review ............................................................................ 19
    2.3.1 Time series analysis ....................................................................................... 19
    2.3.2 Time series components ............................................................................... 19
    2.3.3 Assumptions of Time Series ......................................................................... 22
  2.4 Stationarity and Non-Stationarity of time series ............................................... 23
  2.5 Seasonal ARIMA Models .................................................................................... 27
  2.6 Generalized Additive Model (GAM) ................................................................... 28
Data Representation and Analysis

Chapter 3

Research Methodology

3.0 Introduction

3.1 Research Design

3.2 Research Instrument

3.3 Data collection

3.4 Box-Jenkins Approach Data Analysis Procedures

3.5 Development of Mathematical Model

3.6 Volatility forecasting

3.7 Volatility

3.8 Summary

Chapter 4

Data Representation and Analysis

4.0 Introduction

4.1 Uploading the data onto python 2

4.2 Zimbabwean Holidays

4.3 Model Identification and model fitting

4.4 Forecasting Stock Price Prediction Model

4.5 Automatic changepoint detection

4.6 Validating The Time Series Model

4.7 ARIMA Model

4.8 Decomposing the time series

4.9 The ARIMA Time Series Model

4.9.1 Fitting the ARIMA Time Series Model

4.9.2 Model Diagnostics

5.0 Volatility forecasting

5.1 Forecasting using the Facebook Prophet GAM Package

Summary
LIST OF TABLES
Table 4.1: Econet log-values of the stock data ................................................................. 41
Table 4.2: Zimbabwean public Holidays ........................................................................... 42
Table 4.4: An extract of the days to be forecasted in future .............................................. 43
Table 4.5: An extract of forecasts together with their upper and lower bounds .............. 43
Table 4.6: Time segments ................................................................................................. 49
Table 4.7: Examples of parameter combinations ............................................................... 53
Table 4.8: ARIMA models together with their respective AIC values .............................. 54
Table 4.9: Summary of results for the ARIMA model ....................................................... 54
Table 4.10: Percentage change, standard deviation, volatility and variance .................. 59
Table 4.11: Parameter values for the GARCH (1,1) model .............................................. 60
Table 4.12: Daily volatility forecasts obtained from the selected GARCH (1,1) model .... 60

LIST OF FIGURES
Figure 2.1: Trend (Australian Bureau of statistics, 2008) ................................................. 20
Figure 2.2: Cyclical (Australian Bureau of statistics, 2008) ........................................... 20
Figure 2.3: Seasonal (Australian Bureau of statistics, 2008) ......................................... 21
Figure 2.4: Irregular (Australian Bureau of statistics, 2008) ........................................... 22
Figure 4.1: Uploaded ZSE data ......................................................................................... 39
Figure 4.2: A combined plot of 7 different ZSE stocks ....................................................... 40
Figure 4.3: A plot of forecasts together with original data series-Econet ........................ 44
Figure 4.4: Automatic change point detection by Prophet ................................................. 45
Figure 4.5: Magnitude of rate of change for the detected potential change points ........ 46
Figure 4.6: Trend, Holidays, Weekly and Yearly Seasonality of Econet stocks ............ 47
Figure 4.7: Mean prediction error plot .............................................................................. 49
Figure 4.8: Smooth Error Trend plot ............................................................................... 50
Figure 4.9: Time series plot of Econet data ..................................................................... 51
Figure 4.10: Time series decomposition of Econet data ................................................. 52
Figure 4.11: Residual plots to diagnose the ARIMA model ............................................... 55
Figure 4.12: Model validation .......................................................................................... 56
Figure 4.13: Observed time series and the forecasts from 18/02/2015 – 17/02/2016 ....... 57
Figure 4.14: A plot of Standardized Residuals to check for the ARCH effect ............... 58
Figure 4.15: Model validation- a plot of actual volatility vs their forecasts ................... 61
**ACRONYMS**

<table>
<thead>
<tr>
<th>Abbr.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR</td>
<td>Auto Regressive</td>
</tr>
<tr>
<td>ARCH</td>
<td>Autoregressive Conditional Heteroskedasticity</td>
</tr>
<tr>
<td>ARIMA</td>
<td>Auto Regressive Integrated Moving Average</td>
</tr>
<tr>
<td>EGARCH</td>
<td>Exponential Generalized Autoregressive Conditional Heteroskedasticity</td>
</tr>
<tr>
<td>GAM</td>
<td>Generalized Additive Model</td>
</tr>
<tr>
<td>GARCH</td>
<td>Generalized Autoregressive Conditional Heteroskedasticity</td>
</tr>
<tr>
<td>TGARCH</td>
<td>Threshold Generalized Autoregressive Conditional Heteroskedasticity</td>
</tr>
<tr>
<td>ZSE</td>
<td>Zimbabwe Stock Exchange</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

1.1 BACKGROUND OF THE STUDY
The roles of a Stock Exchange are diverse and highly important in the economic development of a country. Stock markets are the places, where investors actually do business. Transactions involving trading of stocks are executed at the stock exchange through brokers, unless one is a member with that exchange, which enable him/her to trade directly (Gowda, 2015).

The Zimbabwe Stock Exchange (ZSE) is the country's official stock exchange in Zimbabwe whose origins date back to the arrival of the Pioneer Column in 1896 and today it is operated in accordance with the 1996's Zimbabwe Stock Exchange Act: Chapter 24:18. The Securities and Exchange Commission of Zimbabwe (SECZ) serves as the apex regulator of Zimbabwe’s capital markets and it operates in line with The Securities Act of Zimbabwe which was promulgated in 2004 and operationalized in 2008.

Following the decline of the Zimbabwean economy, hyperinflation caused the Zimbabwean dollar to be useless and the United States (US)-Dollar was adopted as the legal tender for trading on the exchange in February 2009. By March 2009, trade on the ZSE has been very slim, with few foreign investors willing to risk trading on the Zimbabwean market. Since then, an analysis of the ZSE Industrial performance from the available historical data has always been inevitable to enhance the knowledge of investors and other players in the stock market business.

1.2 OBJECTIVES OF THE STUDY

General objective
To investigate the performance of seven selected registered counters on the ZSE using time series models.
Specific objectives
The specific objectives of the study are:

1. To fit the best time series models describing stock data of selected counters
2. To forecast stock prices using a suitable time series models.
3. To compare and contrast the performance of GAM and ARIMA models in financial modeling
4. To model volatility of ZSE stock data using a suitable GARCH (1,1) model.

1.3 RESEARCH QUESTIONS
1. How does the ZSE industrial indices behave with time?
2. When will be the best time to sell or buy shares on the stock market?
3. What will be the closing stock price for such a company tomorrow?
4. What are the potential future losses of a portfolio of stocks?
5. What is the estimate of future volatility for a certain stock?

1.4 STATEMENT OF THE PROBLEM

1.5 JUSTIFICATION OF THE STUDY
According to The SECZ’s website (www.seczim.co.zw), their mission statement states, “To provide an optimal regulatory environment for the protection of investors and the sustainable development of capital markets for national economic growth.” This implies that a critical understanding of performance of various local industries on the ZSE is of great importance in Zimbabwe’s economic turnaround efforts. The study is aimed at giving a deep insight to local and foreign investors, stock brokers and custodians of ZSE to help them identify solutions to some or all of the questions mentioned in section 1.3 of this chapter.

Stock research is vital because by taking time to look over the financial history of the companies that one is thinking of investing in, the prospective investor will have a better sense of the future. While no one can be certain that such a stock will go up in value, taking the time to evaluate the
concerned company or group of industries’ historical data (growth) can give some intuition into the possibility (Butler, 2016). The study will give a descriptive and interpretation of the industrial performances on the ZSE basing on the available historical data. This will try to bring an insight to all players of the ZSE concerned on the possibility of making it despite the economic hardships that the nation is experiencing.

The study also aims to forecast on the future industrial indices which will help investors make wiser decisions when putting their hard-earned money into a stock and such a research have never been carried out in Zimbabwe. The study is also expected to expand the researcher’s understanding of time series analysis and appreciate its usefulness to financial markets. This will equip the researcher with skills and techniques of analysing data and also on the use of various the statistical packages.

1.6 ASSUMPTIONS
Technical analysis is based on two basic assumptions namely;

a) At any given time, the given stock prices of a company reflect everything that has or could affect the company that is the company's fundamentals, its broader economic factors along with market psychology are all priced into the stock.

b) Stock price movements are assumed to be stochastic by nature.

1.7 LIMITATIONS
In order to come up with a very comprehensive model there is need to use a larger sample size of say a ten-year period, but due to data availability and a switch from the inflated Zimbabwean Dollar to the multi-currency system, the study covers the period 18 February 2009- 19 February 2015.

1.8 DELIMITATIONS
1. Due to unavailability of financial resources, the study relies on secondary data obtained from ZSE offices in Harare.
2. The study only looks at Technical analysis of industrial performances at the ZSE, thus fundamental analysis was ignored.

1.9 DEFINITION OF KEY TERMS
Holden and Peman (1994) defined time series as a collection of observations or measurements on quantitative variables made sequentially or in a uniform set of times, usually daily, weekly, monthly, quarterly or annually. More so, Pandit and Wu (1983) views time series as a statistical series which tells us how data has been behaving in the past and gives value of the variable we are considering at various points in time. Time series examples include total monthly crimes for a period of ten years, daily stock prices of a firm for a period of one year, monthly electricity consumption for a household for a period of five years, etc. Time series are often used to project future values by observing how the value of a variable has changed in the past (Scott, 2003).

Time series analysis consist of methods or processes that break down a series into components and explainable portions that allow trends to be identified, estimates and forecasts to be made. Granger and Newbold (1974) assert that time series analysis attempts to understand the underlying context of the data points through the use of a model to forecast future values based on known past values. Such time series models include GAM, GARCH, EGARCH, TARCH, CGARCH and ARIMA, etc. The main focus of this study is based on GAM by Trevor Hastie and Robert Tibshirani, ARIMA model by Box-Jenkins (1976) as well as GARCH model by Bollerslev (1986). An Industrial Index is a stock index derived from the values of industrial stocks on a given stock exchange of interest.

According to a Business Dictionary online (www.businessdictionary.com), performance in the context of stock markets is defined as the behavior of a security or asset in the marketplace.

1.10 SUMMARY
The introductory chapter laid down the background of the study as well as justifications, objectives of the study and research questions to be answered. The importance of the chapter is that it draws a roadmap for the entire study and it defines the course of the study. The following chapter contains literature reviews on time series analysis and its applicability to the context under study. This is then followed by a discussion regarding research methodology. Chapter 4 gives an analysis and presentation of findings, chapter 5 highlighting final conclusions and recommendations.
CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION
This chapter provides theoretical and empirical literature review of previous researches on time series analysis of various statistical data. A framework for the case study is introduced and it comprises the main focus of the dissertation described herein. In trying to come up with a comparative approach on performance of autoregressive and generalized additive models in modelling stock prices in Zimbabwe, this research was conducted. (Adebiyi, Adewumi, & Ayo, 2014) pointed out that ARIMA models have shown efficient capability to generate accurate financial forecasts and constantly outperformed complex structural models in prediction power. On the other hand, (Berg, 2005) after cross validation, had the opinion that generalized additive models significantly outperform popular models at all levels of risk in financial modelling. This, seemingly “tug-of-war”, led the researcher to find out which case might it be with selected counters on the Zimbabwe Stock Exchange.

2.2 EMPIRICAL LITERATURE REVIEW
(Shakira, 2011), did a research to carry out a time series analysis of stock prices using the Box-Jenkins approach on eight selected companies namely Apple, Inc. (APPL), Microsoft Corp.(MSFT), Kroger Company (KR), Winn-Dixie Stores, Inc. (WINN), ASML Holding (ASML), Advanced Analogue Technologies, Inc. (AATI), PepsiCo, Inc. (PEP), and Coca-Cola Bottling Co. Consolidated(COKE). The closing prices for each stock going back ten years or from the date the company went public, whichever was reached first, were obtained from Yahoo! Finance. Time series plots of stock prices sampled weekly were obtained for each company and patterns were analysed. The results of these analyses allowed the researcher to determine which ARIMA model was most appropriate for each stock and decide if there were some similarities between industries. The results revealed that stocks do not behave in a certain way based on the industry they are in rather their behaviour has more do to with that particular company and how much their stock prices are influenced by factors that cannot be quantified.
(Devi, Sundar, & Alli, 2013) also, employed the time series analysis technique in their research work entitled “An Effective Time Series Analysis for Stock Trend Prediction Using ARIMA Model for Nifty Midcap-50. The data had been collected from National Stock Exchange of India (NSE)’s website. The historical data for the period of five years since 2007 to 2011 were taken into account for analysis. The Box-Jenkins methodology (Time series analyses tool), was used to identify the model. The Akaike and Bayesian information test criteria (AICBIC) was applied against the data to select the best the model. The best model equation was derived for all indices. The Mean Absolute Percentage Error (MAPE) also known as Mean Absolute Percentage Deviation (MAPD) accuracy was applied to determine the discrimination between the actual historical data and the forecast data. Basing on the minimum percentage error that was obtained from the performance measures, preceding forecasts for each index were made.”

Due to an increase in the number of people noticing the high risks and high turnover from stock markets nowadays, (Angadi & Kulkarni, 2015) carried out a research entitled, “Time Series Data Analysis for Stock Market Prediction using Data Mining Techniques with R”. In their findings they said, “A stock exchange market depicts savings and investments that are advantageous to increase the effectiveness of the national economy. The future stock returns have some predictive relationships with the publicly available information of present and historical stock market indices. Auto-Regressive Integrated Moving Average (ARIMA) is a statistical model which is known to be efficient for time series forecasting especially for short-term prediction.” In their paper, they proposed an ARIMA model for forecasting the stock market trends based on the technical analysis using historical stock market data. The selected model automated the process of finding the direction of future stock prices and aided financial specialists to choose the better timing for purchasing and/or selling of stocks. The obtained results revealed that the ARIMA model had a strong potential for short-term prediction of stock market trends.

(Xu, 2014) also carried out an investigation on Stock Price Forecasting Using information from Yahoo Finance and Google Trend for Apple Inc.(aapl). Plots of the raw and transformed data were obtained. The conventional ARMA time series analysis was applied on the weekly historical stock price data and it was noted that the transformed stock price data essentially followed a random walk process the ARIMA (0,1,0) model.
(Achia, Wangombe, & Anyika, 2007) studied the characteristics of the Nairobi Stock Exchange (NSE) index from January 1998 to March 2007. A graphical approach to identify possible patterns that existed in the NSE data was implemented. Basing on the plot, the authors identified that a steady growth in the amount of stock returns experienced late 2002 was due to the positive investor confidence perceived during an era of political stability brought about by the National Rainbow Coalition (NARC). In addition to these findings, the Auto Correlation Function (ACF) plots of the NSE index revealed that the stock returns were serially correlated and dependent. Summary statistics were then obtained from which it was noted that returns were highly volatile and that the mean continuously compounded return for the NSE was 0.019(±0.767). Basing on the AIC values obtained for AR (1), MA (1), ARMA (1,1), ARIMA (1,1,0), ARIMA (0,1,1) and ARIMA (1,1,1) models it was concluded that the ARIMA (1,1,1) model best fitted the daily returns on the NSE 20-share index and forecasts were then made using this most appropriate model. The GARCH (1,1) was also found to provide the best fit in modelling volatility and high volatilities were observed just before elections due to loss of investor confidence.

To illustrate that Generalized Additive Models have wider applications in day to day life, (Annalyn & Kenneth, 2017) carried out a research on analysing trends in Daylight Saving Time (DST) page views for people living in regions with four seasons. Data of page view counts of DST from 2008 to 2015 was extracted from Wikipedia database using Python and a GAM package called Prophet was used to conduct the time series analysis. After fitting a suitable GAM, overall growth, special events, and seasonal variation plots were attained. It was then observed that overall page views of the DST Wikipedia article were generally decreasing across the years. Weekly trends revealed that people were most likely to read about DST on Mondays, and least likely on weekends. And finally, annual trends showed that page views peak in end-March and end-October.

Peter Laurinec (2017) published an article entitled, “Doing magic and analyzing seasonal time series with GAM (Generalized Additive Model) in R. He illustrated the applicability of GAM in analysing electricity consumption using Microsoft R. After performing several data manipulations, two main seasonalities in plotted time series were obtained; weekly and daily. It was from these plots that peak load was noted to be at 3:00 pm daily and it gradually decreased towards weekends.
Residual plots for the model showed that the assumptions of normality were not violated at all. Forecasts for future electricity consumptions were then attained using the most appropriately fitted GAM model and in order to validate the model a plot of the fitted and original series was made. Basing on the findings, in his concluding remarks he says, “GAM models are smoother than Multilinear Regression (MLR)…”, (Laurinec, 2017).

In an effort to model and predict bankruptcy, Daniel Berg (2005) made a research entitled “Bankruptcy Prediction by Generalized Additive Models”. The data used was in the form of extensive collections of annual financial statements for all limited liability firms which were registered at the Norwegian register for business enterprises from 1996-2000. A two year default horizon; out-of-sample and out-of-time validation plot was made to find out the prediction power of Linear discriminant analysis (LDA), generalized linear models (GLM), single-hidden-layer neural networks (NN) and GAM models. It was then concluded that generalized additive models significantly outperformed them all (Berg, 2005).

(Bonga, 2014) published a journal entitled, “Empirical Analysis of Stock Returns and Volatility of the Zimbabwean Stock Markets” in which he tried to investigate the time series behavior of stock returns for Zimbabwe stock market. Testing the relationship between stock returns and unexpected volatility, it was evident that the Industrial Stock Market has significant results whilst the Mining Stock Market shows a different behavior. More so it was noted that when volatility increases, risk increases and so investor confidence is reduced thereby enhancing a decrease in stock prices. Basing on the imperial results, he concluded that the class of generalized autoregressive conditional heteroscedastic (GARCH) models has proved particularly valuable in modelling time series with time varying volatility.

In trying to explore the comparative ability of different statistical and econometric volatility forecasting models in the context of Zimbabwe stock market, Murekachiro D (2016) wrote a paper entitled, ‘Time Series Volatility Forecasting of the Zimbabwean Stock Exchange.’ Having compared the performance of GARCH (1,1) and EGARCH (1,1) for the period 19February 2009 to 31 December 2014, it was found out that ZSE data showed a significant departure from normality and existence of conditional heteroskedasticity in the residuals series. In addition to his
findings, EGARCH (1,1) yielded better results than GARCH (1,1) despite his mentioning that several authors had found out that GARCH (1,1) models outperform other volatility models and among them are Erdington and Guan (2004) and Doidge and Wei (1998), (Murekachiro, 2016).

2.3 THEORETICAL LITERATURE REVIEW

2.3.1 Time series analysis

Time series is defined as an ordered sequence of values of a variable at equally spaced time intervals (Mary & Zey, 2012). (George, Gwilym, & Gregory, 2008) Defined time series as a sequence of observations taken sequentially in time. The series may be denoted by $X_1, X_2, ..., X_t$ where $t$ refers to the time period and $X$ refers to the value of the variable (NCSS, 2013). Time series is a sequence of observations ordered in time (Ramasubramanian, 2015). For example, a monthly sequence of the quantity of goods shipped from factory, a weekly series of the number of road accidents and hourly observations made on the yield of a chemical process (Ansah, 2014). When there is only one variable upon which observations are made then we call that a single time series or more specifically a univariate time series (Ramasubramanian, 2015). Time series can be applied in fields like economics, business, engineering, natural sciences and social sciences (Ansah, 2014).

2.3.2 Time series components

To select an appropriate forecasting method there is need for considering the types of data patterns obtained from time plots, so that the most appropriate methods could be applied. The main types of time series data patterns are trend (T), cyclical (C), seasonal (S), horizontal (H), and irregular (I).

**Trend (T)**

According to The Australian Bureau of Statistics, (2008), “Trend is defined as the ‘long term’ movement in a time series without calendar related and irregular effects, and is a reflection of the underlying level.” Time series data may show upward or downward trend for a period of years and this may be due to factors like increase in population, change in technological progress, large scale shift in consumers demands etc. (Blog, 2008).
An increasing trend in the Australian monthly electricity production from March 1956 to August 1995 and a decreasing trend in the U.S. Treasury bill contracts on the Chicago market for 100 consecutive trading days in 1981 are illustrated below;

![Trend](image)

**Figure 2.1: Trend (Australian Bureau of Statistics, 2008)**

**Cyclical (C)**

(Makridakis, Wheelwright, & Hyndman, 1998), noted that a cyclical pattern exists when the data exhibit rises and cyclical falls that are not of a fixed period. The Australian clay brick production shown in Figure 2.2 below shows cycles of several years in addition to the quarterly seasonal pattern.

![Cyclical](image)

**Figure 2.2: Cyclical (Australian Bureau of Statistics, 2008)**
Seasonal (S)

(Makridakis, Wheelwright, & Hyndman, 1998) said, “a seasonal pattern exists when a series is influenced by seasonal factors (e.g., the quarter of the year, the month, or day of the week). The beer data show seasonality with a peak in production in November and December (in preparation for Christmas) each year.

![Monthly Australian beer production](image.png)

**Figure 2.3: Seasonal (Australian Bureau of statistics, 2008)**

Horizontal (H)

(Makridakis, Wheelwright, & Hyndman, 1998) explained that this type of pattern exists when the data values fluctuate horizontally around a constant mean and such a series is called "stationary" in its mean.

Irregular (I)

Irregular component (also known as the residuals) is an unpredictable component which remains after the seasonal and trend components of a time series have been estimated and removed (Australian Bureau of statistics, 2008). Irregular variations are fluctuations in time series that are short in duration, erratic in nature and follow no regularity in the occurrence pattern (Blog, 2008).

Irregular fluctuations are due to occurrence of unforeseen events like floods, earthquakes, wars, families etc. (Blog, 2008).
2.3.3 Assumptions of Time Series

2.3.3.1 Assumption of Stationarity

In many time series techniques, the most common assumption is that the data are stationary. A stationary series $x_t$ have a property that the mean $E(x_t)$, variance of $x_t$ and the autocorrelation (and also covariance) between $x_t$ and $x_{t-h}$ remain constant over time. One simple test of stationarity is based on the so-called autocorrelation function (ACF) and a lag-$h$ autocorrelation is given by:

$$\frac{\text{Covariance} (x_t, x_{t-h})}{\text{Variance} (x_t)} \cdots \cdots 0.1 \text{ (Ruey S. T., 2005)}$$

A unit-root test may be conducted with the aim of discerning difference stationary from trend stationarity (J.D.Cryer & Chan, 2008). Also for checking stationarity statistically, we use the unit root test together with the help of Augmented Dickey Fuller test; we check stationarity in the level first including an intercept in the equation (Seshanwita & Das, 2012).

2.3.3.2 Assumption of Independence

![Figure 2. 4: Irregular (Australian Bureau of Statistics, 2008)](image-url)
The ACF of the residuals of the fitted model is computed and a correlation analysis is going to be carried out between them. A good model has its residual independent (uncorrelated) and residuals are normally distributed. Randomness (independence) in the residuals can be tested in several ways.

Box and Pierce (1970) propose the portmanteau statistic which is given by

\[ Q^* (m) = T \sum_{\ell} \hat{p}^2_{\ell} \ldots \ldots \ldots \ldots 0.2 \quad (\text{Ruey S.}, 2010) \]

As a test statistic for the null hypothesis \( H_0 \) and alternative hypothesis \( H_1 \) which is equal to zero and not equal to zero respectively (Ruey S., 2010).

### 2.3.3.3 Assumption of Normality

There are both graphical and statistical methods for evaluating normality. Graphical test for normality can be performed by constructing histograms and or normal scores or quantile-quantile plots (QQ). Statistical methods to check for normality include Shapiro-wilk Normality test and Kolmogorov-smirnov (1974) test (G.Kleinbaum, L.Kupper, Mizmam, & E.Muller, 2008).

### 2.3.3.4 Assumption of Constant variance

Plotting the residuals over time if the model is adequate, we expect the plot to suggest a rectangular scatter around a zero-horizontal level with no trends whatsoever (K.Mutsonziwa, L.Dhliwayo, & C.Chimedzo, 2004).

### 2.4 STATIONARITY AND NON-STATIONARITY OF TIME SERIES

A series is said to be stationary if its mean, variance and covariance remain constant over time (Seshanwita & Das, 2012). The stationarity of a time series is related to its statistical properties in time (C.Montgomery, L.Jennings, & Kulahci, 2008). Stationary time series models assume that the process remains in statistical equilibrium with probabilistic properties that do not change over
time, in particular varying about a fixed constant mean level and with constant variance (George, Gwilym, & Gregory, 2008).

A stationary time series exhibits similar “statistical behavior” in time and this is often characterized as a constant probability distribution in time. In particular, such a time series fluctuates around a fixed mean. (C.Montgomery, L.Jennings, & Kulahci, 2008).

“A time series \( \{ r_t \} \) is said to be strictly stationary if the joint distribution of \( (r_{t1},...,r_{tk}) \) is identical to that of \( (r_{t1+t},...,r_{tk+t}) \) for all where k is an arbitrary positive integer and \( (t_{1},...,t_{tk}) \) is a collection of k positive integers. A time series \( \{ r_t \} \) is weakly stationary if both the mean of \( r_t \) and covariance between \( r_t \) and \( r_{t-l} \) are time invariant, where \( l \) is an arbitrary integer. However, if the time series \( r_t \) is normally distributed then weak stationary is equivalent to strict stationary” (Ruey S. , 2010).

Any time series without a constant mean over time is nonstationary (J.D.Cryer & Chan, 2008). For a price series, the non-stationarity is mainly because there is no fixed level for the price. The stochastic model for which the exponentially weighted moving average forecast yields minimum mean square error is a member of a class of non-stationary process called autoregressive integrated moving average (ARIMA) process (George, Gwilym, & Gregory, 2008).

### 2.4.1 The Moving Average (MA)

Moving Averages models were first considered by Slutsky (1927) and Wold (1938) as cited by (J.D.Cryer & Chan, 2008). The moving average is free from seasonal effect and will have randomness to certain extent (MA=TC) (Pannerselvam, 2005).

\[
    r_t = c_0 + a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \cdots - \theta_q a_{t-q} \ldots \ldots 2.3
\]

Where \( q>0 \), \( c_o \) is a constant, \( \theta_1..\theta_q \) are parameters and \( \{ a_i \} \) is a white noise series and we call such a series a moving average of order \( q \) and abbreviate the name to MA (q) (Ruey S. T., 2005).
Autoregressive Models (AR) and Partial Autocorrelation Function (PACF)

A stochastic model that can be extremely useful in the representation of certain practically occurring series is the autoregressive model (George, Gwilym, & Gregory, 2008).

\[ r_t = \phi_0 + \phi_1 r_{t-1} + \cdots + \phi_p r_{t-p} + a_t \] \hspace{1cm} 2.4

Is the autoregressive (AR) process of order \( p \), denoted AR(p), where \( p \) is a nonnegative integer, \( \phi_0 \) is the constant term, \( \phi_i \) are coefficients of \( r_{t-i} \) and \( \{a_t\} \) is assumed to be a white noise series. This model says that the past \( p \) values \( r_{t-i} \) \((i = 1, \ldots, p)\) jointly determine the conditional expectation of \( r_t \) given the past data. There are two general approaches available for determining the value of \( p \) in AR model; these are Partial Autocorrelation Function (PACF) and information criteria (Ruey S., 2010).

The PACF of a stationary time series is a function of its ACF and is a useful tool for determining the order of \( p \) of an AR model (Ruey S., 2010).

Information Criteria and Akaike Information Criteria (AIC)

There are several information criteria available to determine the order \( p \) of an AR process, these are Akaike information criteria (AIC) and Schwarz –Bayesian information criterion (Ruey S., 2010).

AIC is defined as (Akaike 1973)

\[ \text{AIC} = \frac{-2}{T} \ln(\text{likelihood}) + \frac{2}{T} (\text{number of parameters}) \]

Where the likelihood function is evaluated at the maximum likelihood estimates and \( T \) is the sample size (Ruey S., 2010).

For a Gaussian AR (\( \ell \)) model, AIC reduces to

\[ \text{AIC}(\ell) = \ln(\bar{\delta}_a^2) + 2 \frac{\ell}{T} \] \hspace{1cm} 0.1

Where \( \bar{\delta}_a^2 \) is the maximum-likelihood estimate of \( \delta_a^2 \), which is the variance of \( a_t \), \( T \) is the sample size (Ruey S., 2010).

Schwarz-Bayesian Information Function (BIC)
For a Gaussian AR (ℓ) model, the criterion is
\[
\text{BIC}(\ell) = \ln(\delta_\ell^2) + \ell \frac{\ln T}{T} \quad \text{......... 0.2}
\]
Thus, compared with AIC, BIC tends to select a lower AR model when the sample size is moderate or large (Ruey S., 2010).

**Selection rule**
To use AIC to select an AR model in practice, one computes AIC (ℓ) for ℓ = 0… p, where p is prespecified positive integer and selects the order and that has the minimum AIC. The same rule applies to BIC (Ruey S., 2010).

### 2.4.2 The Mixed Autoregressive-Moving Average (ARMA)
This is a combination of both the (MAq) and (ARp) process. In general, an ARMA (p, q) model is given as
\[
\begin{align*}
    r_t &= \phi_0 + \phi_1 r_{t-1} + \phi_2 r_{t-2} + \cdots + \phi_p r_{t-p} + a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \cdots \\
    &= \phi_0 + \sum_{i=1}^{p} \phi_i r_{t-i} + \sum_{i=1}^{q} \theta_i a_{t-i} \quad \text{......... 0.3}
\end{align*}
\]
Where \(\{a_t\}\) is a white noise process (C.Montgomery, L.Jennings, & Kulahci, 2008). We say \(\{r_t\}\) is a mixed autoregressive moving average process of order p and q respectively; we abbreviated the name to ARMA(p,q) (J.D.Cryer & Chan, 2008).

**White Noise**
According to Ruey S (2010), a time series \(r_t\) is called a white noise if \(\{r_t\}\) is a sequence of independent and identically distributed random variables with finite mean and variance. In particular, if \(r_t\) is normally distributed with mean zero and variance \(\sigma^2\), the series is called a Gaussian white noise. For a white noise series, all the ACFs are zero. In practice, if all sample ACFs are close to zero, then the series is a white noise series. A very important example of a stationary process is the so-called white noise process and the ACF and Ljuing-Box statistics of residual can be used to check the closeness of a white noise \(\{a_t\}\).
ARIMA MODELS

A time series \( \{Y_t\} \) is said to follow an integrated autoregressive moving average model if the \( d^{th} \) difference

\[
W_t = \nabla^d Y_t \quad \text{............... 0.4}
\]
is a stationary ARMA process (J.D.Cryer & Chan, 2008). ARIMA model is said to be a unit-root nonstationary because its AR polynomial has a unit-root and a conventional approach for handling unit-root nonstationary is to use differencing (Ruey S., 2010).

\[
Y_t = (1 + \phi_1)Y_{t-1} + (\phi_1 - \phi_2)Y_{t-2} + \cdots + (\phi_p - \phi_{p-1})Y_{t-p-1} + \epsilon_t - \theta_1\epsilon_{t-1} - \cdots - \theta_q\epsilon_{t-q} \\
\text{............... 0.5}
\]

We call this the difference equation form of the model (J.D.Cryer & Chan, 2008).

2.5 SEASONAL ARIMA MODELS

According to Ruey S. (2010), some financial time series such as quarterly earnings per share of a company exhibits certain cyclical or periodic behavior; such a time series is called a seasonal time series. More so, seasonality in time series is of secondary importance and can be removed from the data to make inference by a procedure known as seasonal adjustment. The seasonal ARIMA i.e. ARIMA (p, d, q) * (PDQ)S model is defined by

\[
\phi_q(B)\phi_p(B^s)\nabla^d \nabla_s^D Y_t = \theta_Q(B^s)\theta_q(B)\epsilon_t \\
\text{............... 0.6}
\]

Where
\[
\phi_p(B) = 1 - \phi_1B - \cdots - \phi_pB^p, \\
\text{............... 0.7}
\]
\[
\theta_q(B) = 1 - \theta_1B - \cdots - \theta_qB^q, \\
\text{............... 0.8}
\]
\[
\phi_p(B^s) = 1 - \phi_pB^s - \cdots - \phi_pB^{sp}, \\
\text{............... 0.9}
\]
\[
\theta_Q(B^s) = 1 - \theta_1B^s - \cdots - \theta_qB^{sq}, \\
\text{............... 1.0}
\]

B is the backshift operator (i.e. \( BY_t = Y_{t-1} \), \( B^2Y_t = Y_{t-2} \) and so on), ‘s’ the seasonal lag and ‘\( \epsilon_t \)’ a sequence of independent normal error variables with mean zero and variance \( \delta^2 \) (Ramasubramanian, 2015).

Using the multiplicative seasonal ARIMA (SARIMA) model we have the general notation (p, d, q) X (PDQ) s where p, d, q is the non-seasonal part and P,D,Q the seasonal part with p, d, q having
their usual meaning and $P$ is the order of the seasonal AR process, $D$ the differencing of the seasonal process, $Q$ the order of the seasonal MA process of the time series and $s$ is the order of seasonality (Ansah, 2014).

### 2.6 GENERALIZED ADDITIVE MODEL (GAM)

Besides using correlations between values from similar time points, we could model overall trends. A time series could be seen as a summation of individual trends. Financial Markets data could well be modeled by adding a seasonal trend to an increasing growth trend, in what’s called a generalized additive model (GAM) (Annalyn & Kenneth, 2017).

The principle behind GAMs is similar to that of regression, except that instead of summing effects of individual predictors, GAMs are a sum of smooth functions. Functions allow us to model more complex patterns, and they can be averaged to obtain smoothed curves that are more generalizable. Because GAMs are based on functions rather than variables, they are not restricted by the linearity assumption in regression that requires predictor and outcome variables to move in a straight line. Furthermore, unlike in neural networks, we can isolate and study effects of individual functions in a GAM on resulting predictions (Wood, Goude, & Shaw, 2015).

A generalized additive model (GAM) in statistics is a generalized linear model in which the linear predictor depends linearly on unknown smooth functions of some predictor variables, and interest focuses on inference about these smooth functions. GAMs were originally developed by Trevor Hastie and Robert Tibshirani to blend properties of generalized linear models with additive models (Hastie & Tibshirani, 1990).

The model relates a univariate response variable, $Y$, to some predictor variables, $x_i$. An exponential family distribution is specified for $Y$ (for example normal, binomial or Poisson distributions) along with a link function $g$ (for example the identity or log functions) relating the expected value of $Y$ to the predictor variables via a structure such as:

$$ g(E(y_i)) = \beta_0 + f_1(x_{i1}) + \cdots + f_p(x_{ip}) + \epsilon_i \quad 2.15 $$
\( y_i \sim \) is some exponential family distribution, \( g \) is a link function (identical, logarithmic or inverse), \( y \) is a response variable, \( x_1, \ldots, x_p \) are independent variables, \( \beta_0 \) is an intercept, \( f_1, \ldots, f_p \) are unknown smooth functions and \( \varepsilon \) is an independent and identically distributed random error. The functions \( f_i \) may be functions with a specified parametric form (for example a polynomial, or a spline depending on the levels of a factor variable) or may be specified non-parametrically, or semi-parametrically, simply as 'smooth functions', to be estimated by non-parametric means. So, a typical GAM might use a scatterplot smoothing function, such as a locally weighted mean, for \( f_1(x_1) \), and then use a factor model for \( f_2(x_2) \). This flexibility to allow non-parametric fits with relaxed assumptions on the actual relationship between response and predictor, provides the potential for better fits to data than purely parametric models, but arguably with some loss of interpretability. The smooth function \( f \) is composed by sum of basis functions \( b \) and their corresponding regression coefficients \( \beta \), formally written:

\[
  f(x) = \sum_{i=1}^{q} b_i(x) \beta_i
\]

where \( q \) is a basis dimension.

To find the best trend line that fits the data, GAM uses a procedure known as backfitting. Backfitting is a process that makes fine adjustments to the functions in a GAM iteratively so that they produce a trend line that minimizes prediction errors (Annalyn & Kenneth, 2017).

### 2.7 Forecasting Using the Facebook Prophet GAM Package

Prophet is a procedure for forecasting time series data. It is based on an additive model where non-linear trends are fit with yearly and weekly seasonality, plus holidays. It works best with daily periodicity data with at least one year of historical data. Prophet is robust to missing data, shifts in the trend, and large outliers (Taylor & Letham, 2017).

The Prophet package allows us to specify different types of functions comprising of the resulting GAM trend. There are three main types of functions:

- **Overall Growth.** This can be modeled either as a straight (linear) or slightly curved (logistic) trend. In this analysis, the default linear growth model shall be used.

- **Seasonal Variations.** This is modeled using Fourier series, which is simply a way to approximate periodic functions. The exact functions are derived using a process known as backfitting. We can
specify if we anticipate weekly or/and annual trends to be present. In this analysis, we include both a weekly trend since brokers trade during weekdays, while a yearly trend might coincide with the usual political risk.

**Special Events.** Besides modeling regular trends, there is also need to account for one-off events. This includes any phenomenon, be it policy announcements or natural disasters, that would add ripples to an otherwise smooth trend such as elections, announcements of bond notes, downgrading of Zimbabwe credit rating. If we do not account for irregular events, the GAM might mistake them to be persistent occurrences and their effects would be erroneously propagated. In our analysis, special events included exact dates when we have the **Zimbabwean Public Holidays**. We can also specify windows before and after each event to account for their ripple effect (Annalyn & Kenneth, 2017).

**2.8 VOLATILITY**
According to Investopedia, volatility is a statistical measure of the dispersion of returns for a given security or market index. It can either be measured by using the standard deviation or variance between returns from that same security or market index. Commonly, the higher the volatility, the riskier the security (Orabi & Alqurran, 2015).

Stock price volatility is an indicator that is commonly used by investors to observe changes in trends in the market pace. In the options markets, the two main types of stock volatility are historical and implied volatility. The former is also referred to as actual or realized volatility and is a measure of the stock price’s movement relative to its historical data. More specifically, historical volatility is measured by taking the daily percentage price changes in a stock and calculating the mean over a specified period (Radtke, 2014).

Implied volatility is the current volatility of a stock and is estimated by its option price. In order to come up with the implied volatility, investors use an options pricing model that takes into account strike price, the expiration date, the current stock price and stock dividends that would have been paid by the stock (Radtke, 2014).
According to Ruey S. (2010), some statistical methods and econometric models available for modelling the volatility of an asset return are referred to as conditional heteroscedastic models. In this research work only the autoregressive conditional heteroscedastic (ARCH) model of Engle (1982) and the generalized ARCH (GARCH) model of Bollerslev (1986) will be employed.

Model building
Ruey S. mentioned that the exercise of building a volatility model for any return series consists of four steps:

a) Specify a mean equation by testing for serial dependence in the data and, if necessary, building an econometric model (e.g., an ARMA model) for the return series to remove any linear dependence.

b) Use the residuals of the mean equation to test for ARCH effects.

c) Specify a volatility model if ARCH effects are statistically significant and perform a joint estimation of the mean and volatility equations.

d) Check the fitted model carefully and refine it if necessary.

The autoregressive heteroscedastic (ARCH) model
According to Ruey S. (2010), The ARCH model of Engle (1982) is the first model that provides a systematic framework for volatility modeling. The basic idea of ARCH models is that (a) the shock \( a_t \) of an asset return is serially uncorrelated, but dependent, and (b) the dependence of \( a_t \) can be described by a simple quadratic function of its lagged values. Specifically, an ARCH(\( m \)) model assumes that

\[
a_t = \sigma_t \epsilon_t, \quad \sigma_t^2 = \omega + \alpha_1 a_{t-1}^2 + \cdots + \alpha_m a_{t-m}^2 \quad \text{......... 2.16}
\]

where \( \{\epsilon_t\} \) is a sequence of independent and identically distributed random variables with mean zero and variance 1, \( \omega \) and \( \alpha_i \geq 0 \) for \( i \geq 0 \). The coefficient \( \omega \) is the weighted long-return variance and coefficients \( \alpha_i \) must satisfy some regularity conditions to ensure that the unconditional variance of \( a_t \) is finite.
The ARCH effect and the corresponding order could be clearly observed from standardized residual plots and if ‘big spikes’ are present in the plot this would suggest that the data are not serially independent and so do have some ARCH effect.

**The generalized autoregressive heteroscedastic (GARCH) model**

Even though the ARCH model is simple, it requires several parameters to sufficiently describe the volatility of the data under analysis. Alternatively, the generalized ARCH (GARCH) model, profounded by Bollerslev (1986) which is merely an extension of the ARCH model could be used instead. For a log return series $r_t$, let $a_t = r_t - \mu_t$ be the innovation at time $t$. Then $a_t$ follows a GARCH ($m, s$) model if

$$a_t = \sigma_t \epsilon_t , \quad \sigma_t^2 = \omega + \sum_{i=1}^{m} \alpha_i r_{t-i}^2 + \sum_{j=1}^{s} \beta_j \sigma_{t-j}^2, \ldots \quad 2.17$$

Where again $\{\epsilon_t\}$ is a sequence of identically and independent random variables with mean 0 and variance 1, $\omega$ and $\alpha_i \geq 0$ for $\beta_j \geq 0$, and $\sum_{i=1}^{\max(m,s)} (\alpha_i + \beta_i) < 1$. Equation 2.17 above reduces to a pure ARCH$(m)$ model for $s = 0$ and the $\alpha_i$ and $\beta_j$ are the ARCH and GARCH parameters respectively (Ruey S. T., 2005).

(Lachowicz, 2013) further explains that since the form of the GARCH $(m, s)$ model above leaves us with a wry face as we know what it really means to obtain all the coefficients $\alpha_i$ and $\beta_j$ in real life, a some sort of simplification that has been given a wide applause in financial hassle is the GARCH(1,1) model:

$$\sigma_t^2 = \omega + \alpha r_{t-1}^2 + \beta \sigma_{t-1}^2, \ldots \quad 2.18$$

Which derives its value based on the most recent update of $r$ and $\sigma$ and so GARCH (1,1) should provide us with a good forecast of volatility when the series of past returns were similar which is not achievable in stock data which is noisy. To obtain the accurate values of $\omega, \alpha$ and $\beta$ the maximum likelihood method is used.
2.9 SUMMARY
A detailed theoretical and empirical literature review have been laid, from which a need and possibility for this dissertation have been established. (Murekachiro, 2016) in his concluding remarks mentioned that further studies should be carried out to come up with financial models that could accurately model stocks together with their volatility to enhance investor appetite. More so literature reviewed showed that up to this day only autoregressive models have been applied to analyse stock trends in Zimbabwe and no literature have employed generalised additive models yet they and neural networks have proven to be widely useful (Wood, Goude, & Shaw, 2015). This research is therefore inevitable in trying to further establish which model best fits stock trends irrespective of the widely accepted traditional autoregressive models. The chapter gave a review of the information that will later be used as a basis for evaluating the performance of selected counters on the Zimbabwe Stock Exchange. Stock price and volatility forecasts will be obtained using the most appropriate models and comparisons made. The forthcoming chapter will give the research methodology.
CHAPTER 3

RESEARCH METHODOLOGY

3.0 INTRODUCTION
According to (Dipak Kumar, 2003) research methodology is a scientific and systematic way to solve research problems. This section clarifies the research design, data collection procedures, instruments to be used and how the data shall be analyzed. It also looks at the development of the time series model which is to be used for forecasting.

3.1 RESEARCH DESIGN
According to Kerlinger as cited by (Bhattachayya & Kuma, 2006), “Research design is the plan, structure and strategy of investigation conceived so as to obtain answers to research questions and to control variance. This research is going to use a descriptive research design. (Panneerselvam.R, 2006), defines descriptive studies as a research that tries to describe the characteristics of the respondents in relation to a particular product or a practice or culture of importance. The emphasis was on researching performance of various players on the ZSE in order to observe their behavior with time and possibly give the underlying factors enhancing these behaviors. This is a secondary data based research where secondary data was used as the main source to examine stock price trends from which suitable time series models were derived and precise forecasts were made.

3.2 RESEARCH INSTRUMENT
The data collected was analyzed using a statistical package called Python 3. Various textbooks and internet were used to provide basic information, literature review, theorems and proofs of mathematics formulas. The internet was used to access previous work done on time series analysis and it has the advantage that it provides wide range and up-to-date information.

3.3 DATA COLLECTION
The data used in this research was collected from ZSE Head Offices in Harare and from the ZSE website. The statistics were recorded over a period of seven years that is since dollarization.
Secondary Data

Secondary data is when an investigator uses the data which has already been collected by others according to (Pannerselvam, 2005), e.g. government publications, foreign government publications etc. The researcher used secondary data from The Zimbabwe Stock Exchange.

3.4 Box-Jenkins Approach Data Analysis Procedures

Box-Jenkins analysis refers to a systematic method of identifying, fitting, checking and using integrated autoregressive, moving average (ARIMA) time series models (NCSS, 2013). One of the steps in the Box - Jenkins method is to transform a non-stationary series into a stationary one (NCSS, 2013).

Box-Jenkins uses a three-step iterative approach of model identification, parameter estimation and diagnostic checking to determine the best parsimonious model from a general class of ARIMA models. This three-step process is repeated several times until a satisfactory model is finally selected. Then this model can be used for forecasting future values of the time series as cited by (Ratnadip, 2009).

3.5.1 Model Specification/Identification

The objective of the model identification step is to select values of $d$ and then $p$ and $q$ in the ARIMA ($p$, $d$, and $q$) model (NCSS, 2013). In the process of model selection; we shall attempt to use the principle of parsimony, that is the model used should require the smallest number of parameters that will adequately represents the time series (J.D.Cryer & Chan, 2008).

As cited by (J.D.Cryer & Chan, 2008), Albert Einstein is quoted in Parzen (1982.p68) as remarking that “everything should be made as simple as possible but not simpler”.

There are different approaches used for determining the order of $p$ for AR models. The first approach is to use the PACF and the second approach is the use of some information criteria function (Ruey S., 2010).

The researcher will use a "grid search" in Python to iteratively explore different combinations of parameters $p$, $d$ and $q$. For each combination of parameters, we will fit a new seasonal ARIMA model with the SARIMAX () function from the statsmodels module and assess its overall quality.
by (J.D.Cryer & Chan, 2008). And in the case of identifying and specifying the most appropriate GAM model, the researcher will use the Prophet package in Python as suggested by the Facebook Team Sean J. Taylor and Benjamin Letham (2017).

3.5.2 Model Fitting
(Hogg, Bovy, & Lang, 2010) Model fitting is a process that has three stages parameter estimation, obtaining a predicted data set and finally obtaining an error analysis to assess the fitness of the model. The term parameter estimation refers to the process of using sample data to estimate the parameters of the selected distribution. Several parameter estimation methods are available, these includes relatively simple method of probability plotting, least of squares, maximum likelihood and Bayesian estimation method (parameter estimation, 2015). As recommended by Murekachiro D. (2016) in his findings, the researcher is going to use the method of maximum likelihood to estimate parameters using SARIMAX (), a function in Python. In this research, the Mean Square Error will be used for error analysis. In the case of fitting the GAM model, the researcher will consider suggestions by Sean J. Taylor and Benjamin Letham (2017) using Python 3.

3.5.3 Model Diagnostic
A fitted model must be examined carefully to check for possible model inadequacy (Ruey S. , 2010). Model adequacy is checked by accessing whether the model assumptions are satisfied. If the model is adequate, then the residual should behave like a white noise (Ruey S. , 2010). The basic assumption is that \( \{e_t\} \) are white noise, that is they possess the properties of independence, identically and normally distributed random variables with zero mean and constant variance \( \delta^2 \).

Model diagnostics or model criticism is concerned with testing the goodness of fit of a model and, if the fit is poor, suggesting appropriate modifications (J.D.Cryer & Chan, 2008). A commonly used statistic to measure goodness of fit of a stationary series is the R square (R\(^2\)) defined as

\[
R^2 = 1 - \frac{\text{residual sum of squares}}{\text{total sum of squares}} \quad ............0.1 \quad \text{(Ruey S. , 2010)}
\]

The researcher is going to use the plot_diagnostics object in Python 3 which allows us to quickly generate model diagnostics and investigate for any unusual behavior. If the residual plots exhibit a standard normal distribution, i.e. if the histogram plot has a “bell” like shape, if the quantile-
quantile (QQ) plot follows a linear trend and if the autocorrelation function shows that the residuals are not correlated then we proceed to the next phase, otherwise we start over (Annalyn & Kenneth, 2017).

3.5 DEVELOPMENT OF MATHEMATICAL MODEL
The time series GAM and ARIMA models together with GARCH (1,1) for volatility were used to analyze the ZSE data. (Ocaya.Bruno, Ruranga.Charles, & Kaberuka.William, 2013) Cited that, the time series analysis of the study involves plots of series to check whether there was a trend or not, determine the maximum lag of the model and test for stationarity.

A series is said to be stationary if its mean, variance and covariance remain constant overtime (Seshanwita & Das, 2012). In the case of ZSE data, the researcher will use log values of the stock prices to remove the unequal variances in order to achieve stationarity as required in the time series analysis. A GAM package called Prophet will be used to determine the model and to come up with the best GAM model, the researcher will extract and analyze all the holidays separately. This is so because holidays tend to distort stock prices, in the sense that the appetite for trading is generally high just before the holiday and tend to stabilize soon after hence may render the model inaccurate if not taken into account.

The SARIMAX function in Python will used to tentatively determine the model parameters after performing several iterations. The model whose parameters would have been determined is then fitted to the data and later validated to asses its accuracy.

3.6 FORECASTING
Forecasting or predicting future, yet unobserved, values are one of the main reasons for developing time series models. It is one of the objectives of building a model for a time series so that we will be able to forecast future values for that series at future times (J.D.Cryer & Chan, 2008).

Once a model has been selected accurately and its parameters estimated appropriately, the model is used to make forecast and the users of the forecasts will be evaluating the pros and cons of the model as time progresses (Makridakis.Hyndman.Wheelwright, 2007). In this regard, the
researcher will obtain forecasts for the ZSE data using the obtained GAM, ARIMA and GARCH models.

3.7 VOLATILITY
In this section of the research, the researcher will use Python 3 to illustrate how one can model stock price volatility for a given counter. The researcher will demonstrate this, using Econet data for the period under study and the same can be performed on any stock market counter active on the ZSE. To achieve this, the following steps are to be followed;

a) Check for the ARCH effect by obtaining a plot of the standardised residuals and if spikes are present, that will be evident enough for volatility.
b) Obtain the percentage changes of the stock prizes.
c) Obtain the standard deviation a period of say 21 days.
d) Obtain the variance between stock prices.
e) Obtain the historical/ observed volatility for the period selected in (c) above.
f) Obtain the stock returns
g) Fit an ARCH model to the returns
h) Obtain the parameter values omega, alpha and beta for the ARCH model.

3.8 SUMMARY
Following suggestions made by researchers reviewed in chapter 2, the methodology which shall be used to carry out the research was laid. The research design and the data sources used in the research were also highlighted. The chapter went further to look at the data analysis procedures to be carried out in the next chapter. The next chapter will now discuss the representation and analysis of the data. This is followed by a discussion regarding final conclusions and recommendations basing on the findings.
CHAPTER 4

DATA REPRESENTATION AND ANALYSIS

4.0 INTRODUCTION
This chapter focuses on data presentation, discussion and interpretation of the research findings. The chapter illustrates a step by step process of data analysis using a GAM package called Prophet in Python published by various Facebook researchers.

4.1 UPLOADING THE DATA ONTO PYTHON 2
The available ZSE data was supplied on excel sheets in the .xlsx format and were then converted to .csv format. To upload and view the entire ZSE data into Python, Code 1.1 in appendix was run first followed by;

```
In [2] ZSEdataFrame = pd.read_csv("ZSECleanStockData.csv")
pylab.rcParams['figure.figsize'] = (12, 6)
ZSEdataFrame
```

![Figure 4.1: Uploaded ZSE data](image)

Each counter has a different GAM model that fits its data. To illustrate that stock prices for various counters do behave differently with time, let us visualise a combined time series plot of CBZ Holdings, Delta, Econet, Innscor, Lafarge, OK and Old Mutual. To obtain this plot, Code 1.2 in appendix was run and the corresponding output would be figure 4.2 below. From the output, it can be observed that for sure various stocks behave differently and so each can be entitled to a different model to suit it.
To come up with an analysis for one of the counters on the ZSE say for instance ECONET, an extract of Econet data is made separately and saved as .csv file first and then to observe how the Econet data behaves with time, we need to plot the original data set, which is one of the major goals of time series analysis and the following output was obtained using Code 1.3 in Appendix;
Figure 4.3 above is quite similar but smoother than S. Mutendadzamera & F. K. Mutasa (2014)’s Runs Sequence Plot of the same data for the period February 2009 to December 2012, which confirms our findings. It can be noticed that, stock prices rose sharply throughout 2009 before stabilizing slightly from 2010 to mid-2011. After falling around 2012, the stock prices continued to exhibit an increasing trend until the last quarter of 2014 following the issuing of bond coins on 18/12/2014, before falling sharply towards the beginning of 2015. Reasons for these fluctuations are closely linked to economic announcements and transforms, political issues as well as cash flows. There was need to transform the data by taking logarithms to remove the unequal variances in order to achieve stationarity as required. Obtaining the log values of the data, and checking the head of the data;

In [6]  `EconetDataFrame.y = np.log(EconetDataFrame.y)`
In [7]  `EconetDataFrame.head()`
The corresponding output will be;

<table>
<thead>
<tr>
<th></th>
<th>ds</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2009-02-19</td>
<td>2.302585</td>
</tr>
<tr>
<td>1</td>
<td>2009-02-20</td>
<td>2.302585</td>
</tr>
<tr>
<td>2</td>
<td>2009-02-23</td>
<td>2.302585</td>
</tr>
<tr>
<td>3</td>
<td>2009-02-24</td>
<td>2.302585</td>
</tr>
<tr>
<td>4</td>
<td>2009-02-25</td>
<td>2.351375</td>
</tr>
</tbody>
</table>

**Table 4.1: Econet log-values of the stock data**

In order to come up with accurate forecast for our data, there was need to consider separately Zimbabwean holidays as explained in the following section.

### 4.2 ZIMBABWEAN HOLIDAYS

As mentioned earlier on in chapter 3, holidays distort findings if included in our data as investors tend to buy stocks before a given special day and then sell after. Hence, there is need to take into consideration a model for the holidays, and to do so a dataframe was created. It has two columns (holiday and ds) and a row for each occurrence of the holiday. Also included were all occurrences of the holiday, both in the past (back as far as the historical data go) and in the future (out as far as
the forecast is being made). If they won’t repeat in the future, Prophet will model them and then not include them in the forecast.

Columns lower_window and upper_window were included which extend the holiday out to days around the date. The known Zimbabwean holidays are;

<table>
<thead>
<tr>
<th>Day</th>
<th>Holiday</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 01</td>
<td>New Year Day</td>
</tr>
<tr>
<td>Apr 10</td>
<td>Good Friday (Holy or Great Friday)</td>
</tr>
<tr>
<td>Apr 11</td>
<td>Holy Saturday</td>
</tr>
<tr>
<td>Apr 13</td>
<td>Easter Monday</td>
</tr>
<tr>
<td>Apr 18</td>
<td>Independence Day</td>
</tr>
<tr>
<td>May 01</td>
<td>Workers or Labor Day</td>
</tr>
<tr>
<td>May 25</td>
<td>Africa Day</td>
</tr>
<tr>
<td>Aug 10</td>
<td>Heroes Day</td>
</tr>
<tr>
<td>Aug 11</td>
<td>Armed Forces Day</td>
</tr>
<tr>
<td>Dec 22</td>
<td>Unity Day</td>
</tr>
<tr>
<td>Dec 25</td>
<td>Christmas Day</td>
</tr>
<tr>
<td>Dec 26</td>
<td>Boxing Day</td>
</tr>
</tbody>
</table>

| Table 4. 2: Zimbabwean public Holidays |

In order to extract all the public holidays in our data Code 1.4 was run.

4.3 MODEL IDENTIFICATION AND MODEL FITTING

To identify and fit the model represented by a new Prophet object \( m \). Any settings to the forecasting procedure are passed into the constructor. We then call its fit method and pass in the historical dataframe, to achieve this Code 1.5 in appendix was used.

Predictions are then made on a dataframe with a column \( ds \) containing the dates for which a prediction is to be made. The number of days to be forecasted is then specified, \( H = 365 \) using the make_future_dataframe function. By default, it will also include the dates from the history, and the model fit will be visualized as well.

In [12]  \( H = 365 \)

\[
\text{m.fit(EconetDataFrame)}
\]
\[
\text{future} = \text{m.make_future_dataframe(periods=H)}
\]
\[
\text{forecast} = \text{m.predict(future)}
\]
future.tail(H)

Part of the output:

<table>
<thead>
<tr>
<th>ds</th>
<th>yhat</th>
<th>yhat_lower</th>
<th>yhat_upper</th>
</tr>
</thead>
<tbody>
<tr>
<td>1509</td>
<td>2015-02-19</td>
<td>4.216926</td>
<td>4.364740</td>
</tr>
<tr>
<td>1510</td>
<td>2015-02-20</td>
<td>4.223434</td>
<td>4.367208</td>
</tr>
<tr>
<td>1511</td>
<td>2015-02-21</td>
<td>4.205227</td>
<td>4.347414</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1872</td>
<td>2016-02-17</td>
<td>4.244846</td>
<td>5.481914</td>
</tr>
<tr>
<td>1873</td>
<td>2016-02-18</td>
<td>4.254856</td>
<td>5.515252</td>
</tr>
</tbody>
</table>

Table 4.3: An extract of the days to be forecasted in future.

The predict method was assigned each row in future a predicted value which it names yhat. If historical dates were passed in, it will provide an in-sample fit. The forecast object here is a new dataframe that includes a column yhat with the forecast, as well as columns for components and uncertainty intervals.

In [13] forecast = m.predict(future)

In [14] forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail(H)

The following results were obtained

<table>
<thead>
<tr>
<th>ds</th>
<th>yhat</th>
<th>yhat_lower</th>
<th>yhat_upper</th>
</tr>
</thead>
<tbody>
<tr>
<td>1509</td>
<td>2015-02-19</td>
<td>4.216926</td>
<td>4.364740</td>
</tr>
<tr>
<td>1510</td>
<td>2015-02-20</td>
<td>4.223434</td>
<td>4.367208</td>
</tr>
<tr>
<td>1511</td>
<td>2015-02-21</td>
<td>4.205227</td>
<td>4.347414</td>
</tr>
<tr>
<td>1512</td>
<td>2015-02-22</td>
<td>4.222040</td>
<td>4.360100</td>
</tr>
<tr>
<td>1513</td>
<td>2015-02-23</td>
<td>4.235083</td>
<td>4.376484</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1872</td>
<td>2016-02-17</td>
<td>4.244846</td>
<td>5.481914</td>
</tr>
<tr>
<td>1873</td>
<td>2016-02-18</td>
<td>4.254856</td>
<td>5.515252</td>
</tr>
</tbody>
</table>

Table 4.4: An extract of forecasts together with their upper and lower bounds.
4.4 FORECASTING STOCK PRICE PREDICTION MODEL

To visualise how the model is performing versus the actual log-stock prices, the following is run;

```
In [15]  fig = m.plot(forecast)
In [16]  m.plot(forecast);
        for cp in m.changepoints:
            plt.axvline(cp, c='gray', ls='--', lw=2)
```

![Forecasting the GAM Model from 19 Feb 2015 to 18 Feb 2016](image)

**Figure 4.4: A plot of forecasts together with original data series-Econet.**

It can be observed from the above outcome that the obtained GAM model (solid blue line), is performing pretty fine as it is following closely to the known observations (black dotes) having the ability to “jump” sudden change points. This is as expected since the GAM model has an automatic choice of smoothing thereby modelling the closing stock prices as a smooth function (Maindonald, 2010). The gray shading shows the confidence interval, which tend to widen as we increase the forecast horizon which is natural in all time series models (Ruey S., 2010).

4.5 AUTOMATIC CHangepoint DETECTION

Real time series frequently have abrupt changes in their trajectories (Taylor & Letham, 2017). By using Prophet, these changepoints can be automatically detected and allow the trend to adapt appropriately. Prophet does so by first specifying a large number of potential changepoints at
which the rate is allowed to change. It then puts a sparse prior on the magnitudes of the rate changes (equivalent to L1 regularization) - this essentially means that Prophet has a large number of possible places where the rate can change, but will use as few of them as possible. Considering the Econet Stock Price forecast, the vertical lines in the diagram below indicate where the potential changepoints were placed:

![Figure 4.5: Automatic change point detection by Prophet](image)

As indicated, several sudden change points were detected, the main reasons for these factors being the ones which enhance volatility as highlighted in section 4.26 of this chapter. In order to view the magnitude of the change in stock price Code 1.6 in appendix was run and figure 4.5 is obtained as output. From figure 4.4, it is quite clear that towards the end of 2009 stock prices fell sharply and this corresponds to the longest bar in figure 4.5 indicating the greatest negative change (fourth bar from the left). If stock prices were not prone to sudden changes, which is impossible in real life stock data, we were not going to observe any protruding bars in figure 4.5 thus the Prophet package is good in stock data analysis as its results are in line with what is expected (Ruey S., 2010).
Figure 4. 6: Magnitude of rate of change for the detected potential change points.

**Yearly and weekly seasonality of the time series**

Further inference could be made after decomposing the time series into its various components. To have sub-plots for these components, the following code is run and the output is;

```python
In [18] m.plot_components(forecast);
```
Figure 4. 7: Trend, Holidays, Weekly and Yearly Seasonality of Econet stocks.

Econet data exhibits an increasing trend for the rest of 2009 and stabilizes beginning of 2010 fluctuating at a log-price of 4 until beginning of 2013 (Mutendadzamera & Mutasa, 2014). Ever since then, the log-price was fluctuating above 4 as in table 1.5 above. The second plot evidently reveals the effects of holidays on the stock market. At the beginning of each year (new year holiday) stock prices tend to fall significantly and soon after these spikes are other sharp spikes occurring around the 1st quarter of each year (independence holiday). The same scenario with the one observed by (Congsheng, 2013) in which the Chinese New Year holiday had the same effects. One explanation could be that short-sellers close their risky positions prior to holidays and another could be investor’s poor mood around holidays thereby becoming less optimistic about future prospects hence a high probability of negative market moves. (StockCharts.com, 2017)

The third plot in figure 4.6 above reveals that Econet stock prices have an increasing trend from Sunday until Thursdays and Fridays before falling sharply to record minimums on Saturdays. This would imply that the best time for short-sellers to buy Econet stocks would be on Saturday, when the price is low, and sell either on Thursday or Friday when its highest.
From the last plot of figure 4.6, it can be noticed that Econet stock prices are generally low in February and it’s the best time to buy for long-sellers and despite the existence of sub-peaks, the stock prices are highest around mid-June throughout the year and that’s the best time to sell. Thus, the same intuition could be made using GAM models on the other six counters as tabulated below;

<table>
<thead>
<tr>
<th>Name of Counter</th>
<th>Time of Year to buy</th>
<th>Time of Year to sell</th>
<th>Day of week to buy</th>
<th>Day of week to sell</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. CBZH</td>
<td>Mid-April</td>
<td>Mid-June</td>
<td>Tuesday</td>
<td>Saturday</td>
</tr>
<tr>
<td>2. Delta</td>
<td>Mid-March</td>
<td>January</td>
<td>Sunday</td>
<td>Monday</td>
</tr>
<tr>
<td>3. Econet</td>
<td>February</td>
<td>Mid-June</td>
<td>Saturday</td>
<td>Thursday/Friday</td>
</tr>
<tr>
<td>4. Innscor</td>
<td>Mid-March</td>
<td>Mid-January</td>
<td>Tuesday</td>
<td>Saturday</td>
</tr>
<tr>
<td>5. Lafarge</td>
<td>End-April</td>
<td>End-August</td>
<td>Friday</td>
<td>Saturday</td>
</tr>
<tr>
<td>6. OK</td>
<td>Mid-March</td>
<td>August</td>
<td>Tuesday</td>
<td>Saturday</td>
</tr>
<tr>
<td>7. Old Mutual</td>
<td>Mid-December</td>
<td>January</td>
<td>Tuesday</td>
<td>Wednesday</td>
</tr>
</tbody>
</table>

Table 4.6: Best times to buy/sell stocks according to each counter’s GAM model.

4.6 VALIDATING THE TIME SERIES MODEL

Cross-validation is a popular technique for assessing a model’s effectiveness in predicting future values. Time series models are one exception where cross-validation would not work. Cross-validation involves dividing the dataset into random subsamples that are used to train and test the model repeatedly. Crucially, data points used in training samples must be independent of those in the test sample. But this is impossible in time series, because data points are time-dependent, so data in the training set would still carry time-based associations with the test set data. This calls for different techniques to validate time series models (Ruey S., 2010).

Instead of sampling our data points across time, we can slice them based on time segments (Annalyn & Kenneth, 2017). If we want to test our model’s forecast accuracy one year into the future (i.e. forecast horizon), we can divide our dataset into training segments of one-year (or longer), and use each segment to predict values in its subsequent year. This technique is called simulated historical forecasts. As a guide, if our forecast horizon is one-year H=365, then we should make simulated forecasts every month H/12. Having implemented Code 1.7 in appendix, part of the output of training time segments would be;
4.6.1 Mean Prediction Error Plot
To better assess the model’s accuracy, we can take the Mean Prediction Error from all the 14 simulated forecasts and plot that against the forecast horizon. To achieve this plot in appendix was run. Notice how the error increases as we try to forecast further into the future thus forecast accuracy decreases with an increase in forecast horizon as profounded by (Gor, 2009).

![Mean Prediction Error Plot](image)

**Figure 4.8: Mean prediction error plot.**

### 4.6.2 Mean Squared Error from the GAM model
(Ruey S., 2010, p. 194), Three statistics are commonly used to measure performance of point forecasts. They are the mean squared error (MSE), mean absolute deviation (MAD), and mean absolute percentage error (MAPE). For this matter, the mean squared error (MSE) for our forecast, was calculated for our GAM model as below;
The output shows that the MSE was 0.005 to 3 decimal places, which implies that the forecasts were more accurate and thus the GAM model was performing good as cited by (Maindonald, 2010).

### 4.6.3 Smooth Error Trend plot

One of the parameters we need to tune is the values of priors, which determine how sensitive our trend should be to changes in data values. One way to do this is to try different parameter values and compare the resulting errors via the plot shown below after running Code 1.9 in appendix. As we can see, we have a somewhat generalizable trend, with reasonable errors. Besides tuning the priors, we can also tweak settings the base growth model, seasonality trends, and special events. Visualizing our data also helps us identify and remove outliers.

![Smooth Error Trend plot](image)

**Figure 4.9: Smooth Error Trend plot**

### 4.7 ARIMA MODEL

In this section, the author will take you through a step by step process of finding the best ARIMA model using Python 3 for Econet data, as an illustration. The aim being to compare the performance of GAM and ARIMA models in financial data analysis. So as to load the data unto Python, format the date stamp to the required format as well as clean the data Code 2.0 in appendix was run.

Daily data can be tricky to work with, so monthly averages of our time-series will be used instead. This can be obtained by using the convenient resample function, which allows the grouping of the
time-series into buckets (1 month), apply a function on each group (mean), and combine the result (one row per group). All this is achieved after running Code 2.0 (appendix), and the output of line 6 within the same code reveals that the Econet data does not have any missing values so we can proceed.

When working with time-series data, a lot can be revealed through visualizing it. A few things to look out for are seasonality, trend and noises that is outlier points that are inconsistent with the rest of the data. The following code was implemented;

In [7]  Econet.plot(figsize=(15, 6))
plt.show()

![Figure 4.10: Time series plot of Econet data](image)

It can be noted that, Econet data exhibit some seasonality, a generally increasing trend and has noises in form of sharp peaks.

### 4.8 Decomposing the Time Series

Figure 4.10 above shows an obvious seasonality pattern, as well as an overall increasing trend as mentioned earlier on and we can further explore and visualize our data using a method called time-series decomposition. As its name suggests, time series decomposition allows us to decompose our time series into three distinct components: trend, seasonality, and noise. The script below In[8], shows how to perform time-series seasonal decomposition in Python3. By default, seasonal_decompose returns a figure of relatively small size, so the first two lines of this code chunk ensure that the output figure is large enough for us to visualize.

In [8]  from pylab import rcParams

| 51 | Time Series Analysis | Hakata Jonathan | B1336501 |
rcParams['figure.figsize'] = 11, 9

decomposition = sm.tsa.seasonal_decompose(y, model='additive')

fig = decomposition.plot()

plt.show()

Figure 4.11: Time series decomposition of Econet data.

Using time-series decomposition makes it easier to quickly identify a changing mean or variation in the data. The second and third plots of Figure 4.10 above clearly show a generally increasing trend in our data, along with its yearly seasonality. The residual plot shows that the data has noise that is, there is activity that confuses or misrepresents genuine underlying trends. This is mainly due to holiday effects, political activities, economic transformations and announcements. For instance, the sharp peak in residuals towards mid 2013 is due to the Zimbabwe general elections which were held from June until August 2013. The above decompositions can be used to
understand the structure of our time-series. The intuition behind this is important, as many forecasting methods build upon this concept of structured decomposition to produce forecasts (Makridakis.Hyndman.Wheelwright, 2007).

4.9 THE ARIMA TIME SERIES MODEL
In this section, a description of how to automate the process of identifying the optimal set of parameters for the seasonal ARIMA time series model is given.

Parameter Selection for the ARIMA Time Series Model in Python
The "grid search" will be used to iteratively explore different combinations of parameters. For each combination of parameters, we fit a new seasonal ARIMA model with the SARIMAX () function from the statsmodels module and assess its overall quality. To come up with examples of parameter combinations for Seasonal ARIMA models Code 2.1 in appendix was run and the corresponding output was:

```
<table>
<thead>
<tr>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examples of parameter combinations for Seasonal ARIMA...</td>
</tr>
<tr>
<td>SARIMAX: (0, 0, 1) x (0, 0, 1, 12)</td>
</tr>
<tr>
<td>SARIMAX: (0, 0, 1) x (0, 1, 0, 12)</td>
</tr>
<tr>
<td>SARIMAX: (0, 1, 0) x (0, 1, 1, 12)</td>
</tr>
<tr>
<td>SARIMAX: (0, 1, 0) x (1, 0, 0, 12)</td>
</tr>
</tbody>
</table>
```

**Table 4.6: Examples of parameter combinations.**

Code 2.2 (appendix), iterates through combinations of parameters and uses the SARIMAX function from statsmodels to fit the corresponding Seasonal ARIMA model. Here, the order_argument specifies the (p, d, q) parameters, while the seasonal_order argument specifies the (P, D, Q, S) seasonal component of the Seasonal ARIMA model. After fitting each SARIMAX() model, the code prints out its respective AIC score as shown in Table 1.8 below:

```
| ARIMA(0, 0, 0) x(0, 0, 1, 12)12 - AIC:606.8760885396434 |
| ARIMA(0, 0, 0) x(0, 1, 1, 12)12 - AIC:387.88345672525065 |
| ARIMA(0, 0, 0) x(1, 0, 0, 12)12 - AIC:508.3940605738556 |
| ARIMA(0, 0, 0) x(1, 0, 1, 12)12 - AIC:482.5832549320294 |
| ARIMA(0, 1, 1) x(1, 0, 1, 12)12 - AIC:274.53111825875936 |
| ... ... ... ... |
| ARIMA(1, 1, 1) x(1, 0, 1, 12)12 - AIC:324.13848955033416 |
```
Table 4.7: ARIMA models together with their respective AIC values.

The output of our code suggests that ARIMA (0, 1, 1)x(1, 1, 12) yields the lowest AIC value of 274.53. We should therefore consider this to be the optimal option out of all the models we have considered (Ruey S., 2010). Likewise, running the same code for the other counters will yield the following ARIMA models which have the least AIC values;

<table>
<thead>
<tr>
<th>Name of Counter</th>
<th>ARIMA (p,d,q)x(P,D,Q,s)Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. CBZ Holdings</td>
<td>(1, 1, 1)x(1, 0, 1, 12)</td>
</tr>
<tr>
<td>2. Delta</td>
<td>(1, 1, 1)x(1, 0, 1, 12)</td>
</tr>
<tr>
<td>3. Innscor</td>
<td>(1, 1, 0)x(1, 0, 1, 12)</td>
</tr>
<tr>
<td>4. Lafarge</td>
<td>(0, 1, 1)x(0, 0, 0, 12)</td>
</tr>
<tr>
<td>5. OK</td>
<td>(1, 1, 1)x(0, 1, 1, 12)</td>
</tr>
<tr>
<td>6. Old Mutual</td>
<td>(0, 1, 0)x(1, 0, 1, 12)</td>
</tr>
</tbody>
</table>

Table 4.9: Selected ARIMA models for the other counters.

4.9.1 Fitting the ARIMA Time Series Model

Using grid search, a set of parameters that produces the best fitting model to our time series data have been obtained. An analysis of Econet’s ARIMA model in more depth will start by plugging the optimal parameter values into a new SARIMAX model to get coefficients;

```
mod = sm.tsa.statespace.SARIMAX(y,
    order=(0, 1, 1),
    seasonal_order=(1, 1, 12),
    enforce_stationarity=False,
    enforce_invertibility=False)
results = mod.fit()
print(results.summary().tables[1])
```

Table 4.8: Summary of results for the ARIMA model
The summary attribute that results from the output of SARIMAX returns a significant amount of information, but we'll focus our attention on the table of coefficients. The coef column shows the weight (i.e. importance) of each feature and how each one impacts the time series. The P>|z| column informs us of the significance of each feature weight. Here, the auto-regressive and moving average seasonal parameters have p-values above 0.05, so they are less significant and can be ignored in our model (Ruey S., 2010).

4.9.2 Model Diagnostics
When fitting seasonal ARIMA models (and any other models for that matter), it is important to run model diagnostics to ensure that none of the assumptions made by the model have been violated. The plot_diagnostics object allows us to quickly generate model diagnostics and investigate for any unusual behavior.

```
In [12] results.plot_diagnostics(figsize=(15, 12))
plt.show()
```

![Residual plots to diagnose the ARIMA model.](image)

Figure 4.12: Residual plots to diagnose the ARIMA model.
Our primary concern is to ensure that the residuals of our model are uncorrelated and normally distributed with zero-mean. If the seasonal ARIMA model does not satisfy these properties, it is a good indication that it can be further improved.

As stated in section 2.3.3, Ansah (2014) highlighted that, if the residuals are normally distributed, the points on the normal quantile-quantile (QQ) plot should approximately be linear, with residual mean as the intercept and residual standard deviation as the slope whilst the shape of the histogram shows “a bell-like” shape (Ansah, 2014). Those observations lead us to conclude that our model produces a satisfactory fit that could help us understand our time series data and forecast future values.

### 4.9.3 Model Validation
We have obtained a model for our time series, ARIMA (0,1,1), that can now be used to produce forecasts. We start by comparing predicted values to real values of the time series, which will help us understand the accuracy of our forecasts.

```python
In [13] ax = y['2009:'].plot(label='observed',figsize=(12, 8))
    pred.predicted_mean.plot(ax=ax, label='One-step ahead Forecast', alpha=.7)
    ax.fill_between(pred_ci.index,
                    pred_ci.iloc[:, 0],
                    pred_ci.iloc[:, 1], color='k', alpha=.25)
    ax.set_xlabel('YEAR')
    ax.set_ylabel('CLOSING STOCK PRICE')
    ax.set_ylabel('CLOSING STOCK PRICES');plt.legend();plt.show()
```

![Figure 4.13: Model validation](image.png)
Overall, our forecasts align closely with the true values, showing an overall increase trend. It is also useful to quantify the accuracy of our forecasts. We will use the MSE (Mean Squared Error), which summarizes the average error of our forecasts. For each predicted value, we compute its distance to the true value and square the result. The results need to be squared so that positive/negative differences do not cancel each other out when we compute the overall mean.

In [14] y_forecasted = pred.predicted_mean
    y_truth = y['2013-01-01':]
    # Compute the mean square error
    mse = ((y_forecasted - y_truth) ** 2).mean()
    print('The Mean Squared Error of our forecasts is {}\,format(round(mse, 2))

Output
The Mean Squared Error of our forecasts is 26.71

The MSE of our one-step ahead forecasts yielded a value of 26.71, which is very high as it is greater than 0. An MSE of 0 would imply that the estimator is predicting observations of the parameter with perfect accuracy, which would be an ideal scenario which is not typically possible especially when dealing with financial data which is generally noisy (Devi, Sundar, & Alli, 2013).

**Forecasts**

Code 2.3 was used to produce forecasts and visualize a plot of the existing time series together with its future values.

![Figure 4. 14: Observed time series and the forecasts from 18/02/2015 – 17/02/2016.](image)
Both the forecasts and associated confidence interval that we have generated can now be used to further understand the time series and foresee what to expect (Hyndman & Athanasopoulos, 2014). Our forecasts show that the time series is expected to continue fluctuating lowly around a closing stock price of 50.

As we forecast further out into the future, it is natural for us to become less confident in our values. This is reflected by the confidence intervals generated by our model, which grow larger as we move further out into the future indicated by the gray shade around the orange line (Ruey S., 2010).

4.9.10 VOLATILITY
Asset prices are subject to high risk due to factors like inflation which affect future cash flows hence they are ever volatile (Radtke, 2014). To check for volatility in the data, a plot of the residuals was obtained as follows;

![Standardized Residuals](image)

**Figure 4.15: A plot of Standardized Residuals to check for the ARCH effect.**

Visual inspection of the above output clearly shows that the residuals are not independent and identically distributed (i.i.d), and such a behaviour is a common feature of GARCH processes (Wang, Van Gelder, J.K, & Ma, 2005).

By implementing Code 2.4 in Jupyter Notebook, Python’s open source web-application, an analysis of volatility of closing stock prices is achieved. Econet data was used and after running Code 2.4 data was loaded and formatted. Percentage changes, 21-day standard deviation, 21-day historical volatility and variances were calculated and part of the corresponding output was;
Calculating the stock returns, finding and fitting a suitable GARCH (1,1) model for our data, the following is run:

```
In [4] from arch import arch_model
returns = EconetDataFrame['pct_change'] * 100
am = arch_model(returns)
res = am.fit(); res.params  #Extracting the model parameters omega, alpha and beta
```

After performing several iterations using the method of likelihood estimator the best estimates for the parameters mu, omega, alpha and beta are $0.020127, 0.010221, 0.031487$ and $0.965265$ respectively as in Table 4.11. Observe that $\alpha + \beta < 1$ as expected, (Ruey S., 2010, p. 115). More so the p-values of coefficients alpha and beta are both less than 0.05 indicating that they are highly significant in our model. As profounded by (Osama, 2014) in his conference publication entitled “Modelling Stock Market Volatility Using Univariate GARCH Models: Evidence from Pakistan”, the significance of alpha and beta indicates that, these two have an impact on the conditional variance, in other words this means that history about volatility from the previous periods have an explanatory power on current volatility (Ruey S., 2010).

<table>
<thead>
<tr>
<th>ds</th>
<th>y</th>
<th>pct_change</th>
<th>stdvol21</th>
<th>hvol21</th>
<th>variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>42</td>
<td>2009-04-22</td>
<td>13.0</td>
<td>0.058911</td>
<td>0.047951</td>
<td>0.761196</td>
</tr>
<tr>
<td>43</td>
<td>2009-04-23</td>
<td>15.5</td>
<td>0.192308</td>
<td>0.057313</td>
<td>0.909822</td>
</tr>
<tr>
<td>44</td>
<td>2009-04-24</td>
<td>15.5</td>
<td>0.000000</td>
<td>0.057313</td>
<td>0.909822</td>
</tr>
<tr>
<td>45</td>
<td>2009-04-27</td>
<td>16.0</td>
<td>0.032258</td>
<td>0.052943</td>
<td>0.840441</td>
</tr>
<tr>
<td>46</td>
<td>2009-04-30</td>
<td>15.6</td>
<td>-0.025000</td>
<td>0.054224</td>
<td>0.860773</td>
</tr>
</tbody>
</table>

Table 4.9: Percentage change, standard deviation, volatility and variance.
Table 4.10: Parameter values for the GARCH (1,1) model.

Following the model formula $\sigma_t^2 = \omega + \alpha r_{t-1}^2 + \beta \sigma_{t-1}^2, \ldots \ldots \ 2.18$, assigning letters A, B and C to the parameters alpha, beta and omega respectively, Code 2.5 is executed to obtain daily volatility forecasts for the entire period and part of the output is;

Table 4.11: Daily volatility forecasts obtained from the selected GARCH (1,1) model.

For us to be confident with the performance of our model, we need to visualise how the actual 21-day historical volatilities are related to their forecasted volatilities. Thus, in order to validate our model this way, Code 2.6 was run and the output was;
Figure 4.16: Model validation - a plot of actual volatility vs their forecasts.

In order to get the MSE for the forecasts we obtained, the following chunk of code was run and the output was as follows;

```python
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(EconetDataFrame['hvol21'], EconetDataFrame['forecast_vol'])
print('The Mean Squared Error of our forecasts is {}'.format(round(mse, 3)))
```

The Mean Squared Error of our forecasts is 0.001

As shown by the blue line in Figure 4.14, the Econet stock prices were highly volatile at the beginning of 2009, a period when Zimbabwe had the highest inflation rates ever recorded as also cited by (Murekachiro, 2016). However, the graph falls significantly until mid-2009 following the introduction of the multi-currency system in February 2009. Volatility increased to levels above 0.5 towards the beginning of 2012 due to political instability and violence among political groups (Mutendadzamera & Mutasa, 2014).

Following the mentioning of the re-introduction of the Zimbabwean-dollar on 26th of June 2009 in a speech by The President (Herald, 2009), trade fell significantly and this is shown by that sharp peak. Stock trading business became unstable again, as investors lost confidence following the Zimbabwe general elections held from 31st of July 2013 until beginning of August 2013 when President Robert Mugabe was officially declared the winner according to The Herald (2013, August 3).
The brown line, which shows the forecasts obtained from the selected GARCH (1,1) model, is following closely the fluctuations in the actual historical volatilities recorded thus we can say our model is performing good. More so, an MSE of 0.001 that we obtained for our forecasts supports the fact that the volatility model used was good. So, by following the same procedure for any other counter, a suitable GARCH (1,1) model is accurately determined in order to ascertain the level of volatility (Brooks, 2008).
CHAPTER 5

SUMMARY, RECOMMENDATIONS AND CONCLUSION

5.1 INTRODUCTION
This chapter gives a conclusion drawn from the research as well as recommendations on the analysis of Industrial performances on the Zimbabwe Stock Exchange. Highlights of key findings are given together with conclusions that address research objectives and finally the recommendations on what can be done to enhance investor participation on the local bourse.

5.2 SUMMARY OF FINDINGS FROM THE STUDY
Counters do not behave in the same manner with time as indicated by a time series plot for the randomly selected namely CBZH, DELTA, ECONET, LAFARGE, INNISCOR, OK and OLD MUTUAL. As illustrated in section 4.4 above, each counter then will have a different GAM and ARIMA model to be fitted to its data. From the GAM model obtained, further inferences can be made for instance, Econet’s GAM model shows that there were several sudden changes in stock trades. One major sudden change point is being observed around July 2013 and this was due to general elections held then. This may imply that traders lost confidence in doing business on the ZSE and hence the sudden drop. As mentioned by (Ruey S. T., 2005), “to handle the price variations caused by rare events (e.g., a profit warning), we also study some simple diffusion models with jumps,” and in our case, we used the GAM model which accounted for spikes in the Econet Stock prices that coincided with special events. Thus, stock prices are very sensitive to political waves, announcements on economic transformations, inflation as well as special events.

Unlike ARIMA models, GAM models use the Prophet package to further decompose a financial time series to get the holiday, annual and weekly trends. It is from these plots that financial analysts advice both short and long-time investors the right time either to buy or sell. For instance, weekly trends (Figure 4.6) revealed that the Econet Stock Price is at its lowest on a Monday and highest on a Friday and annual trends show that the Econet Stock Price peaks in March, June and October. So, the best time to buy the Econet Stocks is when it is low (April, end of September, December and January) and the best time to sell is when it is high that is around mid-June. Such a deeper analysis in trying to address investor queries, cannot be achieved using ARIMA models.
More so, in this research work, it was found out that GAM models are more accurate than ARIMA models in coming up with forecasts. This is so because, in the case of Econet, the MSE obtained from GAM forecasts was 0.005 (section 4.6.2) compared to 26.71 for ARIMA (0,1,1) model forecasts (section 4.9.3). A MSE of 26.71 that was obtained for ARIMA (0,1,1) may suggest more room for further developing it despite the fact that financial data which was under study is volatile. The other reason owing to the poor performance of the ARIMA (p,d,q) could be that, the model does not take into account holidays which tend to distort fluctuations in stock prices. It was also observed that the DELTA and CBZ Holdings had the same ARIMA(1, 1, 1)x(1, 0, 1, 12) model while LAFARGE, INNSCOR, OK and OLD MUTUAL were modelled differently by ARIMA(0, 1, 1)x(0, 0, 0, 12), ARIMA(1, 1, 0)x(1, 0, 1, 12), ARIMA(1, 1, 1)x(0, 1, 1, 12) and ARIMA(0, 1, 0)x(1, 0, 1, 12) respectively.

The GARCH (1,1) whose parameters were carefully selected was thought to be the best for modelling volatility for stock prices as evidenced by Figure 4.14 as well as a MSE of 0.001 which is close to zero in the case of Econet data. As explained by Brooks (2008), the lower the model’s error statistic is, the more accurate the model is in estimating volatility. Also, the sums of the GARCH coefficients are less than 1 for which ensures that the GARCH processes are covariance stationary. Still on, volatility, it was found out that political instability enhances volatility to a greater extend.

5.3 CONCLUSION
Overall, GAM models outperform ARIMA models in financial modelling as they help to answer many investor questions on investing their hard-earned moneys. To prospect investors who have been doubting to make it on our local bourse, the findings herein have revealed that one can actually recognise profits having known levels of volatility and the best time, be it of the year or of the week, to buy and sell. In modelling volatility, there is need to try to employ the other general models like GARCH (p,q), ARCH (p,q), TGARCH and EGARCH even though it is complicated to find the parameter values, instead of using the specific one GARCH (1,1).
5.4 RECOMMENDATIONS
There is still need for modelling stock price volatility using conditional heteroscedastic models in Zimbabwe since this area have not been sufficiently researched. More so there is still a need to carry out a fundamental analysis of the ZSE as it will take a closer look at the underlying factors causing the existing trends in stock prices which have been revealed in this research. Just as Selene Yue Xu discovered in her research, “The effect of news on the stock price changes”, the changes in weekly stock prices and the values of important news/events computed from the Google trend website were found to be strongly correlated. As highlighted in the summary of findings about political instability causing stock prices to be volatile, the researcher wishes to recommend the need for strategies to encourage peace. In the event of political instability, investors tend to lose confidence in doing business on our local bourse and dash out in search for greener peaceful pastures thereby retarding any efforts made to turn around the economy.
BIBLIOGRAPHY


IBM. (2013). IBM SPSS Forecasting 22.
Laurinec, P. (2017, January 26). Doing magic and analyzing seasonal time series with GAM (Generalized Additive Model) in R.
RayBall. (1994).


Georgia: Electronic Theses and Dissertations 668.


*Forecasting Methods and Stock Market Analysis.* Creative Math.


APPENDIX

Code 1.1
In [1] # Code for populating the interactive namespace from numpy and matplotlib

```python
%pylab inline
from pandas.tools.plotting import autocorrelation_plot
from pandas.tools.plotting import scatter_matrix
import matplotlib.pyplot as plt
import StringIO
import pandas as pd
import time
import scipy.stats as scs
import statsmodels.api as sm
import numpy as np
import os
import datetime
from datetime import date
from fbprophet import Prophet
import matplotlib.dates as mdates
import matplotlib.cm as cm
from scipy.signal import savgol_filter
```

Code 1.2
In [2] # Code to come up with a combined plot for various stock counters

```python
>start = datetime.datetime(2009,2,19)
>end = datetime.datetime(2015,2,18)

# Below I create a DataFrame consisting of the adjusted closing price of these stocks, first
# by making a list of these objects and using the join method
>stocks = pd.DataFrame({"Econet": Econet["ECONET"],
    "Delta": Delta["DELTA"],
    "Cbzh": Cbzh["CBZH"],
    "Innscor": Innscor["INNSCOR"],
    "Lafarge": Lafarge["LAFARGE"],
    "Ok": Ok["OK"],
})
```
"Old_mutual": Old_mutual["OLD MUTUAL"],
"ds": Delta["Date"],
})

**Code 1.3**

In [2]  
# Code for obtaining a plot of the original Econet Data

EconetDataFrame = pd.read_csv("ECONET.csv")

In [3]  
EconetDataFrame = EconetDataFrame.replace('0', np.NAN)#Replacing the blanks with NA

In [4]  
EconetDataFrame = EconetDataFrame.rename(columns={'DS': 'ds', 'ECONET': 'y'})

In [5]  
EconetDataFrame['ds'] = pd.to_datetime(EconetDataFrame['ds'])#Format date stamp

In [6]  
pylab.rcParams["figure.figsize"] = (12, 6)

fig=EconetDataFrame.plot(x='ds', y='y', grid=True, label='Series')

fig.set_ylabel('Closing Stock Price')

fig.set_xlabel('Year')

fig.set_title('Econet Feb 2009-Feb 2015')

**Code 1.4**

In [8]  
# A code for extracting all the Zimbabwean public holidays

NewYearsDay = pd.DataFrame({
    '2014-01-01', '2015-01-01', '2016-01-01']),
    'lower_window': 0, 'upper_window': 1 })

GoodFriday = pd.DataFrame({
    'holiday': 'GoodFriday', 'ds': pd.to_datetime(["2009-04-10", '2010-04-02', '2011-04-22', '2012-04-06', '2013-03-29',
    '2014-04-18', '2015-04-03', '2016-03-25']),
    'lower_window': 0, 'upper_window': 1 })

HolySaturday = pd.DataFrame({
    '2014-04-19', '2015-04-04', '2016-03-26']),
    'lower_window': 0, 'upper_window': 1 })

EasterMonday = pd.DataFrame({

  '2014-04-21', '2015-04-06', '2016-03-28']),
'lower_window': 0, 'upper_window': 1 })

IndependanceDay = pd.DataFrame({
  'lower_window': 0, 'upper_window': 1 })

WorkersDay = pd.DataFrame({
  'lower_window': 0, 'upper_window': 1 })

AfricaDay = pd.DataFrame({
  'lower_window': 0, 'upper_window': 1 })

HerosDay = pd.DataFrame({
  'holiday': 'HerosDay', 'ds': pd.to_datetime(['2009-08-10', '2010-08-10', '2011-08-10', '2012-08-10', '2013-08-10',
  '2014-08-10', '2015-08-10', '2016-08-10']),
  'lower_window': 0, 'upper_window': 1 })

ArmedForcesDay = pd.DataFrame({
  'holiday': 'ArmedForcesDay', 'ds': pd.to_datetime(['2009-08-11', '2010-08-11', '2011-08-11', '2012-08-11', '2013-08-11',
  '2014-08-11', '2015-08-11', '2016-08-11']),
  'lower_window': 0, 'upper_window': 1 })

UnityDay = pd.DataFrame({
  'holiday': 'UnityDay', 'ds': pd.to_datetime(['2009-12-22', '2010-12-22', '2011-12-22', '2012-12-22', '2013-12-22',
  '2014-12-22', '2015-12-22', '2016-12-22'],
'2014-12-22', '2015-12-22', '2016-12-22')),
'lower_window': 0, 'upper_window': 1 })

ChristmasDay = pd.DataFrame(
    
'holiday': 'ChristmasDay', 'ds': pd.to_datetime(
    ['2009-12-25', '2010-12-25', '2011-12-25', '2012-12-25', '2013-12-25',
    '2014-12-25', '2015-12-25', '2016-12-25']),
'lower_window': 0, 'upper_window': 1 })

BoxingDay = pd.DataFrame(
    
'holiday': 'BoxingDay', 'ds': pd.to_datetime(
    ['2009-12-26', '2010-12-26', '2011-12-26', '2012-12-26', '2013-12-26',
    '2014-12-26', '2015-12-26', '2016-12-26']),
'lower_window': 0, 'upper_window': 1 })

# Now compiling and printing the holidays together we do;
In [9] holidays = pd.concat((NewYearsDay, GoodFriday, HolySaturday,
                          EasterMonday, IndependanceDay, WorkersDay,
                          AfricaDay, HerosDay, ArmedForcesDay, UnityDay,
                          ChristmasDay, BoxingDay))

In [10] print holidays

**Code 1.5**

In [11] # Code for identifying the GAM model and fitting the model to the historical data
m = Prophet(interval_width=0.95, changepoint_prior_scale=0.05, # default is 0.05,
Decreasing it will make the trend less flexible
holidays=holidays,
seasonality_prior_scale=10, # default is 10, strength of the holiday components
yearly_seasonality=True, # Since Zimbabwe Public Holidays reoccur annually
weekly_seasonality=True, # Stock brokers conduct stock trading on weekdays
seasonality_prior_scale=10, # default is 10, larger values allow larger seasonal
fluctuations
# mcmc_samples=500 # to generate confidence intervals for seasonality and holiday
components )
**Code 1.6**

In [17] # A code to visualise the magnitude of the potential stock price changes

deltas = m.params['delta'].mean(0)
fig = plt.figure(facecolor='w', figsize=(10, 6))
ax = fig.add_subplot(111)
ax.bar(range(len(deltas)), deltas, facecolor='#0072B2', edgecolor='#0072B2')
ax.grid(True, which='major', c='gray', ls='-', lw=1, alpha=0.2)
ax.set_ylabel('Rate change')
ax.set_xlabel('Potential changepoint')
fig.tight_layout()

**Code 1.7**

In [19] # Code for training time segments.

    H = 365  # forecast horizon
    # frequency of simulated forecasts
    h = H/12
    # tuning and validation: simulated historical forecast
    # for storing forecast results and cutoffdates
    results = pd.DataFrame()
cutoff = []
    # run forecast simulations
    i = 0
    while (len(EconetDataFrame)-i 3*H):  # define training data
        train = EconetDataFrame[i:(i+(3*H))]  # use 3 periods of data for training
        # fit time series model
        model = Prophet(interval_width=0.95,
                         changepoint_prior_scale=0.05,  # default is 0.05, Decreasing it will make the trend less flexible
                         holidays=holidays,
                         holidays_prior_scale=10,  # default is 10, strength of the holiday components
                         yearly_seasonality=True,  # Since Zimbabwe Public Holidays reoccur annually
                         weekly_seasonality=True,  # Stock brokers conduct stock trading on weekdays only
seasonality_prior_scale=10, # default is 10, larger values allow larger seasonal fluctuations
#mcmc_samples=500 # to generate confidence intervals for seasonality and holiday components
)

model.fit(train); # future dates for which to make forecasts

future = model.make_future_dataframe(periods=H)
future.tail(H)

# make forecast
forecast = model.predict(future)
resultsH = forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail(H)

# get actual values to compare with predicted values
resultsH = EconetDataFrame.merge(resultsH, how='right')

# sort by increasing date
resultsH = resultsH.sort_values(by='ds')

# record cutoff dates

cutoffDate = resultsH['ds'].iloc[0]
cutoffDate = cutoffDate.strftime('%Y %b')
cutoff = cutoff + [cutoffDate]

# compile results
results = pd.concat((results, resultsH), ignore_index=True)

print 'Counting the days...', i

i = i + h

Code 1.8

In [22] # Code for Mean Prediction Error Plot

error = abs(results.yhat - results.y)
error = error.values.reshape(ns,H)

# average error with respect to forecast horizon
errorMean = np.nanmean(error, axis=0)

# smooth error trend
errorMeanSmooth = savgol_filter(errorMean, 365, 3)
Code 1.9
In [24] # Code for obtaining a Smooth Error Trend plot

# prediction error
error2 = abs(results2.yhat - results2.y)
error2 = error2.values.reshape(ns,H)

# average error with respect to forecast horizon
errorMean2 = np.nanmean(error2, axis=0)

# smooth error trend
errorMeanSmooth2 = savgol_filter(errorMean2, 365, 3)

# plot error along forecast horizon
plt.xlim([1,H])
plt.plot(range(H), errorMeanSmooth2, c='r', lw=2, label='scale = 10')
plt.plot(range(H), errorMeanSmooth, c='k', lw=2, label='scale = 0.01')
plt.xlabel('Forecast Horizon (days)')
plt.ylabel('Mean Absolute Prediction Error')
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.,
           title='Change Point Prior')
plt.savefig('time2-errorCompare.png', format='png', bbox_inches='tight', dpi=1000)
plt.show()
import matplotlib.pyplot as plt

In [2] Econet = pd.read_csv("ECONET.csv")
print(Econet.head())

In [3] Econet['DS'] = pd.to_datetime(Econet['DS']) # Format the date stamp column
print(Econet.head())

In [4] Econet=Econet.set_index('DS') # Making the date column be the index
Econet.index

In [5] y = Econet['ECONET'].resample('MS').mean()
y.head()

In [6] Econet.isnull().sum()

    ECONET      0
    dtype: int64

**Code 2.1**

In [9] # Code for examples of parameter combinations
    # Define the p, d and q parameters to take any value between 0 and 2
    p = d = q = range(0, 2)
    # Generate all different combinations of p, q and q triplets
    pdq = list(itertools.product(p, d, q))
    # Generate all different combinations of seasonal p, q and q triplets
    seasonal_pdq = [(x[0], x[1], x[2], 12) for x in list(itertools.product(p, d, q))] 
    print('Examples of parameter combinations for Seasonal ARIMA...')
    print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[1]))
    print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[2]))
    print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[3]))
    print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[4]))

**Code 2.2**

In [10] # A code for various ARIMA models and their respective AIC values
    import warnings
    plt.style.use('fivethirtyeight') # A matplotlib style my plots.
    warnings.filterwarnings("ignore") # specify to ignore warning messages
    for param in pdq:
for param_seasonal in seasonal_pdq:
    try:
        mod = sm.tsa.statespace.SARIMAX(y,
                                          order=param,
                                          seasonal_order=param_seasonal,
                                          enforce_stationarity=False,
                                          enforce_invertibility=False)
        results = mod.fit()
        print('ARIMA{}x{}12 - AIC:{}'.format(param, param_seasonal, results.aic))
    except:
        continue

Code 2.3
In [15] # A code for visualising and producing forecasts.
    # Get forecast 12 steps ahead in future
    pred_uc = results.get_forecast(steps=12)
    # Get confidence intervals of forecasts
    pred_ci = pred_uc.conf_int()
    ax = y.plot(label='observed', figsize=(12, 8))
    pred_uc.predicted_mean.plot(ax=ax, label='Forecast')
    ax.fill_between(pred_ci.index,
                    pred_ci.iloc[:, 0],
                    pred_ci.iloc[:, 1], color='k', alpha=.25)
    ax.set_xlabel('Date')
    ax.set_ylabel('Closing Stock')
    plt.legend(), plt.show()

Code 2.4
In [1] # Code for calculating the 21-day volatility
    import numpy as np
    import pandas as pd
    from io import StringIO as web # calling the necessary packages
In [2] EconetDataFrame = pd.read_csv("ECONET.csv")#Loading the data,
EconetDataFrame = EconetDataFrame.rename(columns={'DS': 'ds','ECONET': 'y'})
EconetDataFrame['ds'] = pd.to_datetime(EconetDataFrame['ds'])

Calculating the annualised volatility of say a 21-day

In [3]  
  EconetDataFrame['pct_change'] = EconetDataFrame['y'].pct_change().dropna()  
  EconetDataFrame['stddev21'] = pd.rolling_std(EconetDataFrame['pct_change'], 21)  
  EconetDataFrame['hvol21'] = EconetDataFrame['stddev21']*(252**0.5) # Annualize.  
  EconetDataFrame['variance'] = EconetDataFrame['hvol21']**2  
  EconetDataFrame = EconetDataFrame.dropna() # Remove rows with blank cells.  
  EconetDataFrame.head()

**Code 2.5**

In [5]  
  # Code for obtaining tabulated daily volatility forecasts for the entire period.  
  EconetDataFrame['C'] = res.params['omega']  
  EconetDataFrame['B'] = EconetDataFrame['variance'] * res.params['beta[1]']  
  EconetDataFrame['A'] = (EconetDataFrame['pct_change']**2) * res.params['alpha[1]']  
  EconetDataFrame['forecast_var'] = EconetDataFrame.loc[:, 'C':'A'].sum(axis=1)  
  EconetDataFrame['forecast_vol'] = EconetDataFrame['forecast_var']**0.5  
  EconetDataFrame.head()

**Code 2.6**

In [6]  
  # Code for visualising the actual volatility versus their forecasts.  
  EconetDataFrame=EconetDataFrame.set_index('ds')  
  %pylab inline  
  pylab.rcParams['figure.figsize'] = (12,6)  
  fig=EconetDataFrame[['hvol21','forecast_vol']].plot()  
  fig.set_ylabel('Volatility')  
  fig.set_xlabel('Year')  
  fig.set_title('Econet Volatility Feb 2009-Feb 2015')