TIME SERIES MODELING OF FERTILISER DEMAND IN KENYA

JAMES MWITI MUTEGI

A Thesis Submitted to the Graduate School in Partial Fulfilment of the Requirements for the Award of the Degree of Master of Science in Applied Statistics of Chuka University

CHUKA UNIVERSITY

OCTOBER 2024

DECLARATION AND RECOMMENDATIONS

Declaration

This thesis is my original work and has not been presented for an award of diploma or conferment of degree in this or any other University.

Signature _____ Date 14/10/2024

James Mwiti Mutegi
SM18/58007/22

Recommendations

This thesis has been examined, passed and submitted with our approval as University supervisors.

Signature _____ Date 14/10/2024

Dr. Elizabeth W. Njoroge, PhD
Chuka University

Signature _____ Date 14/10/2024

Prof. Moses M. Muraya, PhD
Chuka University

DEDICATION

I dedicate this thesis to my beloved wife, whose unwavering love and support have been the foundation of my academic journey. Their encouragement, patience, and understanding have been instrumental in helping me overcome challenges and stay focused on my goals. They have been my pillar of strength, and I am forever grateful. To my cherished daughters, Paula Makena and Loreen Nyaguthii, who are my constant motivation and the reason I strive for excellence. Their bright spirits and boundless curiosity remind me of the importance of continuous learning and growth. May this dedication inspire them to pursue their dreams with determination and fearlessness. A special acknowledgement goes to my mentor, Hussein Gitonga, whose guidance, expertise, and patience have shaped my scholarly development. His mentorship has been transformative, and I am grateful for the knowledge and skills he has imparted to me.

ACKNOWLEDGEMENT

I extend my heartfelt gratitude to my esteemed supervisors, Dr. Elizabeth W. Njoroge and Prof. Moses M. Muraya, whose unwavering guidance, expertise, and constructive feedback have been instrumental in shaping this thesis. Their dedication and commitment to my academic growth have been truly inspiring, and I am honoured to have had the privilege of working under their mentorship.

I am indebted to Chuka University for allowing me to pursue my studies at this esteemed institution. The academic environment and resources offered by the university have played a pivotal role in enriching my learning experience, and I am grateful for the support from the faculty and staff.

Last but not least, I acknowledge the collective efforts of all those who have been a part of my academic and personal journey. Each interaction has contributed to my growth and development as a scholar.

ABSTRACT

The agricultural sector is the backbone of the Kenyan economy, contributing approximately 33% of the Gross Domestic Product. The agriculture sector employs more than 40% of the total population and 70% of the rural population. However, agricultural productivity has stagnated recently due to various constraints, including poor agronomic practices such as fertiliser application. One of the major constraints in crop production is the timeliness of fertiliser application. If fertilisers are applied too early or too late, plants might not absorb the nutrients efficiently, leading to reduced growth and lower yields. Understanding the patterns of fertiliser demand helps in better planning and management of fertiliser supply chains, ensuring that farmers have access to the right types and amounts of fertilisers when they need them most. Therefore, there is need to understand the demand for fertiliser across different agroecological zones and its timely delivery to the farming communities. The objective of this study was to apply time series forecasting techniques to model fertiliser demand in Kenya based on the secondary monthly data from 2010 – 2023. The data was obtained from the Ministry of Agriculture and Livestock Development headquarters in Nairobi, Kenya. R- Studio software (version 2023.12.1+402) was utilised to analyse data for descriptive statistics, fertiliser demand variability, and model fitting. The Box-Jenkins method was used to model and forecast fertiliser demand variability. The findings of this study indicated that the demand for fertilisers is seasonal. In addition, fertiliser demand in Kenya experiences significant demand fluctuations over time due to the seasonality of the agricultural practices. STL decomposition was applied to separate the time series data into trend, seasonal, and residual components, allowing for a clear analysis of seasonal patterns and underlying trends. This study also found that the demand for various types of fertilisers varied from month to month, and the demand was high during the months of March, April, July, August, October and November. The Akaike Information Criterion (AIC) test was used to compare different SARIMA models, with lowest AIC values indicating better model fit and complexity balance. The study's findings revealed that the demand for different types of fertilisers can be modelled by the following Seasonal Autoregressive Integrated Moving Average (SARIMA) models: Calcium nitrate; SARIMA (1,1,1) (0,0,1)[2], Diammonium Phosphate; SARIMA (0,0,0) (2,0,0)[12], Muriate of potash; SARIMA (1,1,4) (0,0,1)[12], NPK; SARIMA (2,0,0) (2,0,0)[12], Calcium ammonium nitrate; ARIMA (0,0,0)w/mean, Urea; ARIMA (0,1,1) and total fertiliser demand SARIMA; (1,1,4) (0,0,1)[12]. Most fertilisers, especially Diammonium Phosphate, Muriate of Potash, and Nitrogen, Phosphorus and Potassium Fertiliser, exhibited clear yearly seasonal patterns, likely corresponding to specific planting or growth seasons. Some fertilisers, such as Urea and Muriate of Potash, exhibit trends and short-term fluctuations, while others like Calcium Ammonium Nitrate are more stable. Fertilisers such as Muriate of Potash and total demand have complex demand dynamics, requiring more sophisticated models to capture both seasonal and non-seasonal components. The study recommends exploring other forecasting methods, such as machine learning models and SARIMAX models that account for seasonal and non-seasonal components and external factors in the model.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

ACF	Autocorrelation Function
ADF test	Augmented Dickey-Fuller test.
AIC	Akaike Information Criterion
AR	Auto Regression
ARIMA	Autoregressive Integrated Moving Average
ARMA	Autoregressive Moving Average
BIC	Bayesian Information Criterion
CAN	Calcium Ammonium Nitrate
DAP	Diammonium phosphate
JB TEST	Jarque Bera Test
KPSS test	Kwiatkowski-Phillips-Schmidt-Shin Test.
MA	Moving Average
MAPE	Mean Absolute Percentage Error;
MP	Muriate of Potash
NPK	Nitrogen: Phosphorus: Potassium
OLS	Ordinary Least Squares
PACF	Partial Autocorrelation Function
PAR	Periodic Autoregressive
SARIMA	Seasonal Autoregressive Integrated Moving Average
SPE	Squared Prediction Error
STL	Seasonal-Trend decomposition using Loess.

CHAPTER ONE
INTRODUCTION

1.1 Background of the Study

Over 75% of Kenyans make a living through agriculture (World Bank, 2022). The agricultural sector accounts for 33% of GDP (Mose, 2019). The population in Kenya has been increasing by approximately 2% yearly, with a projection of 63,103942 by 2030 (World Bank, 2022). The implication is that more food will be needed to feed the population. However, agricultural productivity has stagnated in recent years. This stagnation is attributed to various factors, including climate change, suboptimal fertiliser application, and poor fertiliser distribution.

Applying fertiliser during production increased crop production (Ritchie *et al.,* 2022). Applying fertiliser in crop production has increased production by 30% (World Bank, 2022). Different types of fertilisers are used in crop production in Kenya, which include calcium ammonium nitrate (CAN), diammonium phosphate (DAP), calcium nitrate, Muriate of potash, NPK, and Urea, among others. The application of different fertilisers differs depending on various factors such as time of application, type of soil, method of application, and type of crop being produced, among other factors (Ritchie *et al.,* 2022). For this reason, the amount of fertiliser demanded differs with time and application type.

The global market experienced an increasing demand for fertiliser between 2015 and 2019 (Statista, 2024). Kenya, as an agrarian economy, is experiencing dynamic fertiliser consumption trends. More people in Kenya are venturing into agriculture, increasing the amount of fertiliser consumed every production period (World Bank, 2022). In 2015, the amount of nutrient phosphate P_2O_5 consumed was 97.9 metric tonnes, nutrient nitrogen 96 thousand metric tonnes, and nutrient potash K_2O was 31.7 metric tonnes. In 2017, the amount consumed was 160.3, 160.4, and 45.2 thousand metric tonnes of P_2O_5, N, and K_2O, respectively. In 2018, the amount consumed was 131.3, 118.8, and 20 thousand metric tonnes of P_2O_5, N, and K_2O, respectively. In 2019, the amount consumed was 150, 139.6, and 42.7 thousand metric tonnes of P_2O_5, N, and K_2O, respectively (Statista, 2024). The trend indicates that the amount of fertiliser needed in Kenya every production period keeps changing. For planning purposes, it is crucial to

have a reliable way of determining the amount of fertiliser needed for each production period. Determination of future quantities can only be made possible by forecasting, which relates the future amounts to the past fertiliser demand. Various methods are used in forecasting, including time series, econometric methods, judgmental forecasting, and qualitative methods (Zellner *et al.*, 2021).

A study by Tenkorang *et al.* (2008) used simple linear regression to forecast global fertiliser demand in Asia. Tenkorang *et al.* (2008) pointed out that forecasting fertiliser demand is the primary factor that will ensure the success of long-term plans for global food security and the profitability of the fertiliser industry. The study forecasted fertiliser demand in 9 regions in Asia. According to Tenkorang *et al.* (2008), Asia is expected to account for about 40% of the global forecast of 187.7 million Metric tonnes (Mt) in 2015 and 223.1 million Mt in 2030. The short comings of using this method are that linear regression assumes a constant relationship between variables over time, which may not hold in real-world scenarios where time series data often exhibit trends, seasonality, and autocorrelation. Additionally, linear regression does not account for the sequential dependence among observations inherent in time series data, leading to potentially inaccurate forecasts (McQuarrie & Tsai, 1998).

Recent studies conducted to forecast demand in different parts of the world have considered time series analysis one of the best forecasting methods. The Seasonal Autoregressive Integrated Moving Average (SARIMA) model is among the proposed methods. Borkar (2023) used a time series model to forecast fertiliser consumption in India. In his study, data on fertiliser consumption in India was gathered from Agricultural Statistics at a glance from 1950 - 1951 to 2020 - 2021 and utilized to fit the ARIMA model and forecast future fertiliser consumption. The study results revealed that the ARIMA (1,2,1) model was best suited to forecasting India's future total fertiliser use. Bezerra *et al.* (2013) used time series analysis to forecast fertiliser consumption in Brazil. Their study utilized ARIMA and logistic function models to forecast fertiliser demand for the next 20 years. A study by Pisuttinusart *et al.* (2022) used time series modelling to forecast import demand for fertiliser in Thailand. The study used the SARIMA model to forecast the demand for imported fertiliser. The study utilized data from 2008 to December 2021 (168 months). The results indicated that SARIMA (0,0,0)

$(0,1,1)_{12}$, SARIMA (2,0,1) $(0,1,1)_{12}$, and SARIMA (1,0,0) $(2,1,0)_{12}$ were the appropriate models. Sukprasertb *et al.* (2020) used SARIMA models to forecast demand for the import of table grapes in Thailand. The study utilized time series data from January 2007 to April 2020. The study results indicated that the best-fitted forecast model was SARIMA (1,1,3) $(2,1,0)_{12}$.

As an agricultural economy characterized by seasonal modes of agricultural production, Kenya has seasonal fertiliser demand. SARIMA models are one of the methods that are used to forecast seasonal time series data. However, SARIMA models are commonly used in forecasting methodologies in different countries. However, there is a notable absence of specific applications for Kenya. In Kenya, several models have been used to forecast demand. These include modelling the effects of input in Kenya (Megan *et al*., 2016) and applying the hybrid Seasonal ARIMA-GARCH Model in modelling and forecasting fertiliser prices in Kenya (Akinyi, 2023). There is limited information on modelling the quantity of fertiliser demand in Kenya. Therefore, a model is needed to forecast the country's future fertiliser demand.

1.2 Statement of the Problem

Fertiliser is one of the production inputs in crop production and is critical to increasing production. Fertiliser use in crop production increase crop yield by 30 %. Farmers in Kenya rely on fertiliser for crop production. Kenya’s population is projected to reach 63 million by 2030, the demand for food will increase significantly, leading to higher fertiliser needs and putting greater pressure on agricultural systems. More people are venturing into agriculture, making the amount of fertiliser consumed in Kenya continue to increase every production period. However, the fertiliser available to farmers has persistently remained inadequate in the past few years leading to food shortage. Kenya imports its fertiliser, which is one of the leading causes of the fertiliser shortage during production. Getting the correct amount of fertiliser to import will solve the issue of inadequacy. It will be achieved by forecasting the quantity of fertiliser needed each month. Models such as the regression model ARIMA and SARIMA have been used in foresting demand. Although different methods of forecasting demand of fertilizer have been used, the nature of production in Kenya being seasonal, it is imperative to use SARIMA model due to its optimal prediction efficiency.

1.3 General Objective

To investigate the trends in fertiliser demand in Kenya using time series analysis techniques.

1.3.1 Specific Objectives

i. To analyse the trends and seasonal patterns in the monthly fertiliser demand in Kenya
ii. To develop SARIMA models for fertiliser demand in Kenya.
iii. To apply the developed SARIMA model for future fertiliser demand in Kenya.

1.4 Research Questions

i. What are the key seasonal patterns and trends in fertilizer demand in Kenya over the past 12 years?
ii. What are the optimal SARIMA model parameters (p, d, q) × (P, D, Q) s that best fit and forecast the seasonal fertiliser demand in Kenya?
iii. How accurate are the forecasts of fertiliser demand in Kenya when applying the developed SARIMA model, and what are the predicted trends for the next five years?

1.5 Significance of the Study

The model will help the government to import sufficient fertiliser that is adequate for production. Forecasting fertiliser demand in Kenya is crucial in achieving national and global objectives. It improves agricultural productivity thus, stimulates economic growth, optimizes resource allocation, ensures environmental sustainability, stabilizes markets, and informs policy-making. This practice enhances food security, supports sustainable agriculture, fosters economic development, and promotes research and innovation. These benefits align with various Sustainable Development Goals (SDGs), such as eliminating hunger, promoting sustainable farming, advancing Kenya's development, and contributing to global sustainability efforts.

1.6 Assumptions of the Study

i. All quantities imported are not reexported: This assumption posits that once fertiliser is imported into Kenya, it remains within the country, safeguarding the integrity of fertiliser distribution channels for domestic agricultural use.

ii. Quantity imported equals quantity used in crop production. It has assumed that all fertiliser cleared by customs for importation into Kenya is efficiently utilized in crop production activities, minimizing wastage and ensuring effective contribution to agricultural productivity.

CHAPTER TWO
LITERATURE REVIEW

2.1 An Overview of Time Series Analysis

2.1.1 Time Series Analysis

Time series data exhibit distinct patterns and components (Kirchgässner *et al.,* 2012). These components include the trend, which represents the long-term movement of the data (e.g., growth or decline); seasonality, which captures repeating patterns at regular intervals; and cyclic patterns, which are influenced by economic or business cycles. Random fluctuations, known as the irregular or residual component, are also present in time series data.

One of the integral parts of time series is time series modelling. It involves using statistical techniques to build models to predict future values based on historical data. Popular models include autoregressive integrated moving averages (ARIMA), seasonal decomposition of time series (STL), and state space models. Forecasting, on the other hand, is all about predicting future values, which is vital for decision-making and planning in various fields. In addition, data anomaly detection is another critical aspect of time series analysis, which helps identify outliers or unusual observations in time series data, which is essential for quality control and fraud detection (Kirchgässner *et al.,* 2012).

Time series analysis is a fundamental concept in data science that deals with recordings of processes or events that vary over time. These recordings can be either continuous or discrete in nature. When working with time series data, there are several essential components to consider.

One key aspect is smoothing, which involves separating the observed values (Y_t) into two parts: a trend component (η_t) and a random noise component (ε_t). The trend captures the overall movement or tendency in the data, helping analysts identify whether the series is increasing, decreasing, or remaining relatively constant over time. The random noise, on the other hand, represents the unpredictable fluctuations in the data (Ihaka, 2005)

$$Y_t = \eta_t + \varepsilon_t \quad (1)$$

Modelling is another crucial element of time series analysis. It involves developing mathematical models to explain the patterns observed in the data over time. These models typically depend on unknown parameters that need to be estimated from the available observations (Ihaka, 2005).

Forecasting is a primary goal of many time series analyses. Using the historical observations (Y1, Y2, ..., YT), analysts can estimate future values (YT+L, where L is the number of time steps into the future). Importantly, these forecasts should also include an indication of the uncertainty associated with the predictions (Ihaka, 2005). Finally, control is an aspect of time series analysis that focuses on manipulating current observations to influence future outcomes. This concept is particularly useful in applications where there's a need to guide the time series towards more favourable future states.

These components of time series analysis - smoothing, modelling, forecasting, and control - provide a comprehensive framework for understanding and working with time-dependent data across various fields and applications (Ihaka, 2005).

2.1.2 Box Jenkin Methodology

Box Jenkins' (BJ) methodology is an approach and process proposed by George Box and Gwilym Jenkins. It is an approach that entails identification, filling, checking, and using ARIMA models in time series analysis. It is based on four main objectives: model identification, parameter estimation, model checking, and forecasting (Box &Jenkins, 1976).

2.1.2.1 Box Jenkin Determination of Autoregressive Integrated Moving Average model

Stationarity: First, check if the time series is stationary, meaning its statistical properties do not change over time. If it is not stationary, use differencing (subtracting the previous observation from the current one) to make it stationary.

2.1.2.2 Autoregressive Function and Partial Autoregressive Function Plots

Analyse the autocorrelation function (ACF) and partial autocorrelation function (PACF) plots to identify the orders of the autoregressive (AR) and moving average (MA) terms. The ACF plot helps to identify the number of MA terms, while the PACF plot helps to identify the number of AR terms.

2.1.2.3 Parameter Estimation

Maximum Likelihood Estimation (MLE): Typically, parameters are estimated using MLE, which finds the values that maximize the likelihood of the observed data. Least Squares: Alternatively, the least squares method can be used, which minimizes the sum of the squared differences between the observed and predicted values.

2.1.2.4 Box Jenkin model Validation process

Residual Analysis: Examine the residuals (differences between observed and fitted values) for patterns. Ideally, residuals should look like white noise, indicating that the model has captured all information in the data.

Ljung-Box Test: This test checks if the residuals are independently distributed. A high p-value suggests the residuals are like white noise, indicating a good fit.

2.1.2.5 Forecasting using Box Jenkin approach

After validating the model, it forecasts future time series values. The accuracy of these forecasts is assessed using metrics such as mean absolute error (MAE) and mean squared error (MSE). For a time, a series y_t, an ARIMA (p, d, q) model is expressed as:

$$\phi_p(B)\nabla^d y_t = \theta_q(B)\epsilon_t \quad (2)$$

where:

∇^d Represents differencing d times to achieve stationarity.

$\phi_p(B)$ is the autoregressive part of order p.

$\theta_q(B)$ is the moving average part of order q.

ϵ_t denotes white noise error terms.

The Box-Jenkins methodology is a comprehensive approach to time series analysis and forecasting. The model provides a systematic process of model identification, parameter estimation, model checking, and forecasting, and practitioners can create

well-suited models for their data, leading to reliable and accurate forecasts (Box & Jenkins, 1976).

2.2 Characterize the Seasonal Variation and Trend in Monthly Fertiliser Demand

2.2.1 The Concept of Linear Time Series Analysis

Different concepts are used in modelling a time series model, including the trend, seasonal, cyclical, and residues. The trend entails long-term movements, which provide a general way for the data to move over a long period. The seasonal components are the movements that occur in regular cycles; however, each cycle happens yearly. Seasonal movements are usually attributed to natural conditions and sometimes human conditions. Cyclic components refer to the movements in long-term cycles. The residues (sometimes called irregular movements) are those not accounted for by the above three-time series components. The irregular components cannot be predicted due to the off-chance events.

The residue components have a plethora of information, which, when analysed, can provide a family of linear models like Autoregressive (AR), Moving Average (MA), and integrated ARMA (ARIMA). When picked up from the "noise heap" using Ordinary least squares (OLS), the time series components leave out the white noise (OLS leaves the variables iid with $\mu=0$, $->\sigma_i^2= \sigma_j^2$ for $i\neq j;\ i, j=1,2, 3\ldots$) (Lütkepohl, 2006).

Let y_t be a univariate time series in which any physical or geometric variable is observed at a specified period t=1,2, 3…, n. Also, let Ω_{t-1} be the information at time t-1 and contain all the available data needed to forecast future values $y_t, y_{t+1}, y_{t+2}...$ if Ω_{t-1} does not contain the needed information, then the time series from such data is the white noise ε_t. Thus,

$E[\varepsilon_t] = 0,$

$E[\varepsilon_t^2] = \sigma^2,$

$E[\varepsilon_t \varepsilon_s] = 0, \ \forall s \neq t,$

In this case, the time series incorporates the known and unknown time series components. Due to the additive nature of the time series components, we, therefore, have

$y_t = E[y_t \,|\Omega_{t-1}] + \varepsilon_t$ (3)

Cryer (1991) indicates that to predict the component of y_t, the commonly used model is a linear combination of p-lagged values $y_t = \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \cdots + \varphi_p y_{t-p} + \varepsilon_t$, t = 1... n. for φ_i represent unknown parameters. The model is AR (p), where p is the order value. Thus, the above can be simplified using the lag operator as follows.

$\varphi_p(L) y_t = \varepsilon_t$, (4)

The above auto-regression model is derived from the lag operator, which is defined by $L^k y_t = y_{t-k}$, also $\varphi_p(L) = 1 - \varphi_1 L - \cdots - \varphi_p L^p$. When AR (p) is large enough, the approximation is made by taking the ratio of two polynomials. The ratio consequently leads to

$\varphi_p(L) y_t = \theta_q(L) \varepsilon_t$,

Which is the autoregressive moving average model, ARIMA (p, q), where $\theta_q(L)$ is the polynomial of the MA (q).

Cryer, 1991 defines explains white noise as follows,

$$E[Y_t] = \mu$$

$$E[(y_t - \mu)^2] = \gamma_0$$

$$E[(y_t - \mu)(y_{t-k} - \mu)] = \gamma_k, \forall\ k = 1, 2, \ldots$$

for μ, γ_0, and γ_k represents finite real numbers. The above AR (p) model is covariance stationary and depends on the parameters $\varphi_1,\ldots,\varphi_p$.

For the first order AR (1); $y_t = \varphi_0 + \varphi_1 y_{t-1} + \varepsilon_t$. The intercept φ_0 points out that y_t has a mean greater than zero. Thus, y_t can be written in a recursive substitution as

$$Yt = \varphi_1^t y0 + \sum_{i=0}^{t-1} \varphi_1^i \ \varphi 0 + \sum_{i=0}^{t-1} \varphi_1^i \, \varepsilon_{t-1} \quad (5)$$

It follows that $Y_t = \varphi_1^t y_0 + \sum_{i=0}^{t-1} \varphi_1^i \ \varphi_0$. When the absolute value of $\varphi_1 < 1$ it follows that

$\sum_{i=0}^{t-1} \varphi_1^i = \frac{1-\varphi_1^t}{1-\varphi_1} < \infty \ \forall \ t \geq 0$. When t is large, the expected value of y_t portrays the relevant condition for Stationarity; $E[y_t] = \frac{\varphi_0}{1-\varphi_1} = \mu$; $|\varphi| < 1$, when $|\varphi| > 1$, the time series is usually explosive and rarely happens in practice (Cryer, 1991).

2.2.2 Autoregressive Integrated Moving Average Models

Given an AR (p) model, say

$$\Delta_1 y_t = \rho y_{t-1} + \varphi_1^* \Delta_1 y_{t-1} + \varphi_2^* \Delta_1 y_{t-2} + \cdots + \varphi_{p-1}^* \Delta_1 y_{t-(p-1)} + \varepsilon_t \quad (6)$$

such that $\varphi_p(L) = 1 - \varphi_1 L - \cdots - \varphi_p L^p$; AR(p) is considered non-stationary if its characteristic equation has a unit root. The presence of the unit root makes the model autocorrelations vary from time to time, and the impact of the shocks remains permanent. When such a case happens, the AR polynomial becomes $\varphi_p(L) = \varphi_{p-1}^*(L)(1-L)$ where $\varphi_{p-1}^*(L)$ represents the lag polynomial of order p-1, which has a root outside the unit circle. The new variable from (1-L)y_t can be described by the covariance of a stationary AR(p-1) model. Considering a general AR (1) model below

$$y_t - \mu - \delta t = \varphi_1\big(y_{t-1} - \mu - \delta(t-1)\big) + \varepsilon_t \quad (7)$$

This model can also be rewritten as

$$y_t = \mu^* + \delta^* t + \varphi_1 y_{t-1} + \varepsilon_t \quad (8)$$

where $\mu^* = (1-\varphi_1)\mu + \varphi_1\delta$ and $\delta^* = (1-\varphi_1)\delta$. Also, consider$z_t = y_t - \mu - \delta t$, if lagged z_t is recursively substituted as

$$z_t = \varphi_1^t z_0 + \sum_{i=1}^{t} \varphi_1^{t-i} \varepsilon_i \quad (9)$$

where z_0 represents the pre-sample starting value of z_t. When $|\varphi_1| < 1$, the effect of the pre-sample starting value, z_0, decrease, which consequently leads to a significant reduction of the shock effect. When the equation

$$y_t - \mu - \delta t = \varphi_1\big(y_{t-1} - \mu - \delta(t-1)\big) + \varepsilon_t \quad (10)$$

is written in terms of z_t, we have

$$\Delta_1 z_t = (\varphi_1 - 1) z_{t-1} + \varepsilon_t \quad (11)$$

where $\Delta_i = (1 - L^i)$ is the differencing operator. The positive and the negative values of z_t correspond to the y_t, which can be larger or smaller than the trend mean $\mu+\delta t$. Under the above scenario, y_t possesses mean-reverting behaviour. When$\varphi_1 = 1$, there is no mean-reverting behaviour because $\Delta_1 z_t = \varepsilon_t$. Therefore

$y_t = \mu^* + \delta^* t + \varphi_1 y_{t-1} + \varepsilon_t$ (12)

become $y_t = \delta + y_{t-1} + \varepsilon_t$ (13)

In the equation (13), the trend variable is not there. Therefore, the model is a random walk with a δ drift. When recursive substitution is done, the model becomes

$y_t = y_0 + \delta t + \sum_i^t \varepsilon_i$ (14)

In this case, the partial sum time series $S_t = \sum_i^t \varepsilon_i$ is the stochastic trend. The implication is that y_t, in this case, can also be represented by the equation. $y_t = \delta + y_{t-1} + \varepsilon_t$ and it also has a stochastic and deterministic trend (Shumway & Stoffer, 2017).

Any AR (p) y_t model that can be decomposed as $\varphi_{p-1}(L)(1-L)$ is said to have a stochastic trend. A model with a stochastic trend can be made stationary by applying the differencing filter Δ_1. The time series that requires first differencing to eliminate the stochastic trend characteristics is referred to as time series integrated into order one and represented as I (1). Thus, a time series with the first difference can be written below

$y_t = \Delta_1 y_t + y_{t-1} = \Delta_1 y_t + \Delta_1 y_{t-1} + y_{t-2} = \cdots = \Delta_1 y_t + \Delta_1 y_{t-1} + \cdots + \Delta_1 y_{t-k} + \Delta_1 y_{t-k-1}$ (15)

Where $\Delta_1 y_t = y_t - y_{t-1}$. When k is large, the effect y_{t-k-1} gets significantly small. Thus, the observation y_t can be obtained by successive summation (integration) of the mixed process. $\Delta_1 y_t$. Thus, ARIMA (Autoregressive Integrated Moving Average) is a popular time series forecasting model used to analyse and predict data points based on their historical patterns. The notation ARIMA (p, d, q) represents the three main components of the model: autoregressive (AR) terms, differencing (I), and moving average (MA) terms. The "p" parameter refers to the number of autoregressive terms, which capture the relationship between an observation and a certain number of lagged observations. It reflects how a series' current value depends on its previous values. The "d" parameter represents the order of differencing required to make a time series stationary. Differencing involves subtracting the current observation from the previous one, which helps eliminate trends and seasonality (Shumway & Stoffer, 2017). The "d" value indicates the number of times differencing needs to be applied to achieve Stationarity. The "q" parameter denotes the number of moving average terms, which model the dependency between an observation and the residual errors from previous

predictions. It captures the influence of past forecast errors on the current value. Combining these three components, ARIMA models capture the complex dynamics of time series data, including trends, seasonality, and random fluctuations. Thus, ARIMA (p, d, q) is a time series forecasting model that considers autoregressive terms, differencing, and moving average terms to capture the patterns and dependencies in the data and make predictions based on historical information (Shumway & Stoffer, 2017).

2.2.3 Seasonality in Time Series Analysis

Seasonality is an essential concept in time series analysis. It entails recurring patterns or fluctuations observed in time series data within a specific time. Time series data are attributed to regular and predictable variations that repeat over fixed intervals, such as days, weeks, months, or seasons (Box *et al.,* 2015). The seasonality concept is crucial for accurate analysis, forecasting, and decision-making in various fields, including economics, finance, sales, and weather forecasting. Seasonality represents the systematic and predictable patterns within a time series due to calendar effects, weather conditions, cultural events, or business cycles. It manifests as cycles or waves that repeat over a specific period, often shorter than a year. Seasonal patterns can exhibit different forms, including additive or multiplicative effects on the data (Box *et al.,* 2015).

Determining seasonality in time series analysis involves identifying patterns that repeat at fixed intervals within the data (Box *et al.,* 2015). There are various ways of testing whether time series data has seasonality attributes. Among them is visual Inspection (Plotting the time series data and visually examining the patterns can provide initial insights into seasonality). Visual Inspection looks for regular patterns, cycles, or fluctuations at fixed intervals. The second method is through the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF). The ACF and PACF plots help identify the correlation between observations at different lags. Seasonality is indicated by significant spikes or peaks at specific lags in the ACF plot. A gradual decline in the ACF plot after these spikes suggests a seasonal pattern. The PACF plot helps identify the order of the seasonal component.

Another method is time series data Decomposition. Time series decomposition separates the series into trend, seasonality, and residual. Various decomposition methods, such as additive or multiplicative, can be used (Box *et al.,* 2015). Additionally, seasonal subseries Plots can be used to test the seasonality of time series data. This method involves creating subplots of the data for each season and visualizing them separately. If seasonality exists, one should observe similar patterns within each subplot. Also, by calculating the moving average of the data over a fixed window, it is possible to smooth out the noise and highlight the underlying trends (Box *et al.,* 2015). The data presents regular fluctuations around the moving average line if seasonality exists. Seasonal Decomposition of Time Series by Loess (STL) is also a suitable method of testing time series data seasonality. STL, the method, decomposes the time series into trend, seasonality, and remainder components. This method uses a locally weighted scatterplot smoothing technique to estimate these components. Another approach is testing seasonality using Boxplot or Heat map. Plotting the data in a boxplot or heat-map format across different time intervals (e.g., months, quarters, or years) helps visualize the seasonal patterns of time series data. If the data consistently show higher or lower values during specific intervals, it indicates seasonality. Other statistical test includes the Dickey-Fuller (ADF) test or the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test (Box *et al.,* 2015), which is used in the estimation of the autoregressive process of the time series and calculating the test statistic, which measures the distance between the observed series and the trend. The test statistic is then compared to critical values to determine the test's significance.

Given empirical time series data observed in a specific time interval, there is a likelihood that the data will have some seasonal patterns. Similar to the trend features, where the trend depends on the model, there is no specific definition of seasonality. Seasonality may be defined based on characteristics of time series data. The definition depends on whether the observations follow the more or less smooth pattern (S) (Box *et al.,* 2015). When it is possible to estimate the seasonal component of time series by a mathematical curve such as a sinusoid, then the AR time series model is accompanied by a pair of deterministic terms as below

$$y_t = \beta_1 \sin\left(\frac{2\pi t}{S}\right) + \beta_2 \cos\left(\frac{2\pi t}{S}\right) + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \cdots + \varphi_p y_{t-p} + \varepsilon_t \quad (16)$$

Where S represents the number of seasons. A less restrictive form of the model is, therefore, represented as

$$y_t = \varphi_{0,1}D_{1,t} + \cdots + \varphi_{0,S}D_{S,t} + \varphi_1 y_{t-1} + \cdots + \varphi_p y_{t-p} + \varepsilon_t \quad (17)$$

Where $D_{S,t}$ Represent seasonal dummy variables. Suppose N denotes the number of seasonal cycles. Then, the number of observations *is n*=SN. Is such a case, the dummy variable will be defined as

$$D_{S,t} = \begin{cases} 1 \; if \; t = (T-1)S + s \\ 0 \qquad\qquad otherwise \end{cases} \quad \text{Where s= 1, 2… S and T=1, 2…N} \quad (18)$$

To ensure there is a variance of means across different seasons, the intercept of $\varphi_{0,S}$ in equation (16),

is given as $\mu_S = \frac{\varphi_{0,S}}{(1-\varphi_1-\varphi_2-\cdots-\varphi_p}$ (19)

If the seasonal variation is approximately deterministic, then $\hat{\mu}_S \neq \hat{\mu}$ where $\hat{\mu}$ is an estimate of the mean in the AR(p) model with a single intercept (Box *et al.,* 2015).

2.2.4 Stationarity

i. *Strict Stationarity*: A time series $\{Y_t: t \in Z\}$ is regarded as strictly stationary if for any value of k>0 and any time $t_1 \ldots t_k \in Z$, the distribution $(Y_{t1}, \ldots, Y_{tk})$ similar to the distribution of $(Y_{t1} + u, \ldots, Y_{t(k)} + u)$ for each value of u. A time series is strictly stationary if for any time points $t_1, t_2, t_3, \ldots, t_k$ and any shift u, the Joint distribution of $Y_{t(1)}, Y_{t(2)}, .., Y_{t(k)}$ is the same as the joint distribution of $Y_{t(1+u)}, Y_{t(2+u)}, \ldots, Y_{t(k+u)}$. This means that the distribution of the series depends only on the intervals between times, not the specific time points, ensuring consistency in statistical properties like mean and variance across time.

The above information implies that the stochastic behaviour of the process does not change with time and is, hence, strictly stationary. Thus, when Y_t is stationary, then $\mu(t) = \mu(0)$. Additionally, $\gamma(s, t) = \gamma(s - t, 0)$. In stationary time series, the mean function is constant, and the autocovariance varies depending on the time lag between two values used in the computation.

ii. *Weak Stationarity:* A time series is regarded as weakly stationary when $E|Y_t|^2 < \infty$, $\mu(t) = \mu$ and $\gamma(t + u, t) = \gamma(u, 0)$. In this type of stationary time series, it is

possible to have a simplified parameterization of functions of the mean and the autocovariance (Lütkepohl, 2006).

2.2.4.1 Stationarity Test

Stationarity is vital in the analysis of time series data, signifying that the statistical characteristics of a time series remain consistent over time. The two primary tests for stationarity include.

2.2.4.1.1 Augmented Dickey-Fuller Test

The Augmented Dickey-Fuller (ADF) test is a prevalent tool for assessing the stationarity of a time series. The ADF test involves using statistical software or programming languages like Python or R (Baum, 2000).

Procedure:

1. H_0: The time series is not stationarity.
2. H_1: the time series is stationary.

The ADF test statistic is compared against critical values from a table to determine whether to reject the null hypothesis. If the test statistic is lower than the critical value, it leads to rejecting the null hypothesis, suggesting stationarity (Baum, 2000).

2.2.4.1.2 Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test

The KPSS test is another widely utilized method for assessing stationarity, focusing on detecting trend stationarity rather than difference stationarity.

Procedure:

1. The null hypothesis (H_0) assumes the time series is stationary for a deterministic trend.
2. The alternative hypothesis (H_1) suggests that the time series has a unit root and is non-stationary.

The test statistics are obtained using Python or R program. Interpretation is similar to the ADF test. If the p-value is less than the chosen significance level (e.g., 0.05), you will reject the null hypothesis, suggesting that the time series is likely non-stationary.

If the p-value exceeds the significance level, you will fail to reject the null hypothesis, indicating that the time series may be stationary (Baum, 2000).

2.2.4 Periodic Autoregression

Periodic Autoregression (PAR) is a specialized autoregressive model incorporating periodic patterns or seasonal behaviour in time series data (Osborn, 2018). It is designed to capture the cyclic or repetitive nature of certain phenomena that exhibit regular variations at fixed intervals. In a PAR model, the current value of a time series is regressed on its lagged values at specific lags corresponding to the periodicity of the data. Unlike a traditional autoregressive model, where all past values are considered, PAR models focus on a subset of lags that align with the seasonal pattern (Osborn, 2018). The critical parameter in a PAR model is the period, denoted as "s," which represents the length of the seasonal cycle. For example, if the data exhibits a yearly seasonality, s would be set to 12 for monthly or 4 for quarterly data. The period determines the lags at which the model considers the past values for prediction (Franses & Ghiassi, 2009). The PAR model captures the seasonal patterns by estimating the autoregressive coefficients at the specific lags corresponding to the period. These coefficients reflect the relationship between the current observation and the past values that align with the same point in previous cycles. By incorporating the periodicity directly into the model, PAR models can provide more accurate forecasts and capture the seasonal variations that may not be adequately captured by traditional autoregressive models (Franses & Ghiassi, 2009).

According to Franses & Ghiassi, (2009) and Osborn (2018), Periodic Autoregression (PAR) can also be used to analyse demand time series data that exhibit seasonal patterns or cyclic variations. This can be only possible by incorporating the periodic nature of the demand into the model, PAR capture, and quantifying the seasonality, allowing for more accurate analysis and forecasting. When PAR is used in time series analysis, it becomes possible to identify the specific lags corresponding to the demand pattern's periodicity. Estimating autoregressive coefficients at these lags, the PAR model captures the relationship between the current demand and past values that align with the same point in previous cycles, enabling the model to account for the seasonal variations and make more accurate predictions. The model resulting from such analysis

is helpful to businesses and organizations because the model makes it clear to understand the underlying patterns and dynamics of demand fluctuations. It allows for better demand forecasting, inventory planning, resource allocation, and decision-making in retail, manufacturing, transportation, and hospitality industries (Osborn, 2018).

2.2.5 Outliers in Time Series Data

Outliers are observations distinctly different from most data points, either in magnitude or timing (Barnett & Lewis, 1994). Detecting and understanding outliers is crucial in time series analysis as they can significantly impact the analysis and forecasting process. There are several types of outliers in time series data:

The first type of outlier is a Point outlier. These are individual data points significantly different from the rest of the data. Measurement errors, data entry mistakes, or rare events can cause them. The second type of outlier is the Contextual outlier. Contextual outliers occur when a data point is anomalous within a specific context or subset of the data. For example, a sudden increase in demand for fertiliser in a season may be considered a contextual outlier if it deviates from the regular sales pattern. The third type is Collective outliers. These outliers involve a group or cluster of data points that collectively deviate from the expected pattern. They can indicate unusual patterns or events, such as a sudden increase or decrease in stock prices due to market shocks (Barnett & Lewis, 1994).

Detecting outliers in time series data can be done using various techniques, including statistical methods (e.g., z-scores, percentile thresholds), machine learning algorithms (e.g., clustering, classification), and visual Inspection (e.g., scatter plots, box plots) (Filzmoser, *et al.,* 2008). Once identified, outliers can be treated in different ways depending on their nature and impact. They can be removed, adjusted, or analysed separately to understand their underlying causes. Understanding and handling outliers in time series data is essential for accurate forecasting, anomaly detection, and decision-making in various domains such as finance, economics, Agriculture, business, weather forecasting, and industrial process monitoring (Filzmoser*, et al.,* 2008).

When given some data for time series analysis, one of the apparent concerns is whether a given data contains aberrant observations that belong to the time series. Thus, when modelling linear or nonlinear time series data, it is vital to determine whether there are aberrant observations and their effect on modelling and forecasting. There are three common representatives of outliers. They include the additive outlier, the innovative outlier, and the permanent level shift. In the case of an additive outlier (AO), the time series data becomes aberrant because of the causes outside the intrinsic nature of the environment that leads to the generation of the data in question. Given a time series y_t, it becomes a challenge to predict the additive outlier using the historical data set Ω_{t-1}(Chen, 2017). Considering τ as the time when an outlier was taken, then

$$y_t = x_t + \omega I[t = \tau], t = 1,2, \dots n \quad (20)$$

From the above model, I [.] is referred to as the usual indicator variable that yields 0 or 1. The time series x_t is considered uncontaminated or unobserved in such a situation. On the other hand, y_t is the observed variable and the size of the additive outlier isω. The time when the outlier was observed, τ, may be unknown. When OLS is applied in the estimation of the parameters in a time series model, for instance, the AR (1) model for y_t while neglecting the AO, there is a likelihood that the parameter. $\hat{\varphi}_1$ will have a biasing effect on absolute value. In addition, the AO may bring significant skewness and kurtosis due to the observations made at time τ and τ+1. Consequently, there is the possibility that the estimated standard error for the auto-regression parameter will increase with the increase in ω (Chen, 2017).

Another type of outlier is the innovative outliers (IO), which occurs in the noise process. In an ARMA model, it is possible to find that.

$$.\varphi_p(L)y_t = \theta_q(L)(\varepsilon_t + \omega I[t = \tau]) \quad (21)$$

In an AR (1), the time series will be

$$y_t = \varphi_1 y_{t-1} + \varepsilon_t + \omega I[t = \tau] \quad (22)$$

The neglection of the innovative outlier brings about the forecast errors associated with the optimal 1-step ahead prediction given as $\varepsilon_{\tau,\tau-1} = \varepsilon_\tau + \omega \neq 0$. Therefore, the predictor for y_τ becomes biased. Additionally, the predictor $\hat{y}_{\tau+1}$ does not have bias.

Also, the OLS estimate for $\hat{\varphi}_1$ is not affected significantly like in AO. When the parameter $\varphi_1 = 1$, the IO observed at time τ leads to a permanent change in the time series level. Another case is the AR model has a parameter that does not require the above case $\varphi_1 = 1$. Is such a case, the AR model is given as

$$\varphi_p(L)y_t = \varphi_0 + \omega I[t \geq \tau] + \varepsilon_t \quad (23).$$

The value of y_t experiences a shift as follows.

$$\frac{\varphi_0}{(1-\varphi_1-\cdots-\varphi_p)} \text{ to } \frac{\varphi_0+\omega}{(1-\varphi_1-\cdots-\varphi_p)} \quad (24)$$

The presence of aberrant observations in time series data can be indicated by considering the value of the JB normality test (Chen, 2017).

2.3. SARIMA Model of Order (p, d, q) × (P, D, Q) s and Estimate Parameters Using Historical Data

2.3.1 Model Identification

When a restriction is placed onto a linear ARMA (p, q) model, the first step is determining the orders p and q. The specification strategy entails the most significant statistic that suggests the appropriate orders in the autocorrelation function (ACF). From the definition of the autocorrelation function, an ACF of stationary time series is $\rho_k = \gamma_k / \gamma_0 \ \forall$ k = 1, 2, ...

Where γ_k represent the k-th order autocovariance of the y_t defined as E [($y_t - \mu$) ($y_{t-k} - \mu$)] = γ_k, ∀ k = 1, 2, thus, the k-th order auto correlation is defined as

$$\dot{\rho}_k = \frac{\frac{1}{n}\sum_{t=k+1}^{n}(y_t - \mu)(y_{t-k} - \mu)}{\frac{1}{n}\sum_{t=1}^{n}(y_t - \mu)^2} \quad (25)$$

In this case, μ is the mean of the sample y_t. Therefore, the kth-order partial autocorrelation is the correlation between y_t and y_{t-k}. The autocorrelation value is counted from the intermediate observations $y_{t-1, \ldots,} y_{t+1-k}$. Therefore, it is possible to estimate the autocorrelation of order k; AR(k) from the above. Thus

$$y_t - \mu = \psi_1^{(k)}(y_{t-1} - \mu) + \cdots + \psi_k^{(k)}(y_{t-k} - \mu) + \upsilon_t \text{ (Cryer, 1991)} \quad (26)$$

The above AR (k) model is applicable for all values of k on the assumption that υ_t does not have to be the white noise time series. From the above model, the k-th order partial autocorrelation is represented by the coefficient estimate $\psi_k^{(k)}$ (Cryer, 1991).

According to Cryer, 1991, and Shumway *et al.,* 2000 a time series is regarded as the most adequate ARMA (p, q) of order p and q if its estimation is done through comparison of values of (P)ACF with the theoretical values as represented by ARMA model from different p and q. such an approach is, however, complex, especially when the model becomes complicated. For complex models, such as ARMA (4,3), it is challenging to know the correct order of the model based on P(ACF).

Linear time series modelling is another alternative specification strategy. The method is based on a rough guess using linear ACF and performing diagnostic tests on the residues.

2.3.2 Model Estimation

The AR(p) model parameters can be estimated using the OLS approach. Consider the AR(p) model.

$$y_t = \varphi_0 + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \cdots + \varphi_p y_{t-p} + \varepsilon_t \qquad (27)$$

Under weak assumptions on the properties of the white noise, the OLS parameter estimates are consistent and asymptotically normal. The implication of this is that the t-statistics under such assumptions can be used in the estimation of the significance of all $\varphi_1, \varphi_2, \varphi_3 \ldots, \varphi_p$. The mean μ of y_t is therefore given as,

$$\hat{\mu} = \frac{\hat{\varphi}_0}{(1-\hat{\varphi}_1-\hat{\varphi}_2-\cdots-\hat{\varphi}_p)} \quad \text{(Cryer, 1991)} \qquad (28)$$

2.3.3 Diagnostic Tests

There are various ways of determining whether a model is satisfactory. Among the most used methods focus on whether the residue series $\hat{\varepsilon}$ is approximately white noise. It can be done by testing whether the auto-covariance and/or autocorrelation are zero. The first approach is the consideration of the sample ACF individual elements of the residues. Thus

$$\aleph_k(\widehat{\varepsilon}) = \frac{\sum_{t=k+1}^{n} \widehat{\varepsilon_t}\widehat{\varepsilon_{t-k}}}{\sum_{t=1}^{n} \widehat{\varepsilon_t^2}} \quad (29)$$

Where k= 1, 2, … This test determines if r lies within ±1.96/√n, which assumes normality at a 95% confidence level. The second approach tests the joint significance of the *autocorrelation of the m residues* (Ljung, 1978). The test is called portmanteau test statistics, which is given as

$$LB\ (m) = n(n+2)\sum_{k=1}^{m} \frac{r_k^2\hat{\varepsilon}}{n-k} \quad (30)$$

This value is approximately chi-square distributed with *m-p-q* degrees of freedom when m/n is small and m is relatively large. The other method is Determination through the application of the Langrage Multiplier Theory. The approach tests the AR (p) model against the AR(p+r) model. This approach considers the auxiliary regression model.

$$\hat{\varepsilon}_t = \propto_1 y_{t-1} + \cdots + \propto_1 y_{t-1} + \beta_1\hat{\varepsilon}_{t-1} + \beta_r\hat{\varepsilon}_{t-r} + \upsilon_t \quad (31)$$

Where $\hat{\varepsilon}_t$ represent the residues of AR(p) model conditioned at $\hat{\varepsilon}_t = 0 \mid t \leq 0$. The test is used to test the significance of β_i; i=1,2, 3, …r. as nR^2. Where R^2 is the coefficient of Determination. The model also follows a chi-square distribution with r, degrees of freedom (Ljung, 1978). Testing if the residues have constant variance is also critical. The residues are considered homoscedastic if their variances are constant; otherwise, they are considered heteroscedastic. It is, therefore, important to consider the variation properties of the data under study. Although the OLS estimates in AR(MA) parameters show some consistency and are asymptotically normally distributed, the variance-covariance matrix is unique (Carnero *et al.,* 2007).

For this reason, using a t-statistic test may not be appropriate in assessing the individual regressor model. In addition, tests on nonlinearity are influenced by heteroscedasticity. Thus, the application of common asymptotic distribution may not apply. Ljung, 1978 explains that assuming heteroscedasticity may suggest spurious nonlinearity on the data condition mean. Tests on homoscedasticity may be done using McLeod and Li's approach. The test for homoscedasticity using the McLeod and Li approach examines the assumption of constant variance, or homoscedasticity, in a time series model. The test is based on the idea that if the error variance is constant, the squared residuals of a well-fitted model should not exhibit any correlation or patterns over time. It entails

estimating a regression model and then examining the autocorrelation of the squared residuals (Carnero *et al.,* 2007).

A few steps are used to run the McLeod and Li test on homoscedasticity. The first is fitting the Time Series Model: The First is to specify and estimate a time series model, such as an autoregressive integrated moving average (ARIMA) model or an autoregressive conditional heteroscedasticity (ARCH) model, which involves the Determination of the appropriate lag structure and estimating the model parameters using methods like maximum likelihood estimation. The second is obtaining the Residuals. The residues are calculated by subtracting the predicted values from the observed values of the time series (Carnero *et al.,* 2007). These residuals represent the differences between the actual and predicted values of the model.

The next procedure is to square the residuals to ensure they are positive and focus on their magnitude rather than their direction. Next is the Performance of autocorrelation Analysis: Conduct an autocorrelation analysis on the squared residuals by calculating the autocorrelation function (ACF) and partial autocorrelation function (PACF) of the squared residuals. Lastly, the ACF and PACF plots are done to assess the presence of significant autocorrelation in the squared residuals (Carnero *et al.,* 2007). If there are no significant autocorrelation values outside the confidence interval, it suggests that the squared residuals are independent and, therefore, satisfy the assumption of homoscedasticity. The McLeod and Li test is based on the principle that if the squared residuals exhibit significant autocorrelation at certain lags, it indicates the presence of heteroscedasticity in the model. In such cases, the assumption of constant variance is violated, and it may be necessary to modify the model or employ heteroscedasticity-robust methods to obtain reliable parameter estimates and inferential results (Carnero *et al.,* 2007).

2.3.4 Model Selection

Estimation, diagnostic tests, and model identification lead to applicable time series models, which cannot be ignored or rejected when the above procedures are used for elevation. To have the best forecasting model, selecting the best model that provides minimal error during estimation is essential. It can be done by minimizing some

information criteria based on in-sample fit. However, one may consider all the models for out-of-sample forecasting (Shumway *et al.,* 2000). In time series analysis models, the standard coefficient of Determination R^2 is not useful when evaluating the model, but it is considered a function of the parameter value. Akaike and Schwarz have provided better and more appropriate model selection criteria. The approach compares the in-sample fit that uses the residue variance and the number of estimated parameters.

Taking k as the total number of parameters in the ARIMA (k=p+q+1) model to be estimated, the Akaike Information Criterion (AIC) is calculated as

$$AIC\ (k) = n\ln\hat{\sigma}^2 + 2k \quad (32)$$

The AIC formula comes from Bayesian arguments (Cavanaugh & Neath, 2019). From the Bayesian argument

$$AIC\ (k) = n\ln\hat{\sigma}^2 + k\ln n \quad (33)$$

Where$\hat{\sigma}^2 = n^{-1}\sum_{t=1}^{n}\hat{\varepsilon}_t^2 \quad | \hat{\varepsilon}_t^2\ is\ the\ residue\ from\ ARIMA\ model$ and the "n" represents the number of observations or data points in the statistical model. In this case, the values of p and q, which minimize AIC (k) or BIC (k), are considered the best orders for the ARIMA model (Cavanaugh & Neath, 2019). Minimization is done to p and q to a point where k does not surpass a given upper bound. It is worth noting that when *ln* n>2, the BIC penalizes any additional parameter more heavily than the AIC. The implication is that increasing the AR or MA order has to be significant for BIC to favour a more elaborate model (Cavanaugh & Neath, 2019).

When comparing models, the model with the lower AIC is generally considered the best (Acquah, 2010). The AIC reflects both the goodness of fit and the complexity of the model. It balances these two factors, avoiding over fitting (which can occur when a model is overly complex and fits the noise in the data) while still capturing the essential patterns and relationships.

The difference in AIC values between the two models, referred to as ΔAIC, can also be informative. If ΔAIC is small (e.g., less than 2), it suggests that the models have similar performance (Acquah, 2010). On the other hand, a larger ΔAIC indicates a more substantial difference in model quality, with the model having the lower AIC being

favoured. It is important to note that the AIC is a relative measure and should not be used as an absolute measure of model quality (Penn state college, 2018). It provides a useful tool for model comparison within a specific set of candidate models. Other information criteria, such as the Bayesian Information Criterion (BIC), provide similar concepts with slight variations in the penalty for model complexity (Acquah, 2010).

2.3.5 Out-of-sample Forecasting and Model Selection

Out-of-sample forecasting is the process of evaluating the performance of a predictive model on data that it has not seen during the training phase. The goal is to assess how well the model generalizes to unseen data and make predictions for future observations. Various evaluation metrics can be used to assess the model's performance, such as

$$MAE = \frac{1}{n}\sum_{i}^{n}\left|Y_i - \widehat{Y}_i\right| \quad (34)$$

On the other hand,

$$RMSE = \sqrt{\sum_{i=1}^{n}\frac{(\hat{y}_i - y_i)^2}{n}} \quad (35)$$

The coefficient of determination $R^2 = 1 - \frac{SSR}{SST}$

where SSR is the sum of squared residuals, and SST is the total sum of squares. These metrics measure the model's accuracy and goodness of fit on the testing data, helping to determine its predictive capability (Box, *et al.,* 2015). The primary purpose of model specification in time series analysis is to forecast future values. Let $\hat{y}_{t+h|t}$ be the forecast of y_{t+h} $made\ at\ time\ t.$ The value is associated with the prediction error, which is calculated as $e_{t+h|t} = y_{t+h} - \hat{y}_{t+h|t}$. As explained by Cavanaugh and Neath (2019), it is possible to obtain different forecasts. y_{t+h} as long as many values of $\hat{y}_{t+h|t}$ are available. In time series analysis, the parameters are selected based on the criteria that the parameter with minimal variance in the forecast is selected as the best parameter for estimating. $\hat{y}_{t+h|t}$which minimizes the *squared prediction error* (SPE). The squared prediction error is given as $SPE(h) = E[e^2_{t+h|t}]$ which consequently turns to be the conditional expectation of the time series y_{t+h} at a given time t, thus $\hat{y}_{t+h|t} = E[y_{t+h}|\Omega_t]$ (Cavanaugh & Neath, 2019 and Aue, & Burman, 2023).

Cavanaugh & Neath (2019) explain that ARMA models forecast for various horizons h are obtained by applying a recursive relationship. Cavanaugh & Neath, 2019, provides the general formulae for the optimal forecast is given as

$\hat{y}_{t+h|t} = \varphi_1 \hat{y}_{t+h-1|t} + \varphi_1 \hat{y}_{t+h-2|t}$. It is, therefore, possible to obtain the optimal forecast given. $\hat{y}_{t+i|t} = \hat{y}_{t+i}$ for i≤0. The most crucial step in getting an expression for SPE from an ARMA (p, q) Model is expressing the MA (∞) model. Thus, the MA (∞) for Y_t can be written as

$y_t = \varepsilon_t + \eta_1 \varepsilon_{t-1} + \eta_2 \varepsilon_{t-2} + \eta_3 \varepsilon_{t-3} + \cdots + \eta_h \varepsilon_{t-h}$ Where $SPE(h) = \sigma^2 \sum_{i=0}^{h-1} \eta_i^2$ | $\eta_0 = 1$. (36)

It is vital to note that SPE (Sum of Squared Prediction Errors) is a measure used to assess a model's goodness of fit or predictive accuracy. In an ARMA (p, q) model context, SPE quantifies the discrepancy between the predicted and actual values of the time series. By calculating the SPE, one can assess the accuracy of the ARMA model in capturing the variations in the time series data. A lower value of SPE indicates a better fit or predictive accuracy, while a higher value suggests more significant discrepancies between the predicted and actual values. It is, therefore, another alternative metric for AIC and BIC, which provides valuable insights into the model's goodness of fit and predictive capabilities.

2.3.6 Jarque-Bera Normality Test

The Jarque-Bera (JB) normality test is a statistical test used to determine if a given sample of data follows a normal distribution. It is based on the skewness and kurtosis of the data. The test calculates a test statistic, the Jarque-Bera statistic, which measures the deviation from normality. If the sample is drawn from a normal distribution, the test statistic will approximately follow a chi-square distribution with two degrees of freedom (Jarque & Bera, 1987; Kim & White, 2004). The p-value associated with the test statistic is then used to determine the statistical significance of the deviation from normality. A small p-value suggests that the data significantly deviates from a normal distribution.

$$JB = \frac{n}{6}\left(S^2 + \frac{(K-3)^2}{4}\right) \quad (37)$$

JB represents the Jarque-Bera test statistic, n is the sample size, S is the sample skewness, and K is the sample kurtosis (Jarque & Bera, 1987; Kim & White, 2004). The test statistic JB follows an approximate chi-square distribution with 2 degrees of freedom under the null hypothesis of normality. To determine the statistical significance of the deviation from normality, the test statistic is compared to the critical values from the chi-square distribution table or using a p-value. A small p-value (usually below a specified significance level like 5%) indicates that the null hypothesis of normality is rejected, suggesting that the data significantly deviate from a normal distribution (Jarque & Bera, 1987; Kim & White, 2004).

2.4 Application of SARIMA model in Forecasting of Events

2.4.1 Arima Vs Sarima Models

2.4.1.1 ARIMA Models

ARIMA models are statistical models used for time series forecasting and analysis. The acronym ARIMA stands for Autoregressive Integrated Moving Average. This model consists of three parts, each representing a different aspect of the time series data. The general presentation of the ARIMA model is ARIMA (p, d, q) (Shumway *et al.,* 2017).

Autoregressive (AR) The AR part of an ARIMA model refers to the autoregressive component. This component captures the relationship between the current value of a time series and its past values. It accounts for the influence of past observations on the present observations. The "p" in ARIMA (p, d, q) indicates the order of the autoregressive component, denoting the number of lag values of the time series considered.

Integrated (I) part: This part represents differencing. Differencing is achieved by subtracting the previous observation from the current one to make the time series stationary. Stationarity is a key concept in time series analysis because many statistical methods, including ARIMA, assume the data is stationary. The "d" in ARIMA (p, d, q) denotes the order of differencing in a model to make the time series data stationary.

Moving Average (MA): The MA component models the relationship between the current value of a time series and past white noise or error terms. White noise is residual errors from previous predictions. The "q" in ARIMA (p, d, q) indicates the order of the moving average component, representing the number of lagged error terms used in the model.

Combining these three components, ARIMA models capture the complex dynamics of time series data, including trends, seasonality, and random fluctuations. Thus, ARIMA (p, d, q) is a time series forecasting model that considers autoregressive terms, differencing, and moving average terms to capture the patterns and dependencies in the data and make predictions based on historical information (Shumway *et al.,* 2017).

The general equation for an ARIMA (p, d, q) model is expressed as follows:

$$Y_t = C + \varphi_1 Y_{t-1} + \varphi_2 Y_{t-2} + \cdots + \varphi_p Y_{t-p} - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q} + \varepsilon_t \quad (38)$$

where,

Y_t is the value of the time series at time t,

C is a constant or drift term (only included if the time series has a trend component).

p represents the order of the autoregressive (AR) component, and φ_i are the autoregressive coefficients.

d is the order of differencing, indicating the number of times differencing applied to make the time series data stationary.

q is the order of the moving average (MA) component, and θ_i are the moving average coefficients.

2.4.1.2 Seasonal Autoregressive Integrated Moving Average Models

Seasonal ARIMA (SARIMA) models are an extension of the classic ARIMA models, designed to handle time series data with seasonal patterns. They combine the concepts of autoregressive (AR), integrated (I), and moving average (MA) components with seasonal components. SARIMA models are beneficial for modelling and forecasting data that recur on time-based patterns.

In a SARIMA model, the "S" component denotes the seasonal part, which introduces additional parameters to capture seasonal variations. SARIMA model is denoted as SARIMA (p, d, q) (P, D, Q), where the lowercase (p, d, q) represents the non-seasonal components, and the uppercase (P, D, Q) _s represents the seasonal components (Shumway *et al.,* 2017). The parameters (p, d, q) follow the same principles as in ARIMA models, while the seasonal parameters (P, D, Q) are applied to the seasonal differences (Otu *et al.,* 2014). SARIMA models provide a robust framework for modelling and forecasting time series data with complex seasonal patterns, making them valuable in economics, finance, and meteorology.

The general equation for a Seasonal ARIMA (SARIMA) model is an extension of the standard ARIMA model and is expressed as

$$Y_t = C + \varphi_1 Y_{t-1} + \varphi_2 Y_{t-2} + \cdots + \varphi_p Y_{t-p} - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q} + \varepsilon_t - \phi_1 Y_{t-s} + \phi_2 Y_{t-2s} + \cdots + \phi_P Y_{t-Ps} - \Theta_1 \varepsilon_{t-s} - \Theta_2 \varepsilon_{t-2s} - \cdots - \Theta_Q \varepsilon_{t-Qs} + \varepsilon_t \quad (39)$$

Where,

P and Q are the orders of the seasonal AR and MA components with corresponding coefficients. Φ_i and Θ_i.

2.4.2 Case Studies on Seasonal Autoregressive Integrated Moving Average Models

Pisuttinusart's (2022) study forecasted the import demand for chemical fertilisers in Thailand, specifically focusing on nitrogen, potassium, and compound fertilisers. The analysis relied on secondary time series data spanning January 2008 to December 2021, covering 168 months. The chosen forecasting technique for this investigation was the Seasonal Autoregressive Moving Average (SARIMA). Upon applying SARIMA models to the data, the empirical results indicate the most suitable models for predicting the import demand for nitrogen fertiliser, potassium fertiliser, and compound fertiliser. These models, determined by the lowest values of the Akaike and Schwarz criteria, are identified as SARIMA (0,0,0) (0,1,1)12, SARIMA (2,0,1) (0,1,1)12, and SARIMA (1,0,0) (2,1,0)12, respectively. In terms of forecasting, the results for the next 12 months (January to December 2022) study indicated a 5.12% increase in the import demand for nitrogen fertiliser, an 8.74% decrease for potassium fertiliser, and a 4.74% increase for

compound fertiliser. These findings provide valuable insights into the anticipated trends in Thailand's chemical fertiliser import market for the upcoming year (Pisuttinusart, 2022).

A study by Sukprasertb *et al.,* 2020 used time series analysis to forecast the import demand for table grapes in Thailand utilizing monthly time series data from January 2007 to April 2020. Stationarity is assessed through the Augmented Dickey-Fuller (ADF) unit root test, and the Seasonal Autoregressive Integrated Moving Average (SARIMA) model was used for forecasting. The study's results indicate that the time series required first-order differencing for both non-seasonal and seasonal components. The optimal SARIMA model identified from the study was SARIMA (1,1,3) (2,1,0)12. The forecasted period for the subsequent eight months has suggested a slight decrease in the import demand for table grapes in Thailand, averaging 11.398 per cent. The overall anticipated decrease for 2020 is projected to be around 15.218 per cent (Sukprasertb, 2022).

Danielle *et al.* (2021) aimed to predict water demand in Joinville, Southern Brazil, using time series analysis for various water consumption categories from January 2013 to December 2017. This study used two models, Exponential Smoothing (ETS) (M, A, A) and Seasonal ARIMA (SARIMA) (1,0,0) (1,1,0) [12], with drift for residential water consumption. The ETS model demonstrated normally distributed residuals and no autocorrelation, whereas the SARIMA model's residuals were not normally distributed and were excluded. In the commercial category, both ETS (M, Ad, N) and SARIMA (1,1,0) (1,0,0) [12] models exhibited normal residuals without autocorrelation. Similarly, for industrial water consumption, ETS (A, N, A) and SARIMA (0,1,1) (0,1,1) [12] models were used with normally distributed residuals and no autocorrelation. In the public category, the results of the study ended up rejecting ETS (M, N, N) due to potential autocorrelation, while SARIMA (0,0,1) (1,0,0) [12] with non-zero mean was considered suitable. For total water consumption, ETS (A, N, N) showed autocorrelation in residuals, while SARIMA (1,1,1) (1,0,1) [12] was considered adequate. The residential ETS model displayed the best performance with a 1.89% Mean Absolute Percentage Error (MAPE), while SARIMA models were suitable for commercial, industrial, public, and total categories. Despite a higher training MAPE in

the public category (5.32%), the SARIMA model showed a less effective forecast during the accuracy-check stage (15.74%). The implication of this study suggests that the SARIMA model, due to its capability to capture seasonality and trends, is a valuable tool for forecasting water demand in Joinville, Southern Brazil, and may find applications in forecasting demand.

Various diagnostic tests and statistical measures were utilized in a 2021 study by Manigandan *et al*. to evaluate the effectiveness of SARIMA and SARIMAX models in predicting monthly natural gas production and consumption from January 1983 to January 2021. Unit root tests, including ADF, KPSS, and PP, confirmed that the time series data became stationary after undergoing first-order and seasonal integration. Analysis of the autocorrelation function (ACF) and partial autocorrelation function (PACF) graphs revealed effective non-seasonal and seasonal autocorrelations in both the natural gas production and consumption sectors. The best-fitting models were determined to be NG production SARIMA (4,1,1) (1,0,1)12 and consumption SARIMA (2,1,1) (5,0,0)12 based on significant parameters and coefficient statistics. These models demonstrated statistical significance with p-values consistently exceeding 0.05. The adequacy of these models was confirmed by LB (Q) test statistics (0.01), showing p-values of 0.90 for production and 0.91 for consumption. Although the Ljung–Box test surpassed the 0.05 p-value, leading to rejecting the null hypothesis, autocorrelation in the SARIMA model residuals was observed. Diagnostic tests on SARIMAX models (2,1,0) (1,0,2)12 and (1,1,0) (1,0,1)12 demonstrated stationary time series residuals, adhering to a normal distribution. In this study, SARIMA models performed better in the training set than SARIMAX.

The scholarly literature has consistently recommended the effectiveness of the SARIMA (Seasonal Autoregressive Integrated Moving Average) model as a top preference for forecasting seasonal data demand. SARIMA incorporates autoregressive and moving average elements alongside seasonal factors, making it well-suited to capture and predict patterns in time series data demonstrating seasonality. The SARIMA model's ability to adapt to and capture seasonal trends positions it as a reliable and robust choice for predicting demand in scenarios where seasonality plays a crucial

role. The literature indicates that SARIMA consistently ranks among the best-performing models in forecasting data demand with distinct seasonal patterns.

CHAPTER THREE
RESEARCH METHODS

3.1 Study Site

The study was conducted at Chuka University, located in Tharaka-Nithi County, Kenya. Chuka University is situated in the eastern region of Kenya, approximately 186 kilometres northeast of Nairobi, the country's capital. The precise location of the university is at coordinates -0.3195457275211977, 37.65747570701258, which places it on the eastern slopes of Mount Kenya. Specifically, the data analysis was conducted in the postgraduate library room of Chuka University.

3.2 Study Design

This study used a longitudinal observational research design to investigate the patterns of fertiliser demand in Kenya from 2010 - 2022. Opting for a longitudinal method enabled the analysis of the changes and trends within the same subjects (months) across the extended timeframe of 12 years.

3.3 Data Collection

The study used secondary data from the Ministry of Agriculture and Livestock Development (Kenya). The goal was to rely on the Ministry's credibility and national coverage to secure accurate and representative data for fertiliser demand within 156 months. This study encompasses monthly observations of fertiliser consumption in Kenya, spanning the period from 2010 to 2023. Box Jenkins methodology requires a minimum of 50 observations (Box & Jenkins, 1976); therefore, the analysis of 156 months was sufficient for modelling the demand.

3.4 Data Analysis

The entire Data analysis is conducted using R studio software program version 2023.12.1+402. The analysis uses Box Jenkins methodology. The research involved application of the Box-Jenkins methodology to develop a Seasonal Autoregressive Integrated Moving Average (SARIMA) model for time series analysis. The SARIMA model used in the research was a univariate model, focusing on a single variable—fertilizer demand. The process began with data visualization to identify seasonal patterns and trends within the dataset. Model building followed, where the appropriate

parameters were determined using autocorrelation (ACF) and partial autocorrelation (PACF) plots. To ensure stationarity, the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test was conducted. After model parameters were estimated, diagnostic checks were performed, including residual analysis. The normality of the residuals was assessed using the Jarque-Bera (JB) test. Once the residuals were confirmed to follow a normal distribution, the SARIMA model was used for forecasting, and the accuracy of the model tested by comparing the actual data with forecasted data. In addition, Shapiro-Wilk normality test was done to test the normality of the residuals.

3.4.1 Data Visualization

Five data visualization methods were used to visualize the characteristics of time series data. These includes GG seasonal plots, ACF and PACF plots, GG subseries plots, and STL series plot.

3.4.2 Time Series Plots

A time series plot is a fundamental and intuitive method for visualizing the overall pattern and trends in time series data. It entails the creation of a chart where the x-axis represents the months (or time points), and the y-axis represents the fertiliser demand. This plot represented the fluctuations, trends, and seasonality in the monthly fertiliser consumption. This visualization played a crucial role in identifying the data's cycles, trends, or irregularities.

3.4.3 Stationarity

In order to test whether the data is stationary or non-stationary, the Kwiatkowski-Phillips-Schmidt-Shin test (KPSS) Test was used. KPSS was done on the data using the R program to formally test for the presence of a unit root in the time series. The KPSS test assessed whether a time series was stationary around a trend.

3.4.4 Model Identification

Model identification is crucial in analysing time series data, particularly when employing SARIMA modelling. The process of model identification involved the determination of the appropriate orders of autoregressive (AR), differencing (I),

moving average (MA), and seasonal components (S) for the SARIMA model. The model identification involved the following steps using the R program.

Identification of Seasonality and Trend: A visual examination of time series data was done to spot any seasonal patterns or trends. Some of the characteristics checked are the consistent fluctuations or systematic alterations in the data over time, indicating the potential presence of seasonality and trend components.

Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF): ACF and PACF plots of the time series data were also generated to uncover potential autoregressive (AR) and moving average (MA) orders. ACF was used to identify the MA order by illustrating the correlation between the time series and its lagged values. PACF, on the other hand, assisted in pinpointing the AR order by displaying the correlation between the time series and its lagged values, taking into account the effects of intermediate lags.

Differencing: The order of differencing (d) was done to achieve stationarity in the time series data. The number of differencing was determined using the R program.

3.4.5 Model Fitting

Model fitting entails choosing the suitable parameters for the designated SARIMA model based on recognized patterns and features within the data. The process of model fitting will entail the following procedure.

i. Initiate the SARIMA Model: The initiation of the SARIMA model entails utilizing the identified values for autoregressive (AR), differencing (d), moving average (MA), seasonal autoregressive (SAR), seasonal differencing (D), and seasonal moving average (SMA) components to initialize the SARIMA model. The R program determined p, d, q, P, D, and Q values.
ii. Training the Model. In this step, time series data was divided into training and testing sets, with the former used to estimate model parameters and the latter reserved to appraise the model's performance. The SARIMA model was then fitted using the training set and the actual set to see if the fitted had a significant deviation from the actual dataset.

iii. Optimize Parameters: R program optimization algorithms will identify parameter values, minimizing the disparity between predicted and actual values in the training set. The model order was autogenerated from "auto.Arima" function in R studio on the basis of AIC and BIC minimisation. The model with the least AIC and BIC value was considered best Model for forecasting fertiliser demand.

iv. Evaluate the Model: The performance of the fitted model was done using the testing set by comparing its predictions against the actual values.

v. Forecasting: The obtained satisfactory model was used to make forecasts.

The resulting model from the above procedure will result in a time series model of the form.

$$
\begin{aligned}
&Y_t \\
&= C + \varphi_1 Y_{t-1} + \varphi_2 Y_{t-2} + \cdots + \varphi_p Y_{12-p} - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{12-q} + \varepsilon_{12} \\
&- \phi_1 Y_{t-s} + \phi_2 Y_{t-2s} + \cdots + \phi_P Y_{12-Ps} - \Theta_1 \varepsilon_{t-s} - \Theta_2 \varepsilon_{t-2s} - \cdots - \Theta_Q \varepsilon_{12-Qs} \\
&+ \varepsilon \qquad (40)
\end{aligned}
$$

3.4.6 Seasonal Decomposition of Time Series

The Seasonal Decomposition of Time Series (STL) method is effective if time series data exhibit long-term trends and seasonal patterns. Therefore, it provides a way to separate these components, allowing a better understanding of the individual contributions and analysing each component separately.

To apply Seasonal Decomposition of Time Series (STL) in R, these steps were utilized: Step 1: Preparation of time series data in R to ensure that the data is suitable for carrying out time series analysis using the R program.

Step 2: Decomposition of time series data breaks down the time series data into its components: trend, seasonality, and remainder. The trend represents the long-term pattern or direction in the data, while the seasonality captures the regular and repeating patterns that occur at fixed intervals. The remainder is the residual component, representing the variability and noise that cannot be attributed to the trend or seasonality.

Step 3: Visualization of the Components after the decomposition. This step helped in understanding how the trend and seasonality contributed to the overall behaviour of the data and how data changed over time.

Step 4: Analyses and use of time series Components were done separately; each component was analysed independently.

3.4.7 Parameter Estimation and Model Validation

Mean Squared Error (MSE) statistical metrics validated the model's performance. Model validation is essential because it ensures that the model accurately captures the patterns and variations in the time series data.

3.4.8 Mean Squared Error

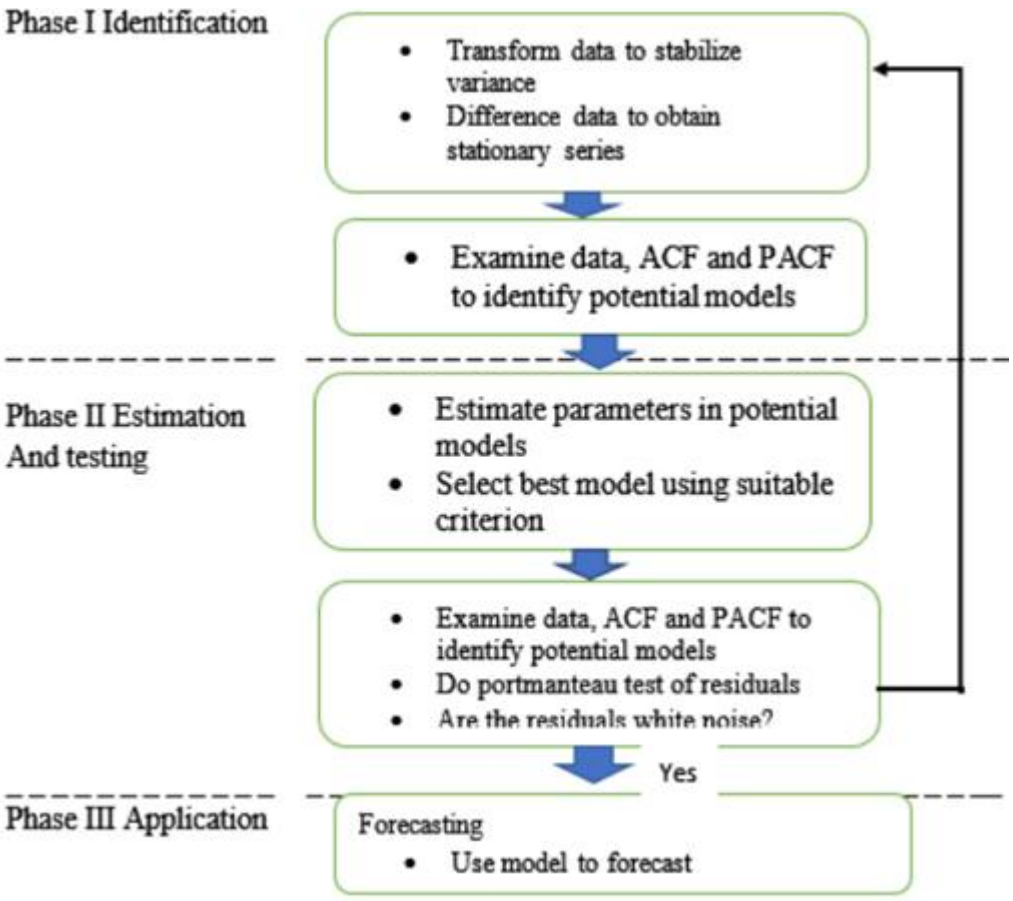
The Mean Squared Error (MSE) is a performance metric used to evaluate the accuracy of forecasts in time series data analysis. It quantifies the average squared difference between the forecasted values and the actual observations. A lower MSE indicates that the forecasted values closely match the actual data points, implying a more precise and reliable forecasting model. The MSE is a common measure for comparing the performance of different forecasting techniques and selecting the most accurate model. Determination of MSE was done using the R program.

3.4.10 Shapiro-Wilk Normality Test

The model identified will be run through the Shapiro-Wilk normality test to evaluate the adequacy of a time series model, particularly in forecasting. This test will be used to assess whether the residuals are normally distributed.

3.4.11 Forecasting and Uncertainty Estimation

After validating the time series model, future values of fertiliser demand in Kenya were done to predict the time series data beyond the observed period, preferably two years. Prediction intervals were based on the characteristics of the time series variations to take care of the uncertainty associated with the forecasts.



Reprinted from "Comparative study of exponential smoothing models and Box-Jenkins ARIMA model of partitioned data of daily stock prices of the CRDB Bank in Tanzania" by Saxena, K. K., & Kamnge, J. S., 2020.

Figure 1: Summary of Box Jenkins procedures

3.5 Time Series Modelling as the Solution to Addressing Fertiliser Demand in Kenya

Time series modelling is the best approach to solve the problem of fertiliser demand for several compelling reasons. Time series modelling is designed to capture and understand patterns and fluctuations occurring over time. When applied to fertiliser demand, it considers the cyclicality, seasonality, and long-term trends that influence agricultural practices. The importance of time series modelling in addressing fertiliser demand lies in its ability to provide accurate, data-driven predictions that guide decision-making with minimal error. The SARIMA model to forecast fertiliser demand in Kenya is expressed in Equation 41.

$$Y_t = C + \varphi_1 Y_{t-1} + \varphi_2 Y_{t-2} + \cdots + \varphi_p Y_{10-p} - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{10-q} + \varepsilon_{10} - \phi_1 Y_{t-s} + \phi_2 Y_{t-2s} + \cdots + \phi_P Y_{10-Ps} - \Theta_1 \varepsilon_{t-s} - \Theta_2 \varepsilon_{t-2s} - \cdots - \Theta_Q \varepsilon_{10-Qs} + \mathcal{E} \quad (41)$$

where

P and *Q* are the orders of the seasonal AR and MA components, with corresponding coefficients Φ_i and Θ_i.

Y_t is the value of the time series at time t,

C is a constant or drift term (only included if the time series has a trend component).

p represents the order of the autoregressive (AR) component, and φ_i are the autoregressive coefficients.

q is the order of the moving average (MA) component, and θ_i are the moving average coefficients.

The model captures Temporal Dependencies. Fertiliser demand is inherently tied to the agricultural calendar. Different crops and regions have specific planting and harvesting seasons. Time series models are well-suited to capture these temporal dependencies, considering that fertiliser demand fluctuates predictably with these patterns (Brockwell, 2016).

3.6 Ethical Considerations

The Chuka University Ethics and Research Committee and the National Council for Science, Technology, and Innovation (NACOSTI) sought ethical clearance for the research before the study's commencement (Appendix 1). This study took into account various ethical considerations. The study upheld the privacy and confidentiality of datasets provided by the Ministry of Agriculture and Livestock Development. The data was used in the analysis as provided, and no alteration or manipulation was made during the modelling process. Therefore, this study was committed to responsible data use, refraining from misuse, unauthorized sharing, or secondary use beyond the agreed-upon scope. The study was also dedicated to minimizing harm, conducting impartial and objective research, and adhering to relevant data laws and ethical guidelines. The research findings presented in this document have prioritized accuracy and clarity to

avoid sensationalism and present a truthful representation of the demand. Through these ethical commitments, the study aims to contribute responsibly to understanding fertiliser demand patterns in Kenya while prioritizing the well-being and rights of all stakeholders involved.

CHAPTER FOUR
RESULTS AND DISCUSSION

4.1 Descriptive Statistics

The descriptive statistics for the different fertilisers are summarized in Table 1. The mean ranged from 966.19 in Muriate of Potash to 17906.31 in DAP. The median ranged from 469.15 in Muriate of Potash to 1751 in CAN. The minimum values were all zero across the fertilisers, while the maximum values ranged from 19557 in Muriate of Potash to 100598 in DAP.

The results indicate that DAP, Nitrogen, Phosphorus and Potassium Fertiliser (NPK), and Urea are the main fertilisers that are highly consumed in Kenya; DAP being the most consumed input with a mean consumption of 17906.31 metric tonnes, NPK has mean consumption of 12296.69, and urea having an average consumption of 6522.18 metric tonnes. In some months, they experienced a demand of 0. The variation of the amount needed is determined by factors such as fertiliser type, the crop being produced, and the time of application. Some crops require specific fertiliser types and do better in long than short rains (Bal *et al*., 2022). Therefore, matching the right type of fertiliser with the specific needs of crops and considering the seasonal rainfall patterns are key to optimizing crop yields. Farmers must carefully plan their fertiliser use to align with these conditions in Kenya, where the long and short rains have distinct characteristics. Table 1 is the summary statistics of the average consumption of each type of fertiliser for the years 2010 to 2023.

Table 1: Descriptive statistics

Variable	n	Mean	Median	Min	Max	Skewness
CAN	168	8853	1751	0	48164	1.52
Calcium nitrate	168	1700.21	1264	0	28164	7.23
DAP	168	17906.31	10962	0	100598	1.4
Muriate of potash	168	966.19	469.15	0	19557	5.7
nitrogen, phosphorus and potassium fertiliser	168	12296.69	4341	0	88876	1.73
Urea	168	6522.18	1256	0	47400	1.72
Total Fertiliser	168	54662.83	49026.07	2339.18	210293	0.9

4.2. Visualization of the Series

4.2.1 Visualisation of Calcium Ammonium Nitrate demand

A time series plot illustrating the monthly imports of calcium ammonium nitrate (CAN) from January 2010 to 2023 is presented in Figure 2. The data reveals a distinct pattern of seasonal fluctuations, characterized by recurring peaks that suggest periods of heightened fertiliser demand. Notably, these peaks are more pronounced during specific periods, such as 2014 - 2015 and 2017 – 2018. While there has been no consistent upward or downward trend over the years, the variability in the data is evident, particularly in the last part of the series. This period shows more intermittent imports, possibly reflecting changes in market dynamics, supply chain disruptions, or regulatory influences.

Furthermore, the graph highlights intervals of zero demand, suggesting months with no CAN inflow, which could be attributed to seasonal demand shifts, logistical challenges, or other external factors. The graph indicates that the demand for CAN fertiliser is seasonal. The demand for CAN changes occasionally, with some months having 0 demand.

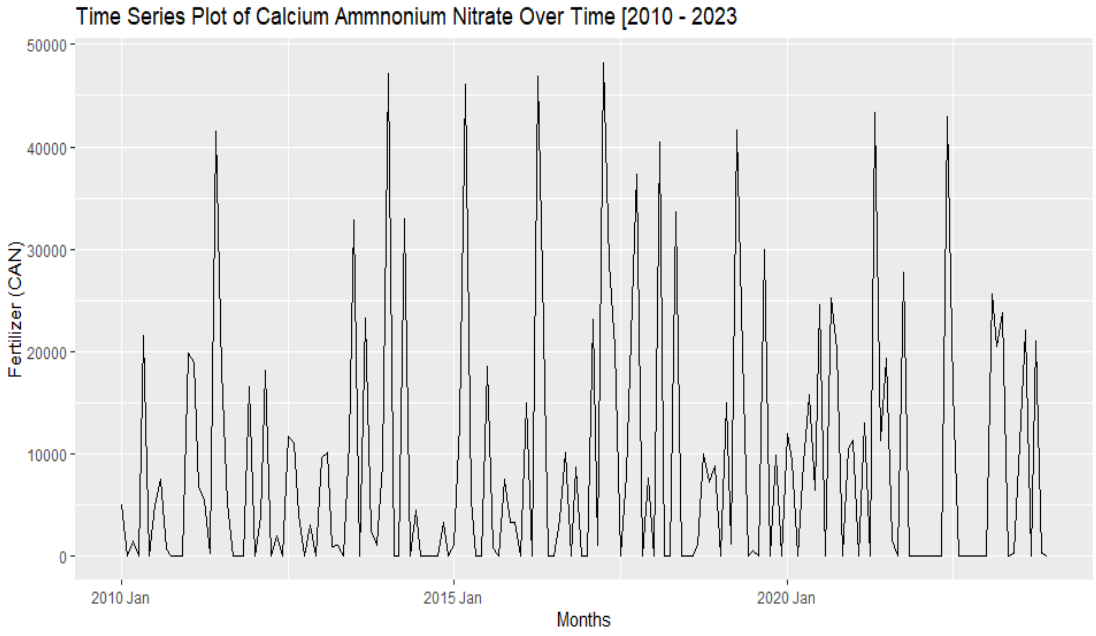


Figure 2: Calcium ammonium nitrate demand time series plot

4.2.1.1 Normality Test of Calcium Ammonium Nitrate Data

The Jarque-Bera test was used to evaluate the normality of the time series data for Calcium Ammonium Nitrate demand. The test produced a test statistic (χ^2) of 82.267 with 2 degrees of freedom, accompanied by a p-value reported as less than 2.2×10^{-1}, leading to the rejection of the null hypothesis that time series follows the normal distribution. Consequently, this departure from normality will be accounted for in subsequent analyses and interpretations, as it impacts the validity of statistical inferences drawn from the data. Summary statistics for the Jarque-Bera test are presented in Table 2. The significant deviation from normality suggests that the CAN demand time series exhibits characteristics that differ from a normal distribution. This could be due to factors such as outliers, non-linear trends, or structural breaks in the time series.

Table 2: Jarque Berra test statistics for calcium ammonium nitrate

Jarque-Bera test		
X-squared	d.f	P value
82.267	2	< 0.0001

4.2.1.2 Seasonality Plot Checks on Calcium Ammonium Nitrate Time Series

Figure 3 "Time Series Plot (gg Season plot) illustrates the seasonal variation in the demand of Calcium Ammonium Nitrate (CAN) across multiple years. Each coloured line (Appendix 4) represents a specific year, with the x-axis denoting the months and the y-axis representing the quantity of CAN demanded. The plot demonstrates a seasonal pattern, with certain months, such as March, April, and September, consistently showing higher demand levels across different years. This seasonality suggests a cyclical nature in the demand for CAN, possibly attributed to the agricultural cycles or market dynamics. Additionally, the plot highlights significant inter-annual variation, with years like 2014 and 2019 displaying notably high peaks compared to other years. This inter-annual fluctuation underscores the impact of external factors that may vary yearly, such as changes in agricultural practices, weather conditions, or economic factors influencing fertiliser importation.

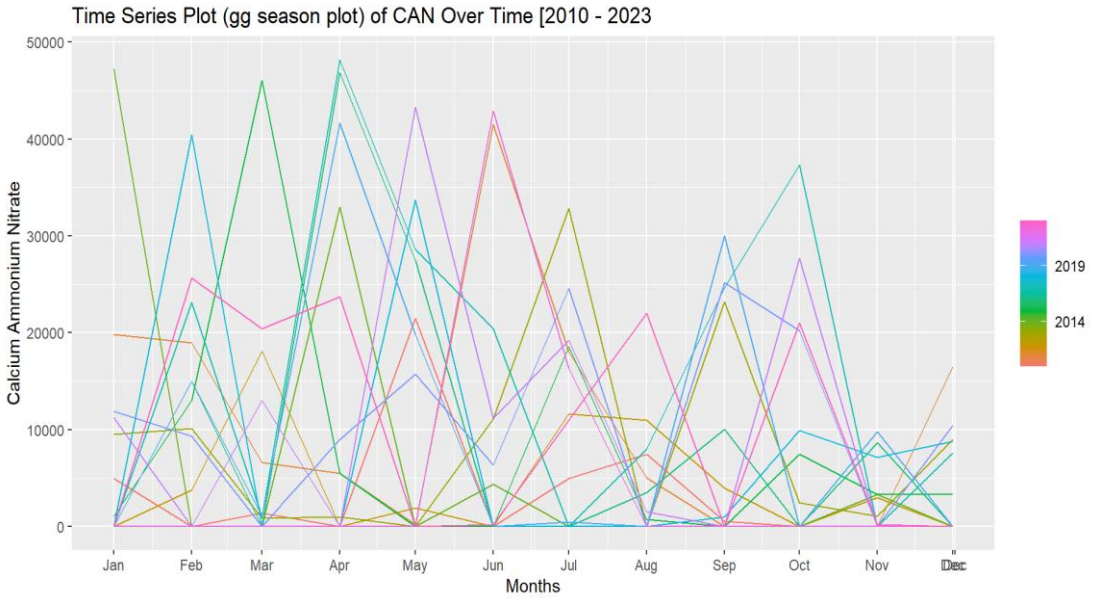


Figure 3: gg seasonal plot for calcium ammonium nitrate

The results in Figure 3 indicate that CAN fertiliser demand differed each year. The total demand for CAN changed over time. In addition, the GG season plot indicates that the demand for CAN will increase significantly in 2022. From 2010 to 2015, the demand was lower than in the rest of the years. The demand for fertilisers in 2022 and 2023 was the highest. This could be explained by the introduction of subsidies in the Kenyan market (GOV, 2022).

Figure 4, labelled "Time Series Plot (GG-Sub series plot) of Total Fertiliser Over Time [2010 - 2023]," provides a more detailed view of the data by breaking down the total CAN demand by month over the years. Each subplot corresponds to a specific month, showing the trend in import quantities from 2010 to 2023. The y-axis indicates the quantity of CAN imports, while the x-axis within each subplot represents the years. The blue line within each month's subplot marks the average import quantity for that month across all years, serving as a reference point. This visualization confirms the seasonal pattern observed in the first plot, with certain months, particularly March and April, consistently exhibiting higher import levels. Conversely, months like July and August tend to show lower or even zero imports, highlighting the variability in demand within the year.

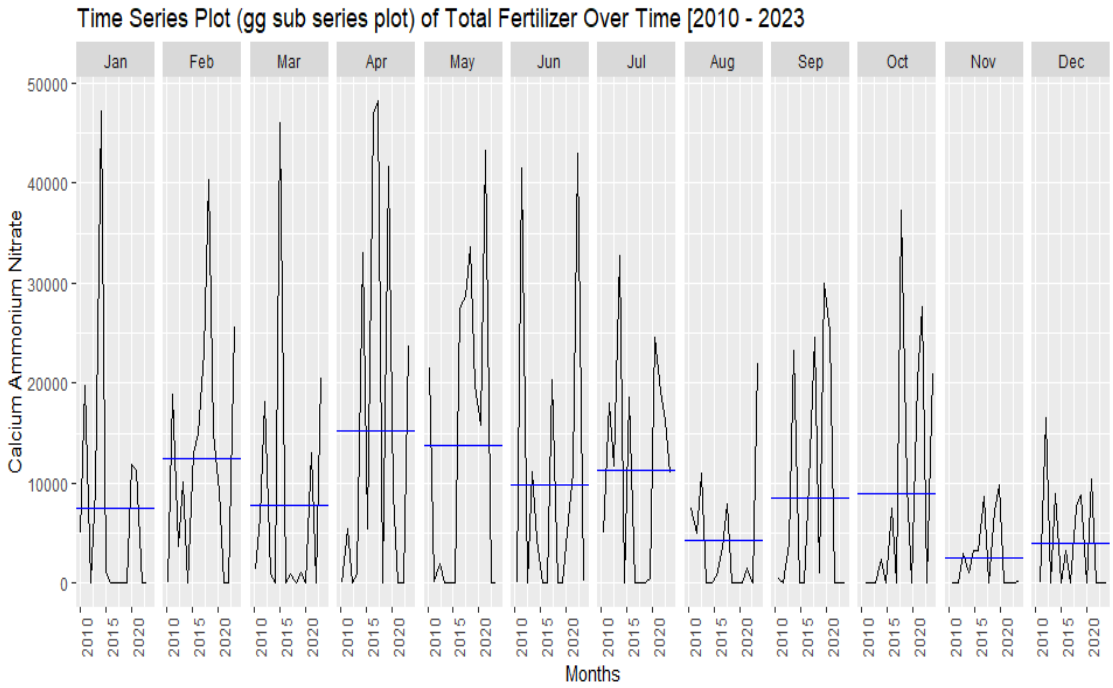


Figure 4: gg subseries plot for calcium ammonium nitrate over 2010 - 2023

Figure 4 shows a comprehensive view of the temporal dynamics of CAN imports, highlighting both seasonal trends and inter-annual variations. The observed patterns are crucial for understanding the factors influencing fertiliser importation and can guide decision-making in areas such as agricultural planning and resource management. The clear depiction of seasonal peaks and troughs emphasizes the need to consider these cycles when analysing the fertiliser market and its impacts on agricultural productivity. The findings of this plot are that February, April, and May have the highest demand for CAN fertiliser. The high demand for Calcium Ammonium Nitrate (CAN) fertiliser in Kenya during February, April, and May, as indicated in Figure 4, is due to the agricultural cycle tied to the long rains season from March to May, which is the main planting period for key crops like maize and wheat (Rural Resilience Programme, 2018). Farmers typically prepare for planting in February and apply CAN as a top-dressing fertiliser in April and May to ensure optimal crop growth. The fertiliser's nitrogen content is crucial for replenishing depleted soils and supporting the early stages of crop development, making these months peak demand periods. Additionally, market dynamics, such as government subsidies and the timing of fertiliser distribution, further amplify the demand during these months.

4.2.1.3 Seasonal Decomposition of Calcium Ammonium Nitrate Time Series

Seasonal decomposition of time series separates the data into three main components: trend, seasonality, and remainder. Figure 5 illustrates the STL (Seasonal and Trend decomposition using Loess) decomposition of "Calcium Ammonium Nitrate" data from 2010 to 2023. The top panel shows the original time series, highlighting variations over time. The second panel reveals the trend, indicating a general increase until around 2015, followed by a slight decrease and stabilization. The third panel displays the seasonal component, capturing consistent yearly patterns. The bottom panel represents the remainder, showing random noise and irregular variations after removing trend and seasonality. This decomposition is vital because it presents the underlying patterns of fertiliser demand over time.

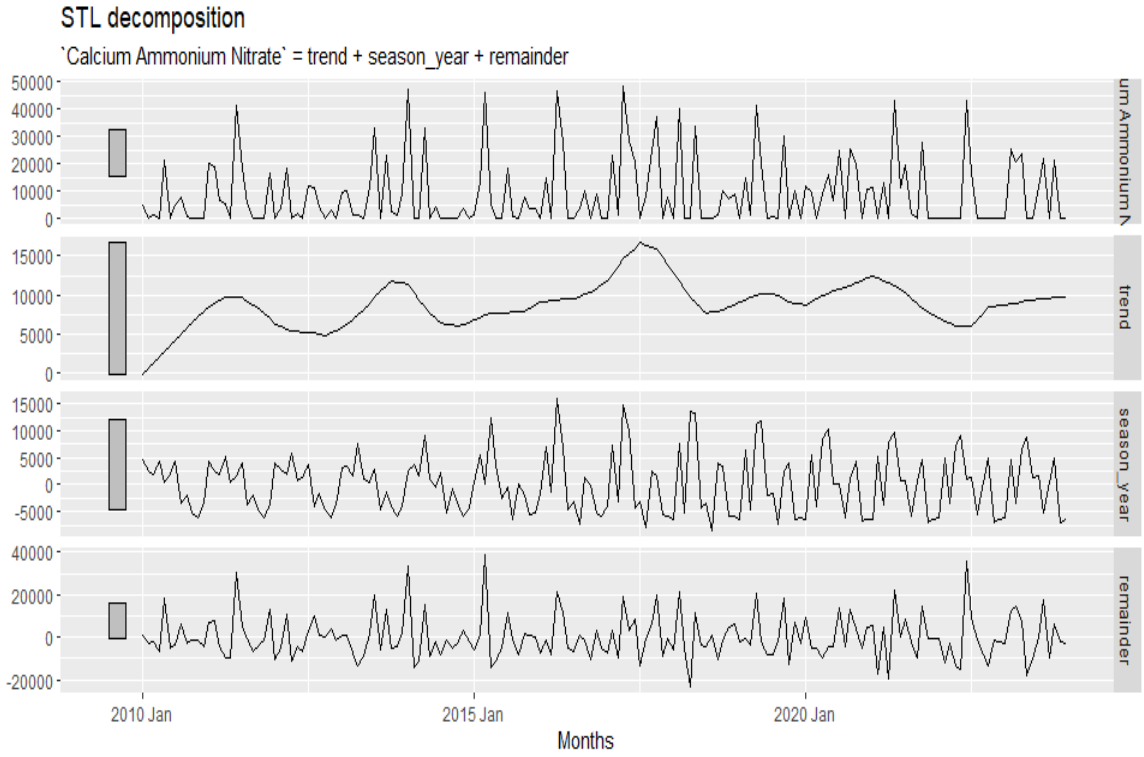


Figure 5: Seasonal-Trend Decomposition of calcium ammonium nitrate

The STL (Seasonal-Trend Decomposition using Loess) analysis revealed that the demand for Calcium Ammonium Nitrate (CAN) fertiliser in Kenya shows distinct seasonal patterns, especially during the long rains season, which aligns with crucial agricultural cycles. The analysis also identified a rising trend in demand over time, indicating a growing dependence on CAN fertiliser, likely driven by intensified

agriculture, increased crop production, and a stronger focus on improving yields through better soil nutrition. This combination of seasonal fluctuations and an upward trend highlights the critical role of CAN in Kenya's agricultural industry.

In order to have a better understanding of the seasonality of the data, a gg-seasonal plot was done to visualize the seasonal patterns of the data correctly. The gg-plot in Figure 6 illustrates seasonal variations in Calcium Ammonium Nitrate (CAN) fertiliser usage across different years, with each line (See appendix 4) representing a specific year from 2014 to 2019 (See appendix 4). The x-axis shows the months from January to December, and the y-axis represents the seasonal component values. The graph reveals consistent seasonal patterns, with peaks in usage from April to May and September to October, such as the high values seen in 2019 around May.

There are noticeable dips from June to August and November to January, indicating lower fertiliser application, as seen in 2014 and 2015. The colour gradient highlights yearly variations, where lighter colours represent earlier years and darker colours represent recent years, showing differences in the magnitude of seasonal peaks, like the higher peaks in 2019 compared to 2014 (Appendix 4). This visualization has highlighted the recurring seasonal trends and the yearly fluctuations in CAN fertiliser usage.

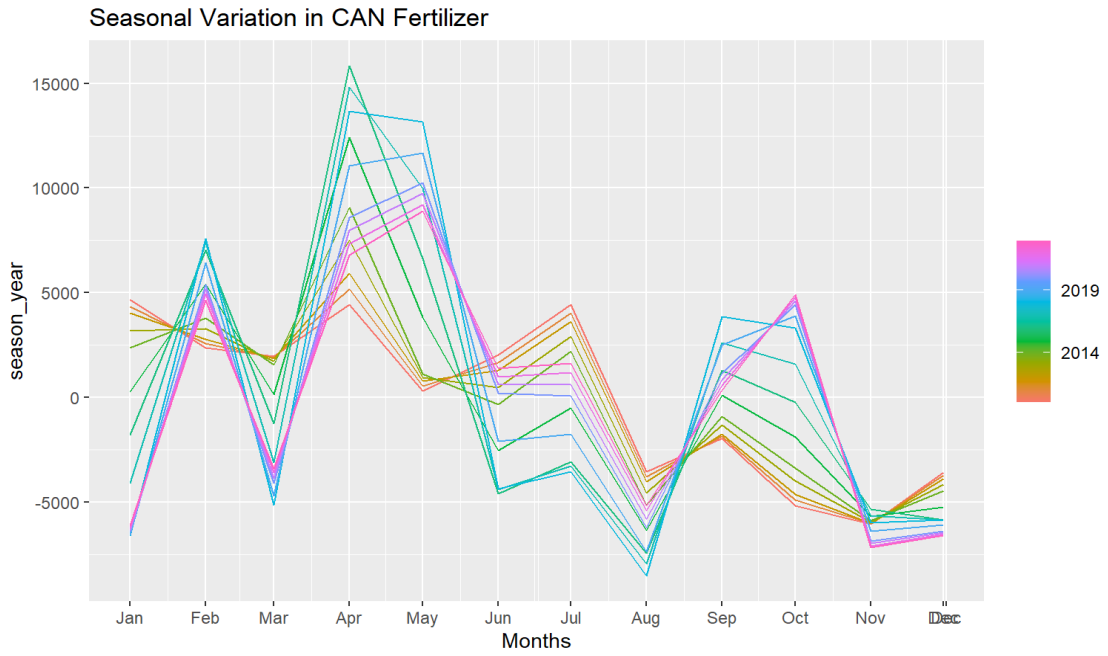


Figure 6: Seasonal variation in calcium ammonium nitrate

This seasonal variation has also been confirmed by the gg-subplots in Figure 7. Figure 7 shows that the demand for CAN is high in all years in February, April, and May. Furthermore, CAN demand is lowest during August, November, and December.

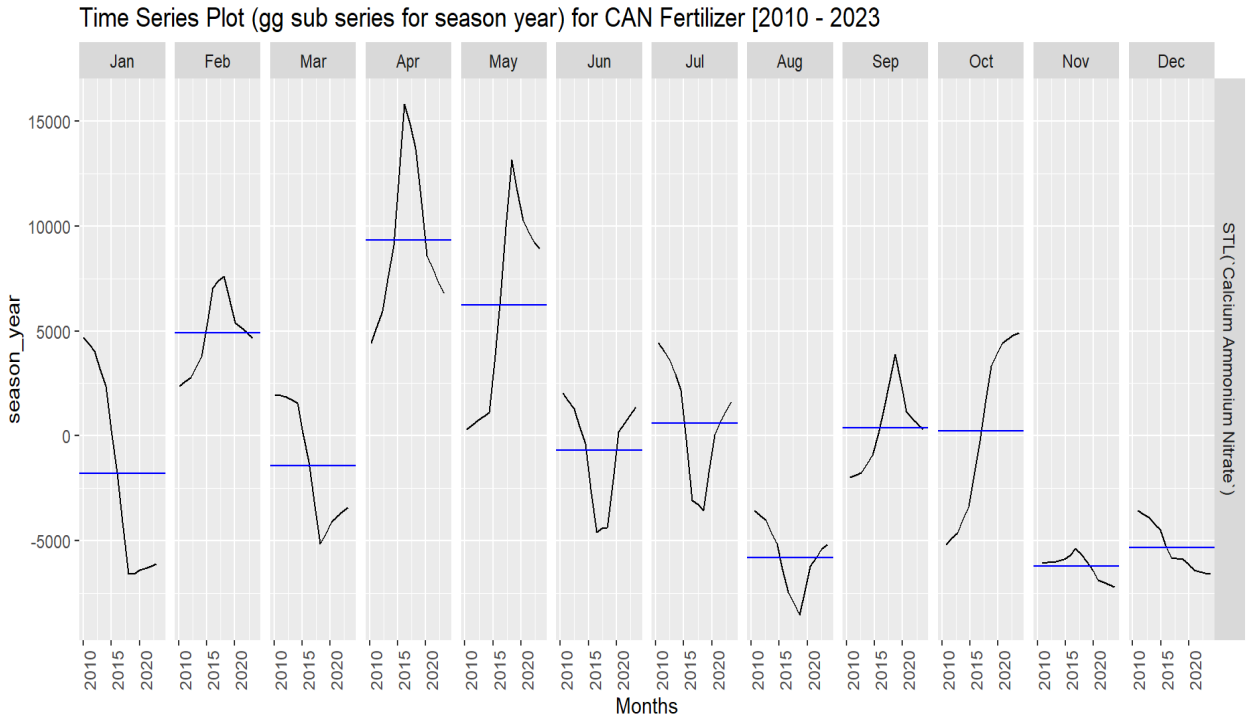


Figure 7: gg subseries plot for season year for calcium ammonium nitrate demand

This variability in demand also reflects the impact of government subsidies that have historically promoted the use of CAN during certain years. The trends observed can be linked to fluctuating market conditions and external factors such as global price changes and supply chain disruptions. The spikes in CAN demand during 2014-2015 and 2017-2018 may have been influenced by government policies or international market conditions that made these fertilisers more accessible or necessary. For example, the global supply chain disruptions in recent years, exacerbated by the COVID-19 pandemic, have directly impacted fertiliser availability and prices in Kenya, which in turn affects demand (Nchanji & Lutomia, 2021).

4.2.1.4 Testing for Stationarity Calcium Ammonium Nitrate Time Series

The data visualisation in Figures 3-7 indicates that the CAN demand keeps changing and has notable seasonal variations caused by various factors such as prices, government policies, and production patterns. This makes the SARIMA model more suitable because it will capture data variability at different times. Due to the seasonality of the data, monthly seasonal factors are considered in the modelling process. The KPSS (Kwiatkowski-Phillips-Schmidt-Shin) test was conducted to assess the stationarity of the time series data. The test yielded a KPSS statistic of 0.1819377 and a p-value of 0.1.

Given that the p-value is more significant than the conventional significance level of 0.05, we fail to reject the null hypothesis of stationarity. This result indicates that there is insufficient evidence to suggest that the time series data is non-stationary, and thus, we conclude that the data can be considered stationary. Figures 8 and 9 show the ACF and PACF for the data CAN demand.

Figure 8 is an ACF for CAN fertiliser, which also shows significant autocorrelations 1,12 and 13. The ACF and PACF plots for Calcium Ammonium Nitrate (CAN) fertiliser data show significant spikes at lags 1, 12, and 13. These spikes indicate strong correlations at these intervals. Significant spikes in the ACF plot suggest that past values at these lags are highly correlated with the current value, highlighting the presence of yearly seasonality (lag 12) and shorter-term dependencies (lags 1 and 13). Figure 9 is a PACF plot for Calcium Ammonium Nitrate (CAN) fertiliser, which shows

significant partial autocorrelations at lags 1, 12, and 13, indicating direct correlations at these intervals. The spike at lag 12 suggests a strong annual seasonal pattern in the data. Significant peaks outside the confidence intervals, marked by dashed blue lines, confirm these lags as statistically meaningful. This pattern highlights the influence of yearly seasonality and some shorter-term dependencies in CAN fertiliser usage.

The significant spikes in the PACF plot at the same points confirm that these lags directly influence the current value, reinforcing the seasonal and short-term patterns in the data.

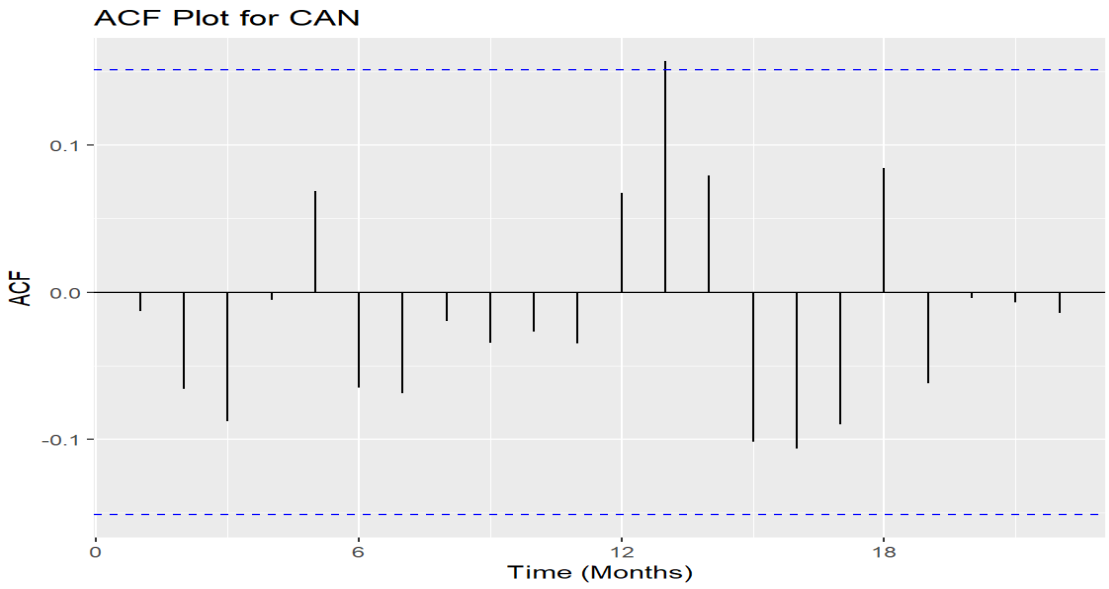


Figure 8: Autocorrelation Function Plots for calcium ammonium nitrate time series data

The Autocorrelation Function (ACF) plot in Figure 8 for the CAN fertiliser data reveals the correlation of the time series with its lagged values over different periods. Notably, there is a significant positive autocorrelation at lag 12, indicating a strong seasonal pattern, with the demand for CAN fertiliser being highly correlated with its value from exactly 12 months prior. This suggests a yearly cyclical trend in the data, consistent with the seasonality identified in the STL analysis. Other lags exhibit weaker correlations, with some negative values, but none are as prominent as the seasonal peak at lag 12.

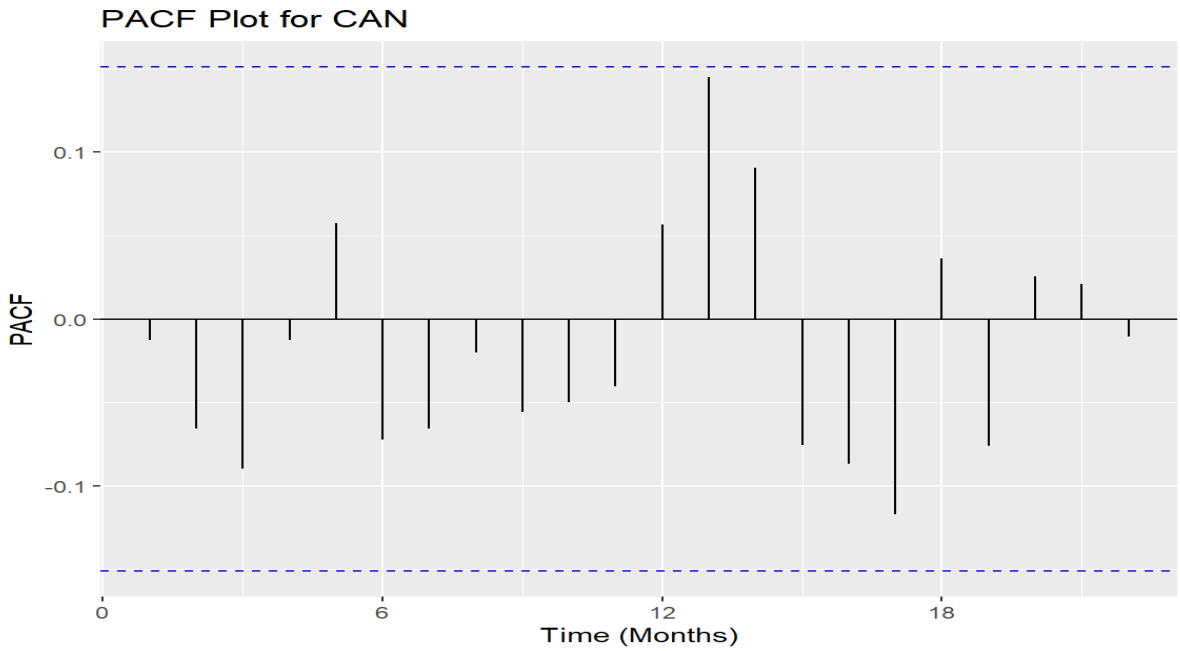


Figure 9: Partial Autocorrelation Function plots for calcium ammonium nitrate time series

The Partial Autocorrelation Function (PACF) plot for the CAN fertiliser data also shows significant partial autocorrelation at lag 12, further confirming the seasonal nature of the demand. The partial autocorrelation at lag 12 suggests that the seasonal effect is still strongly present after accounting for the influence of intermediate lags. Additionally, the PACF plot exhibits a few minor peaks at shorter lags, indicating some short-term dependencies that may need to be considered in modelling the time series.

The ACF and PACF plots confirm that the demand for CAN fertiliser in Kenya follows a strong seasonal pattern with a significant yearly cycle. The pronounced peaks at lag 12 in both plots highlight the importance of incorporating this seasonal effect into any forecasting model. The presence of smaller correlations at other lags suggests that while seasonality is a dominant feature, there may also be other factors influencing demand over shorter periods. These findings reinforce the need for a model that captures both the seasonal and short-term dynamics of CAN fertiliser demand to provide accurate predictions and inform agricultural planning in Kenya.

This variability in demand for CAN reflects the impact of government subsidies that have historically promoted the use of CAN during certain years. The trends observed

in fertilisers like CAN, particularly the spikes and subsequent stabilization, can be linked to fluctuating market conditions and external factors such as global price changes and supply chain disruptions. The spikes in CAN demand during 2014-2015 and 2017-2018 may have been influenced by government policies or international market conditions that made all types of fertilisers more accessible. For example, the global supply chain disruptions in recent years, exacerbated by the COVID-19 pandemic, have directly impacted fertiliser availability and prices in Kenya, which in turn affects demand (Nchanji & Lutomia, 2021).

4.2.2 Visualisation of Calcium Nitrate Time Series

Figure 10 is the time series plot of Calcium Nitrate (CN) fertiliser from January 2010 to 2023. The time series plot reveals a considerable fluctuation in demand over the period. The data shows a cyclical pattern with recurring peaks and troughs, indicating varying levels of fertiliser demand. There are notable spikes in mid-2010 and early 2014, suggesting periods of increased demand followed by stability or lower demand. From 2015 to 2019, the plot displays moderate variability, with less pronounced peaks, indicating more consistent or controlled demand. However, starting in 2020, the demand variability becomes more pronounced and erratic, culminating in a significant spike in early 2023, representing the highest demand level in the series. This suggests that demand for Calcium Nitrate fertiliser is highly variable and influenced by factors such as seasonal requirements, market conditions, or external events.

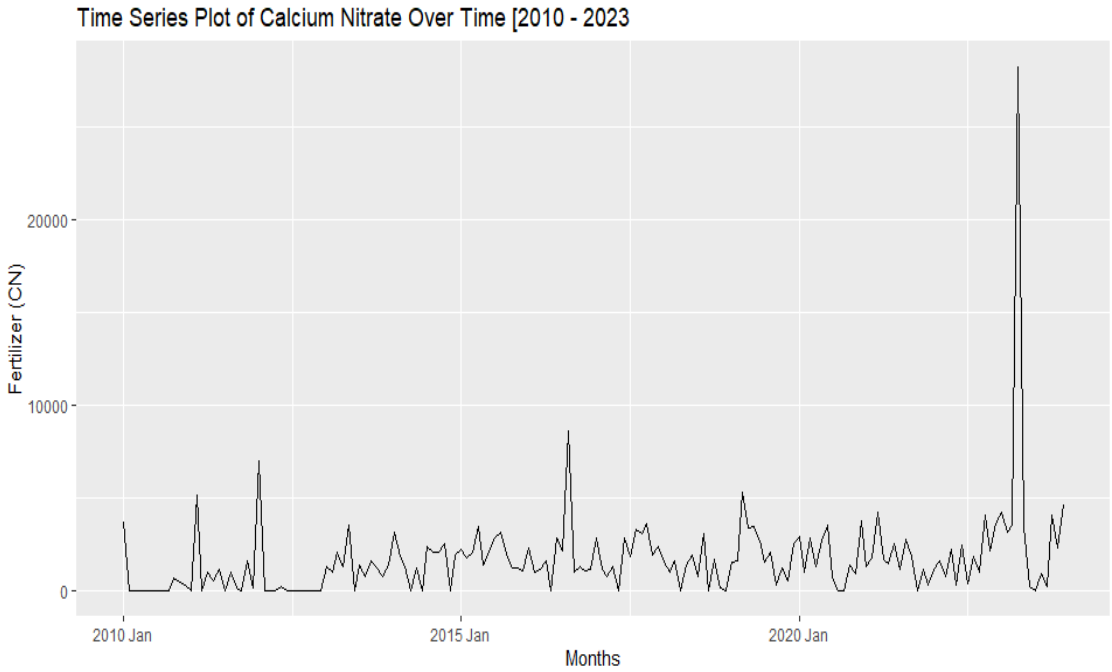


Figure 10: Time series plot of calcium nitrate over time

4.2.2.1 Normality Test of Calcium Nitrate Time Series

The Jarque-Bera test was performed to determine whether Calcium Nitrate fertiliser data distribution is normal. The test produced an X-squared statistic of 38,584 with 2 degrees of freedom and a p-value of <0.0001. The p-value is much lower than the significance level (0.05), so we reject the null hypothesis that the data follows a normal distribution. This result shows that the distribution of the Calcium Nitrate data differs significantly from a normal distribution, indicating possible skewness and/or kurtosis.

Table 3: The Jarque Berra test statistics for calcium nitrate time series

Statistic	Value
X-squared	38584
Degrees of Freedom	2
p-value	< 0.0001

4.2.2.2 Seasonality Plot Checks on Calcium Nitrate Time series

The GG-season plot in Figure 11 illustrates the seasonal trends and fluctuations in fertiliser demand. Each line represents a specific year demand (See appendix 4). The plot shows clear seasonal patterns, with a notable spike in April 2014, indicating a significant rise in demand that month. This peak suggests that agricultural cycles, such as planting seasons or specific crop needs, are likely affected by fertiliser use. The

differences in peak intensity across years also indicate inconsistent demand patterns, possibly due to weather variations, market changes, or shifts in farming practices.

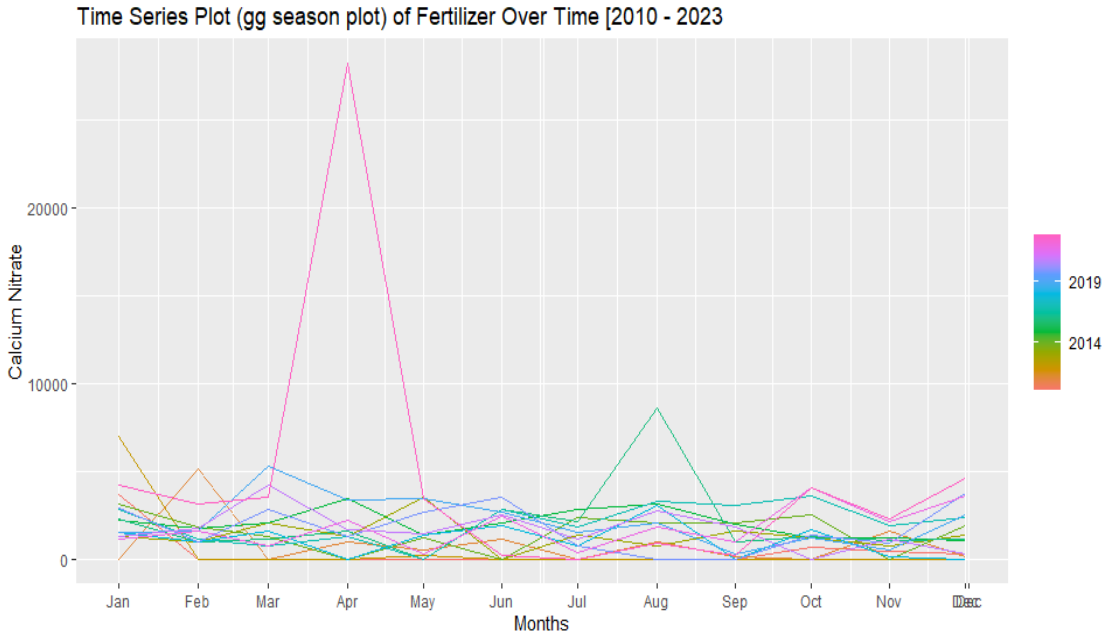


Figure 11: Time series plot (gg season plot) of calcium nitrate fertiliser over time

In April 2014 (Figure 11), the demand for Calcium Nitrate fertiliser in Kenya experienced a significant spike, deviating markedly from the generally stable demand observed over the years. While the demand for this fertiliser typically remained within an average range, showing only slight annual increases, the surge in April 2014 stands out. This unusual increase could be attributed to a combination of factors such as external factors, such as favourable weather conditions during that period, which would have encouraged higher agricultural activity, and the strategic timing of fertiliser distribution just before the planting season, which might have contributed to this significant demand boost (Rural Resilient Program, 2018).

The GG-Subseries plot in Figure 12 presents Calcium Nitrate fertiliser data by showing monthly trends from 2010 to 2023. Each subplot corresponds to a specific month, illustrating the changes in fertiliser quantities within that month across the years. This plot highlights both intra-annual variations and long-term trends. The subplot for April prominently displays a significant spike in 2014, which matches the observation from the GG-Season plot in Figure 11. Additionally, the plot indicates a gradual rise in baseline demand levels, which has been especially noticeable in October and November

in recent years. This trend suggests a possible increase in fertiliser use during these months, possibly due to shifts in crop cycles or expanded farming activities.

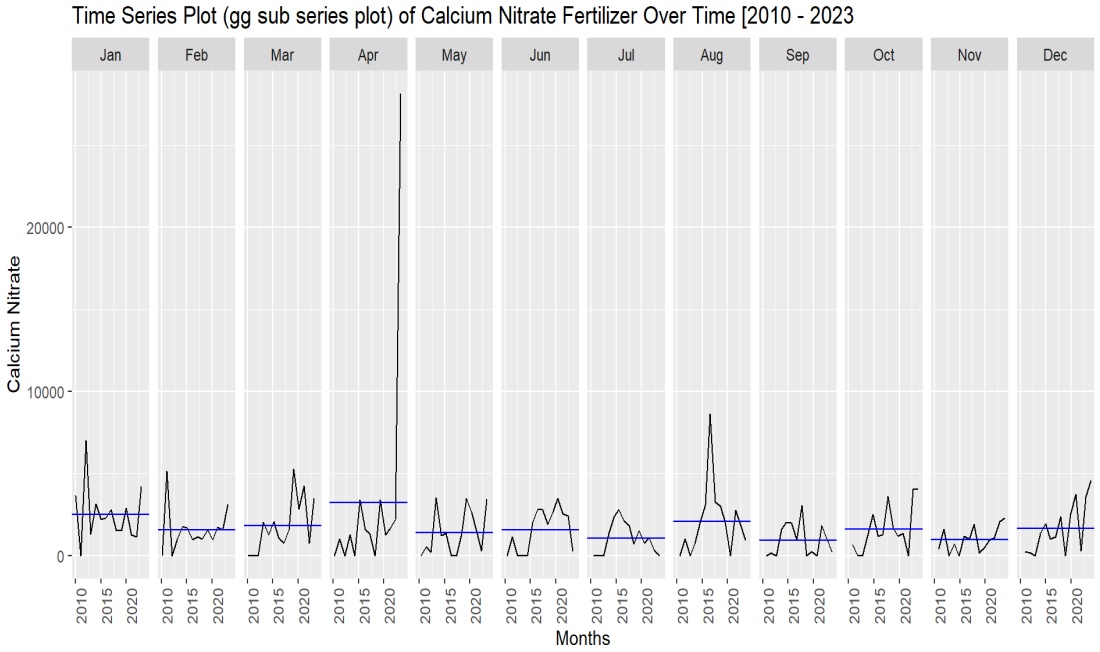


Figure 12: gg sub series plot of calcium nitrate fertiliser

4.2.2.3 Seasonal Decomposition of Calcium Nitrate Time Series

The STL decomposition graph of "Calcium Nitrate" in Figure 13 illustrates the time series data broken down into trend, seasonal variation, and residual components. The original data (top panel) shows the overall changes in Calcium Nitrate values from 2010 to 2023. The trend component (second panel) reveals a steady increase starting in 2018, with a sharp rise around 2022, indicating a long-term upward trend. The seasonal component (third panel) displays recurring yearly patterns, suggesting periodic fluctuations likely due to seasonal factors. The remainder (bottom panel) represents the irregular, random variations not accounted for by the trend or seasonal components, showing occasional spikes and dips. The fertiliser levels have significantly increased since around 2018, with a pronounced spike in 2022. The seasonal component displays consistent yearly patterns, suggesting periodic fluctuations likely due to seasonal influences. The remainder component captures irregular, random variations with occasional sharp spikes, indicating factors affecting the data that are not accounted for by the trend or seasonal components.

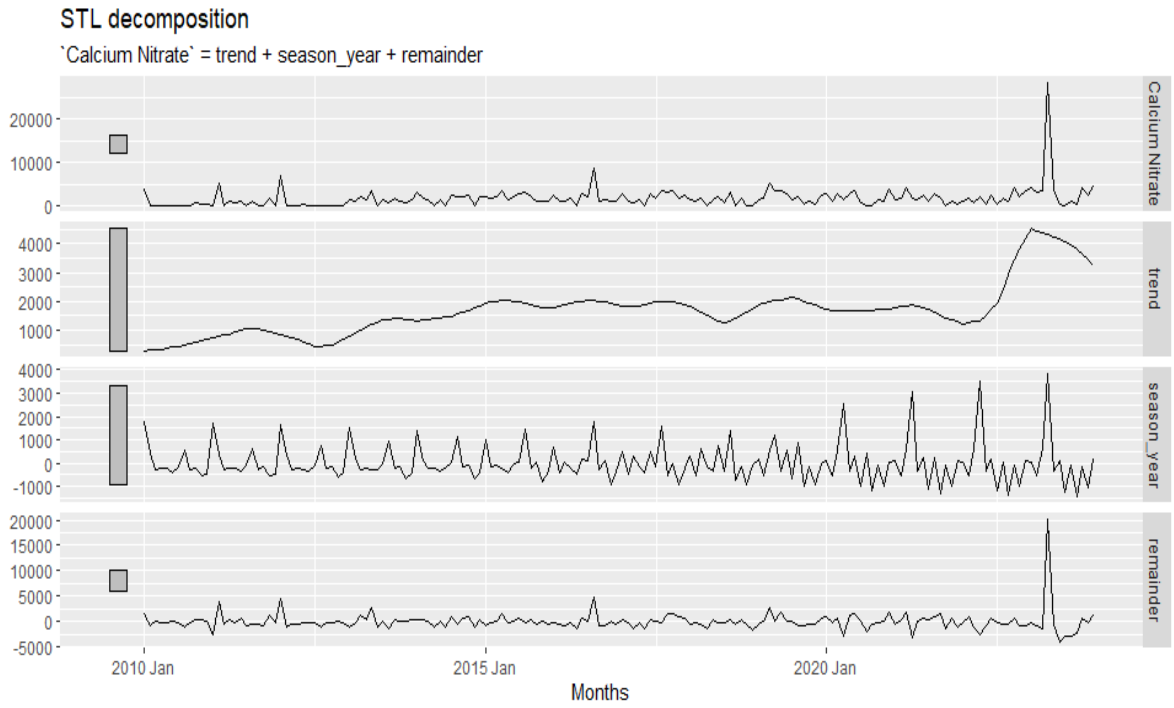


Figure 13: Seasonal-Trend Decomposition plot for calcium nitrate

The time series data for Calcium Ammonium Nitrate (CAN) imports shows a steady increase in demand each year, indicating a rising need for the fertiliser. The STL decomposition also indicates an increasing trend, which implies more use of CAN, likely due to growing agricultural activities or changes in farming methods. Additionally, the data reveals seasonal patterns, with higher demand in certain months reflecting the impact of planting and harvest cycles on fertiliser needs.

In order to better understand the seasonality of the data, a plot of seasonal variation in calcium nitrate fertiliser was done. Figure 14 shows the seasonal patterns in Calcium Nitrate demand levels across years. Each line (See appendix 4) represents a different year from 2010 to 2023, with noticeable peaks in April and August. These peaks indicate a significant increase in Calcium Nitrate usage during these months, suggesting a seasonal dependency likely influenced by agricultural cycles or climatic conditions.

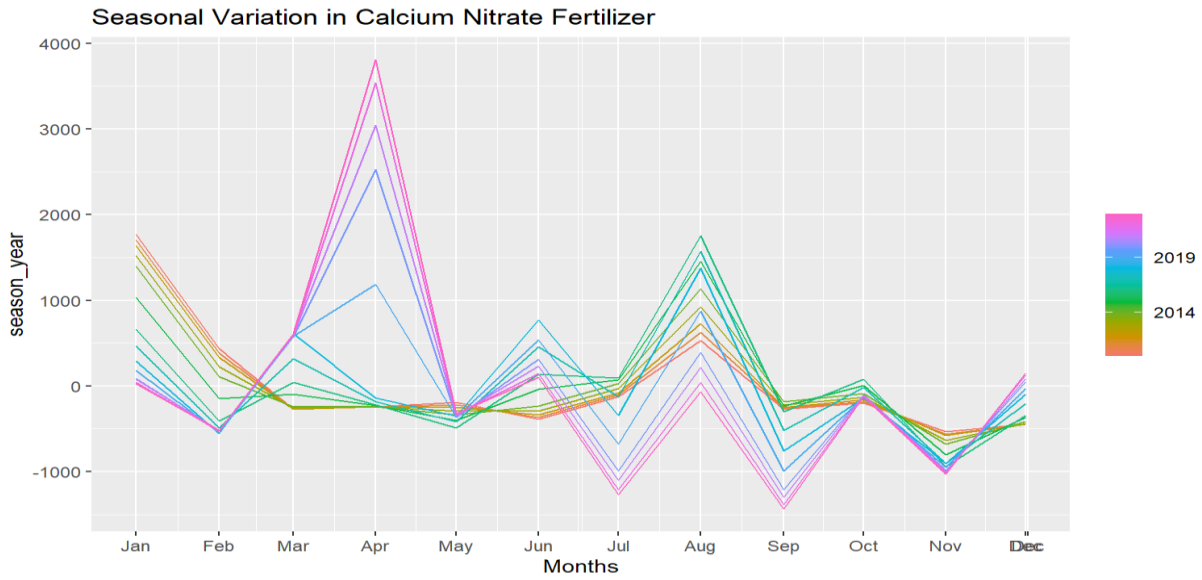


Figure 14: Seasonal variation in calcium nitrate fertiliser

The high demand for Calcium Ammonium Nitrate (CAN) fertiliser during March, April, and May is primarily due to the long rains and increased farming activities in these months. The onset of the rainy season supports planting and crop growth, leading farmers to use more fertiliser. Additionally, there is a slight increase in demand for CAN in July, August, and September, likely due to mid-season fertilisation needs as crops continue to grow and develop during the rainy season.

Figure 15 shows a detailed month-by-month breakdown of Calcium Nitrate demand levels from 2010 to 2023. Each panel represents a specific month, displaying changes in Calcium Nitrate levels across the years for that month. The blue horizontal lines show the average level for each month, highlighting long-term trends and variations. This graph confirms the significant increase in April and August observed in the first graph and shows a general upward trend in recent years.

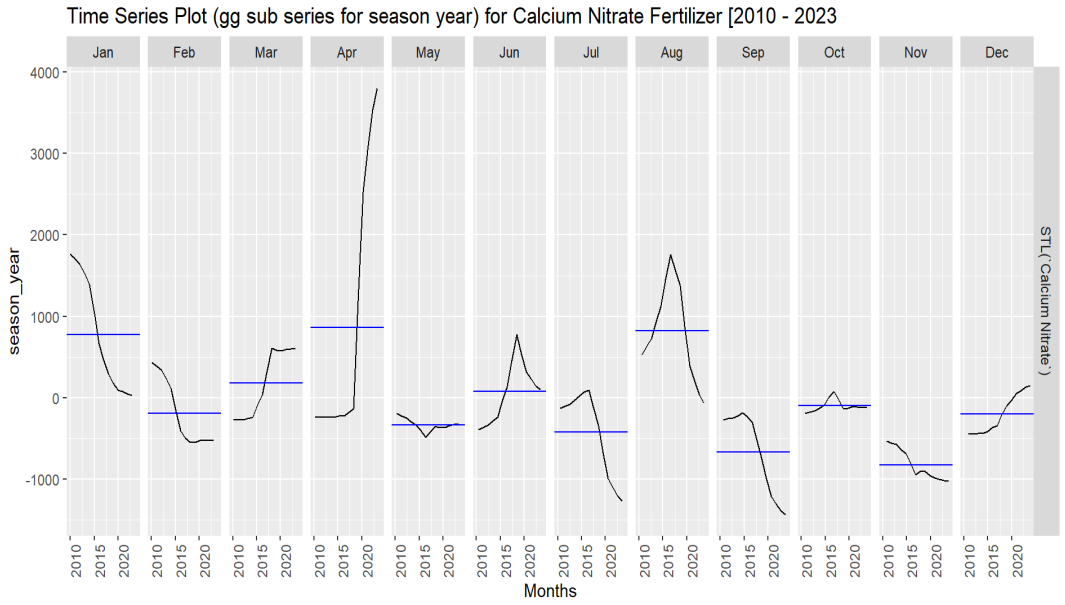


Figure 15: gg subseries for season year for calcium nitrate

The month-specific trends in Figure 15 support the seasonal patterns in the Figure 14 graph. Both graphs indicated that April and August have the highest Calcium Nitrate levels, reflecting consistent seasonal peaks. The upward trend in the second graph matches the increasing peaks over the years in the first graph, indicating that the long-term trend and seasonal variations are linked, contributing to a better understanding of Calcium Nitrate usage or levels over time.

4.2.2.4 Testing for Stationarity Calcium Nitrate Time Series

A KPSS test was conducted to determine if the calcium nitrate data was stationary. The results showed a KPSS statistic of 0.7916768 and a p-value of 0.01, indicating that the time series is not stationary. As a result, the "unitroot_ndiffs" function was used, revealing that one differencing was needed to achieve stationarity. After differencing the data once to address the non-stationarity, Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots were generated, which indicated that the ACF and PACF were within the unit root, confirming the achievement of stationary in the first difference. Figures 17 and 18 show the ACF and PACF generated after differencing, which indicates that the first differencing achieved stationarity of the data.

The first graph in Figure 16 represents the differenced time series data for Calcium Nitrate (CN), intended to stabilize variance and remove seasonality. The first difference

successfully reduced the overall trend and variability in the data, though significant fluctuations and sharp peaks remain. Some repeating patterns suggest that additional seasonal differencing may be necessary, or there might be remaining autoregressive elements within the data.

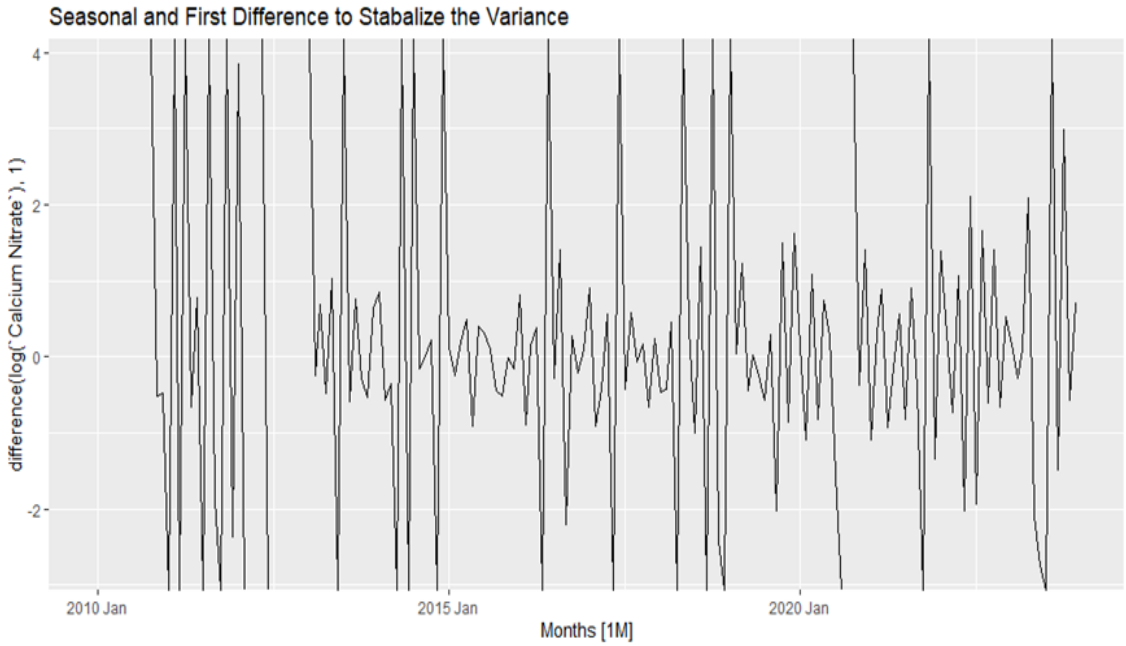


Figure 16: Data plot after the first difference

Figure 17, on the other hand, shows the Autocorrelation Function (ACF) for the differenced CN time series. The ACF reveals significant positive spikes at lags of around 1, 2, and 12 months, suggesting the presence of autoregressive components and seasonality in the data. The tapering off of the autocorrelations indicates that the data may have a mixture of AR and MA processes, but the cut-off after 12 months suggests that a seasonal model component might be necessary.

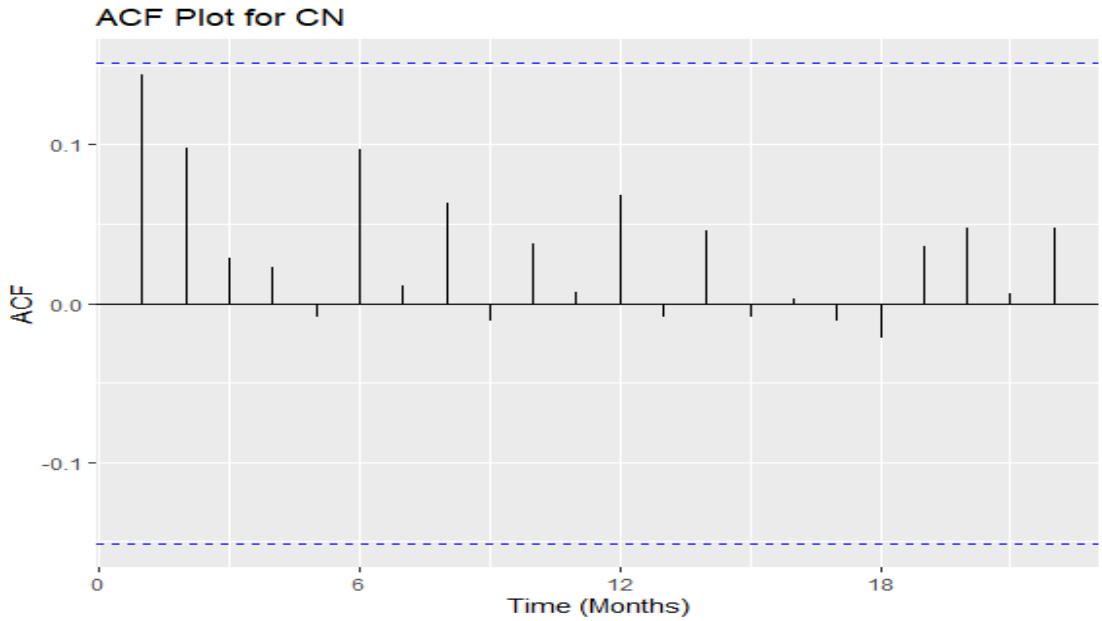


Figure 17: Autocorrelation function plot for calcium nitrate

The graph in Figure 18 displays the Partial Autocorrelation Function (PACF) for the differenced CN time series. Like the ACF, the PACF indicated that the time series had significant spikes at the first few lags (around 1, 2, and 12 months). This pattern suggests that an autoregressive process (AR) is present and that the data may benefit from a low-order AR model, possibly with a seasonal component at lag 12

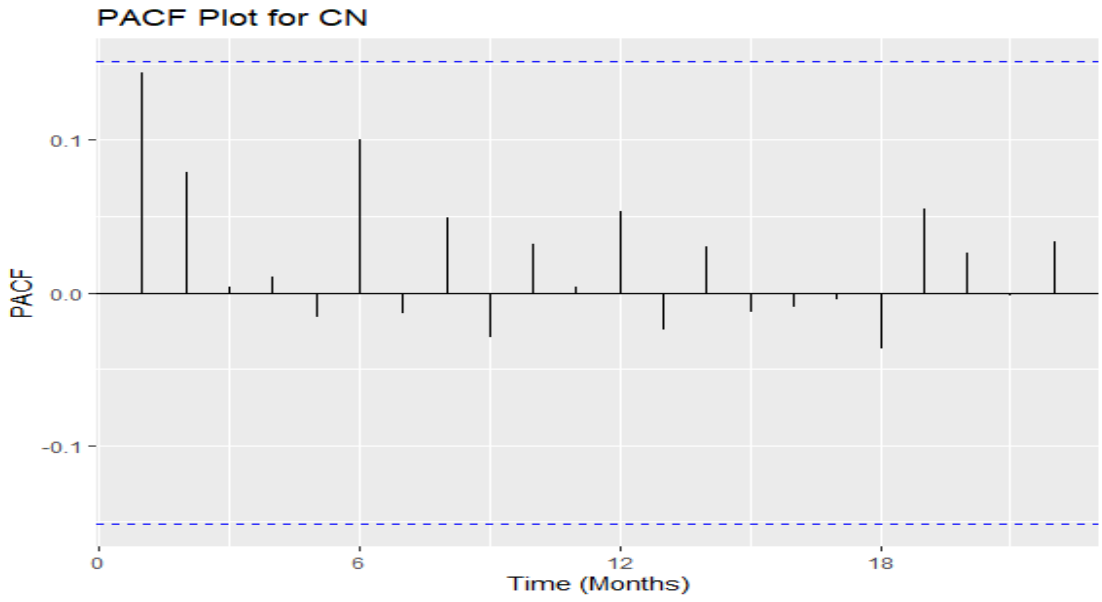


Figure 18: Partial autocorrelation function plot for calcium nitrate

The results from the KPSS test, the subsequent differencing, and the ACF and PACF analyses provide a solid foundation for the ongoing time series analysis, ensuring accurate and reliable modelling and forecasting of Calcium Nitrate levels. The combined analysis of the differenced time series plot, ACF, and PACF indicated that the time series data for Calcium Nitrate exhibits characteristics of both autoregressive and seasonal components. The differencing reduced the non-stationarity, but the remaining spikes in the ACF and PACF indicate that further modelling, perhaps with ARIMA or SARIMA, will be needed to capture the underlying data structure accurately. Therefore, a model with both AR terms and seasonal components is adequate to predict future values and account for the observed autocorrelations.

The demand for calcium nitrate showed significant increases in mid-2010 and early 2014, possibly indicating periods of heightened agricultural activity or the introduction of new crop varieties requiring this fertiliser. Calcium nitrate is vital for crops like tomatoes, widely grown in Kenya for local consumption and export. Stabilizing demand after these peaks may suggest adopting improved soil management practices or consistent crop production techniques. The increased demand during January, April, and August from 2015 to 2019 aligns with the planting seasons for vegetables and fruits, where calcium nitrate is crucial for preventing issues like blossom-end rot (Taylor & Locascio, 2004). Kenya's fertiliser subsidy programs have played a significant role in shaping the demand trends.

4.2.3 Visualisation of Diammonium Phosphate Time Series

Figure 19 is the time series plot of Diammonium Phosphate (DAP) fertiliser from January 2010 to 2023. From the plot, it is clear that the demand for DAP keeps on changing each month. The graph shows a cyclic pattern with recurring peaks and troughs, indicating varying levels of fertiliser demand. A notable drop in demand in 2010 and 2014 suggests periods of decreased demand. From 2015 to 2019, there is significant demand variability, with highly pronounced peaks indicating high demand for DAP in these years. However, in 2021, there was a significant drop in demand for DAP, which may have been attributed to government policies and the COVID-19 effects (Mutegi *et al*., 2024). This suggests that demand for DAP fertiliser is highly

variable and influenced by seasonal requirements, market conditions, or external events.

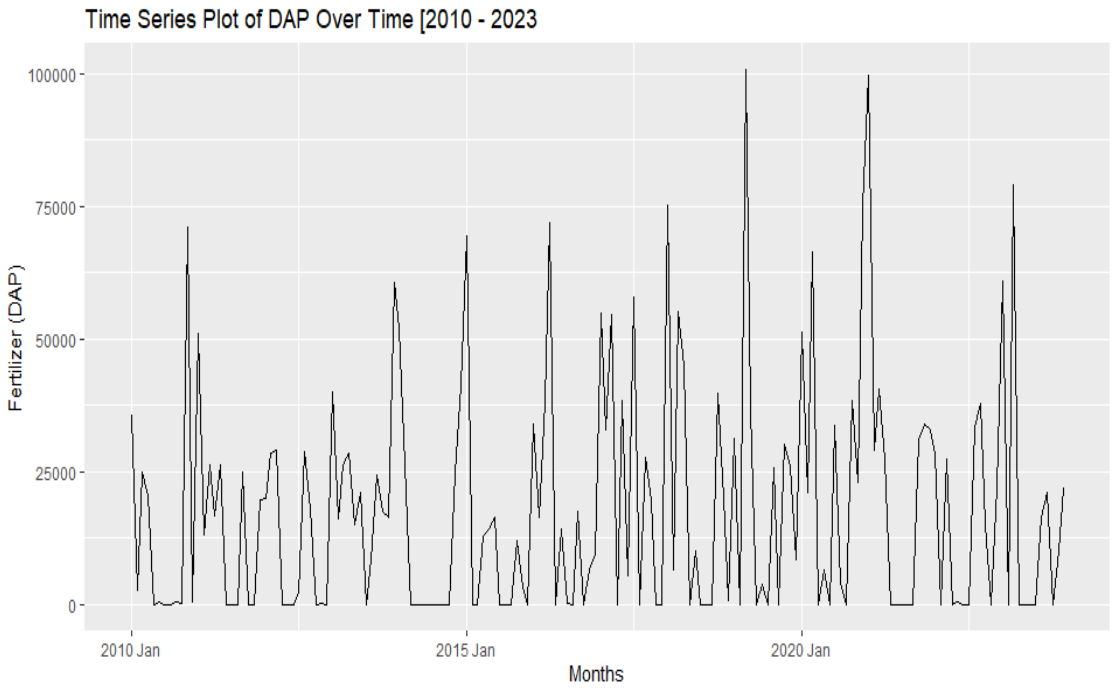


Figure 19: Diammonium phosphate time series plot 2010 – 2023

4.2.3.1 Normality Test of Diammonium Phosphate Time Series

A Jarque-Bera test was conducted to assess the normality of the Diammonium Phosphate data. The test yielded a test statistic of 78.566 with two degrees of freedom and a p-value of <0.0001. Given the extremely small p-value, less than the critical $\alpha = 0.05$, we reject the null hypothesis that the data follows a normal distribution. This result indicates that the distribution of Diammonium Phosphate data deviates significantly from normality.

Table 4: Jarque-Bera test *for* Diammonium phosphate time series

Statistic	Value
Test Statistic (X-squared)	78.566
Degrees of Freedom (d.f)	2
p-value	< 0.0001
Conclusion	Reject H□: Data does not follow a normal distribution

In order to better understand the seasonality of the data, a plot of seasonal variation in calcium nitrate fertiliser was done.

Figure 20 is a GG-Season plot that overlays the DAP demand for each year using differently coloured lines (See appendix 4). This plot highlights clear monthly trends, with noticeable peaks in January, March, and June and smaller peaks in October and December. The variability between years is evident, as each line represents a different annual pattern, suggesting significant shifts in demand. This variability implies that external factors such as weather, market conditions, or policy changes significantly impact DAP demand.

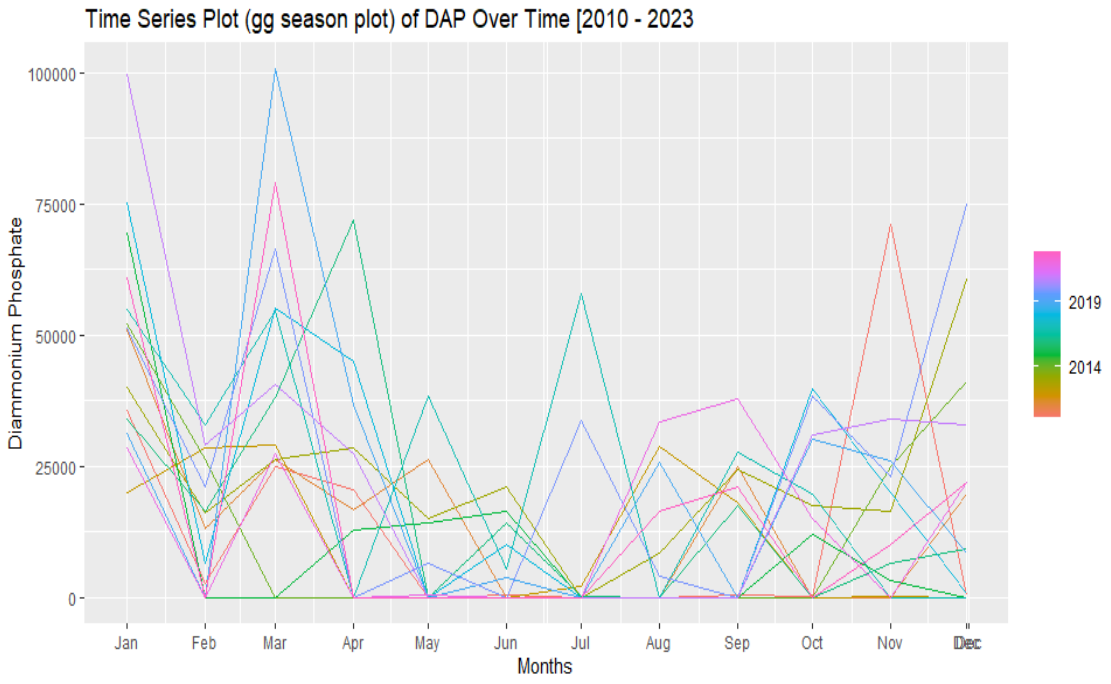


Figure 20: gg Season plot of diammonium phosphate demand time series

The findings (Figure 20) indicate that demand for Diammonium Phosphate (DAP) fertiliser peaks in February, March, and April, as well as in October, November, and December. The high demand during the first set of months can be attributed to the preparation for and early stages of planting, as farmers use DAP to enrich the soil before the main growing season. In contrast, the increased demand in the latter months aligns with the need for fertiliser during the critical stages of crop development and preparation for the next planting season. This pattern reflects the seasonal agricultural cycles where DAP is crucial for initial soil preparation and ongoing crop nourishment.

Figure 21 presents seasonal Diammonium Phosphate (DAP) fertiliser demand trends from 2010 to 2023. The observations reveal high demand in January and lower but

fluctuating demand from February to April. There is another peak from October to December, particularly in December. These patterns suggest two distinct peak periods: one in the early year (January to March) and another towards the end (October to December), with notable year-to-year variability hinting at the influence of external factors.

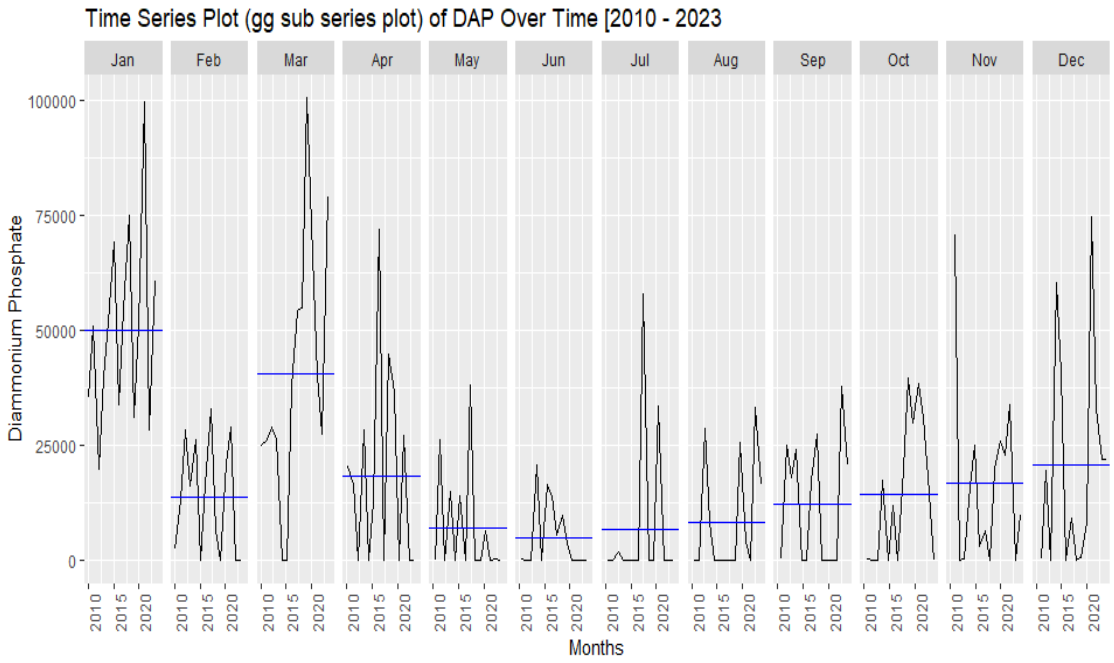


Figure 21: gg subseries plot of diammonium phosphate fertiliser demand

Combining insights from both graphs, it is evident that DAP fertiliser demand is strongly seasonal with considerable variability. The demand peaks generally coincide with planting seasons and preparation for winter crops. Despite the overall seasonal pattern, specific annual conditions can lead to notable deviations, indicating the significant role of external factors. Therefore, the demand for DAP fertiliser is high in Jan, Feb, March, and April; during the preparation for long rain production, October, November, and December during short rain.

4.2.3.2 Seasonal decomposition of DAP time series

The graph in Figure 22 demonstrates the seasonal variation in Diammonium Phosphate (DAP) usage over 2010-2023, with each line representing a different year and distinguished by colours (Appendix 4). The data shows a clear seasonal pattern, with two noticeable peaks in January and March, indicating higher usage during these

months. Usage drops significantly from April to August, reflecting a trough during the summer months, followed by a gradual increase towards the year's end. The variations among the years are relatively consistent, with some years exhibiting slightly higher or lower usage during peak months.

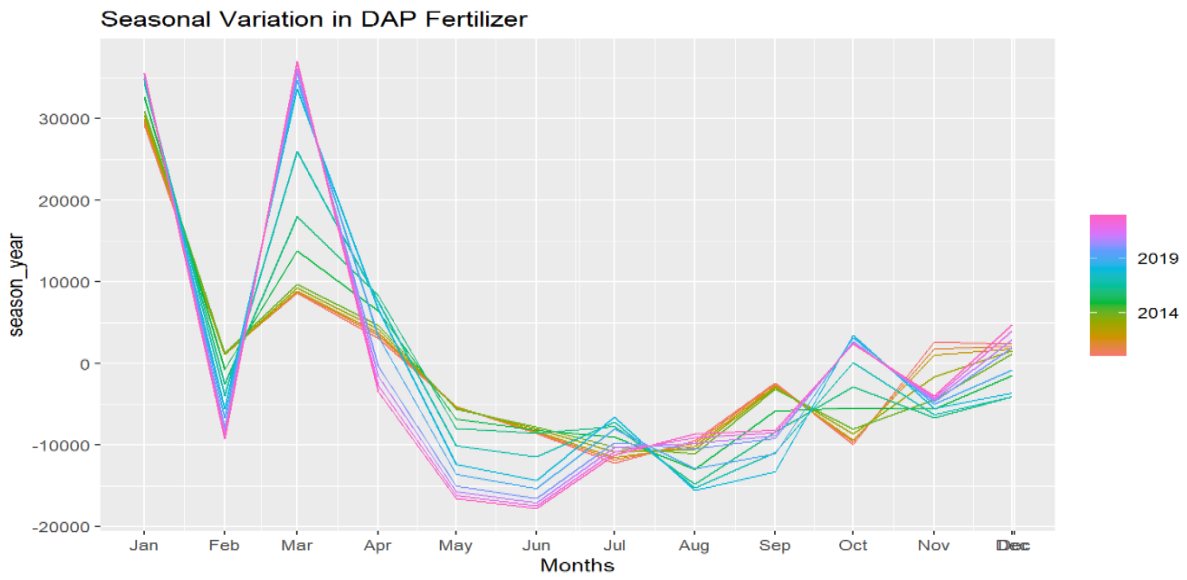


Figure 22: Seasonal variation in diammonium phosphate

The characteristics of the demand patterns are attributed to the pre-planting period when farmers prepare the soil for the growing season. A significant drop in usage is observed from April to August, corresponding to the summer months when fertiliser application typically decreases. The demand for DAP rises towards the end of the year due to preparations for the next planting season and replenishing soil nutrients. While the seasonal pattern remains consistent across the years, individual years have slight variations in peak consumption because of the changes in agricultural practices or climatic conditions that influence fertiliser application.

Figure 23 divides the plot into twelve subplots, each corresponding to a different month. The y-axis represents DAP usage within each subplot, while the x-axis covers 2010 to 2023. The black lines depict the monthly usage trends over the years, and the blue lines show the average usage for each month. This plot emphasizes the monthly seasonality of DAP usage, with notable spikes in January and March, aligning with the findings of

the graph in Figure 22. The trend lines indicate a cyclical pattern with high demand in the early months, a decline in the middle of the year, and a rise towards the year's end.

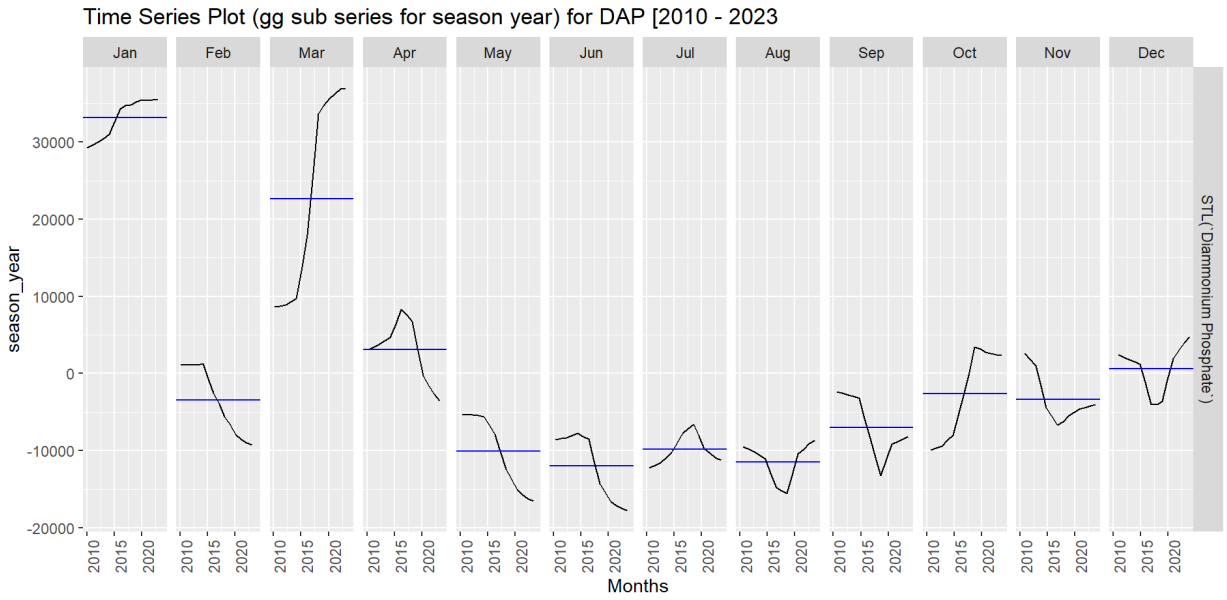


Figure 23: gg subseries plot for season year for diammonium phosphate fertiliser demand

The findings are that the demand for DAP is highest in January and March, which confirms the findings in figure 22. Figure 24, on the other hand, displays a seasonal trend decomposition using LOESS (STL) of the DAP time series. This decomposition divides the data into four components: the observed data, the trend, the seasonal component, and the remainder (residuals). The first panel shows the raw data, which includes evident periodic spikes. The second panel illustrates the trend component, showing a general increase in DAP usage up to around 2017, followed by a slight decline. The third panel highlights the seasonal component, which reveals a consistent pattern of high usage in the early months and lower usage in the middle of the year across the entire timeframe. The final panel represents the remainder component, capturing the irregularities or noise not accounted for by the trend or seasonality.

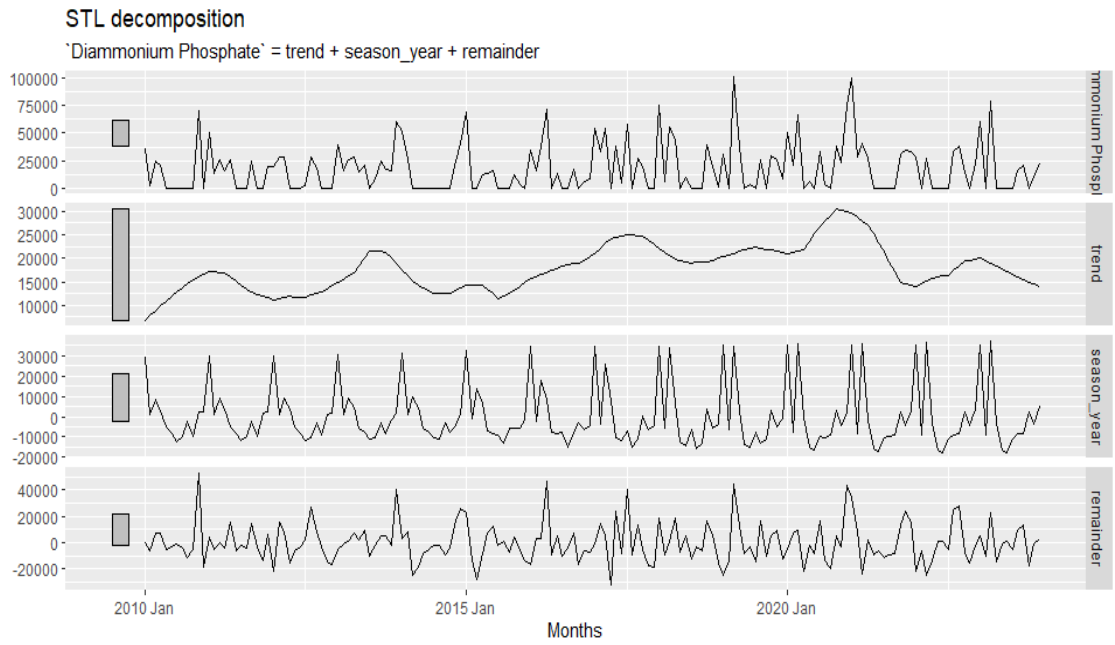


Figure 24: Seasonal-Trend Decomposition plot for diammonium phosphate fertiliser time series

The STL (Seasonal and Trend decomposition using Loess) analysis of the Diammonium Phosphate (DAP) time series from 2010 to 2023 reveals both a trend and seasonality in the data. The analysis shows an upward trend, indicating a gradual rise in DAP demand over the years. This suggests that, even with seasonal fluctuations, the overall demand for fertiliser has been increasing over the years. The seasonal component displays recurring higher consumption patterns during certain months, corresponding with planting cycles. These findings highlight the seasonal variations and long-term increase in DAP demand.

The three graphs reveal that the Diammonium Phosphate (DAP) time series exhibits a strong seasonal pattern, with high usage in January and March and low usage during summer (June to August). The trend component indicates a general increase in DAP usage over the years, reaching a peak around 2017 before a slight decline. Figures 22 and 23 show that the consistent seasonal pattern suggests a predictable cyclic behaviour in fertiliser usage. The STL decomposition further explains this cyclic behaviour by separating it from long-term trends and random fluctuations, providing a clearer understanding of the factors driving DAP usage. The data demonstrates clear

seasonality, which may be influenced by agricultural cycles, potential policy changes, and market dynamics that affect fertiliser application practices.

4.2.3.3 Testing for stationarity Diammonium Phosphate time series

The KPSS (Kwiatkowski-Phillips-Schmidt-Shin) test was conducted to check the stationarity of the time series data. The test returned a KPSS statistic of 0.1985524 and a p-value (KPSS p-value) of 0.1. The null hypothesis of the KPSS test is that the data is stationary, while the alternative hypothesis is that it is non-stationary. Since the p-value exceeds the common significance level (e.g., 0.05), we reject the null hypothesis, indicating that the time series data is likely stationary. The test on the number of differencing needed was done using are, and the test results indicated that no differencing was needed. The ACF and PACF plots below indicate a minimal correlation between the time series and its lags and the partial correlation coefficients between the series and its lags.

The Autocorrelation Function (ACF) plot for DAP demand in Figure 25 shows the correlation of the time series with its lagged values over different time lags. The plot reveals a significant positive correlation at lag one and another significant spike at lag 12. The positive spike at lag 1 indicates that the current demand for DAP is positively correlated with the previous month's demand, indicating some persistence in the demand pattern. The spike at lag 12, on the other hand, indicates a seasonal pattern, suggesting that demand one year ago (12 months prior) strongly influences the current demand.

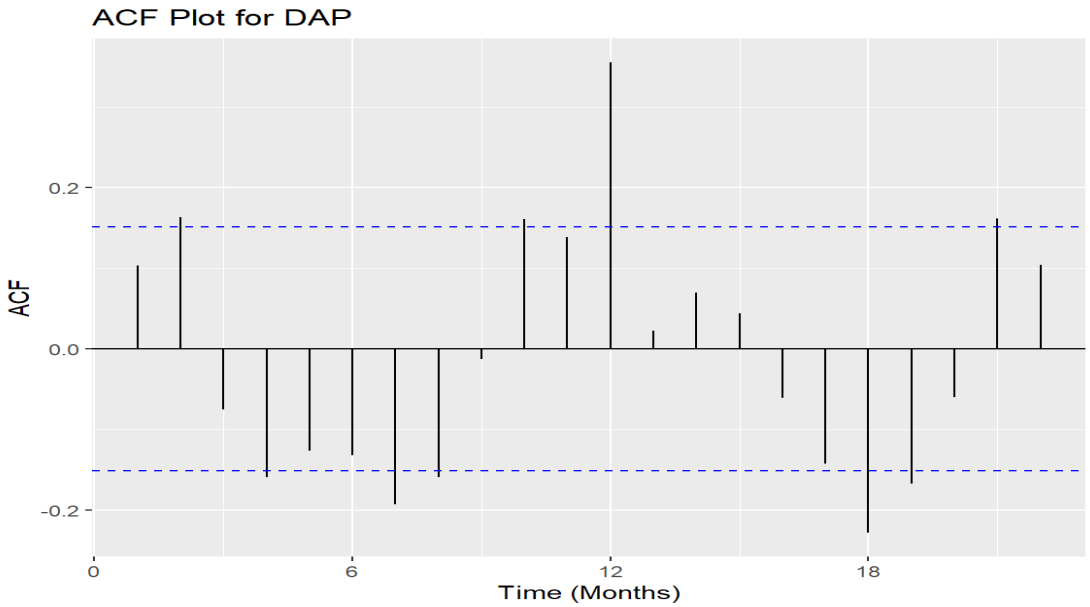


Figure 25: Autocorrelation Function plot for diammonium phosphate

The Partial Autocorrelation Function (PACF) plot in Figure 26 is for DAP demand. It presents the extent of correlation between the time series and its lagged values after removing the effects of intermediate lags. The PACF plot indicates a significant positive partial autocorrelation at lag 1, which implies that the previous month's demand directly influences the current month's demand, with no significant influence from the intermediate months. The spike at lag 12 in the PACF plot suggests that after controlling for the effects of lags 1 through 11, the demand from 12 months prior still directly influences the current demand, reinforcing the seasonal pattern observed in the ACF plot in Figure 25.

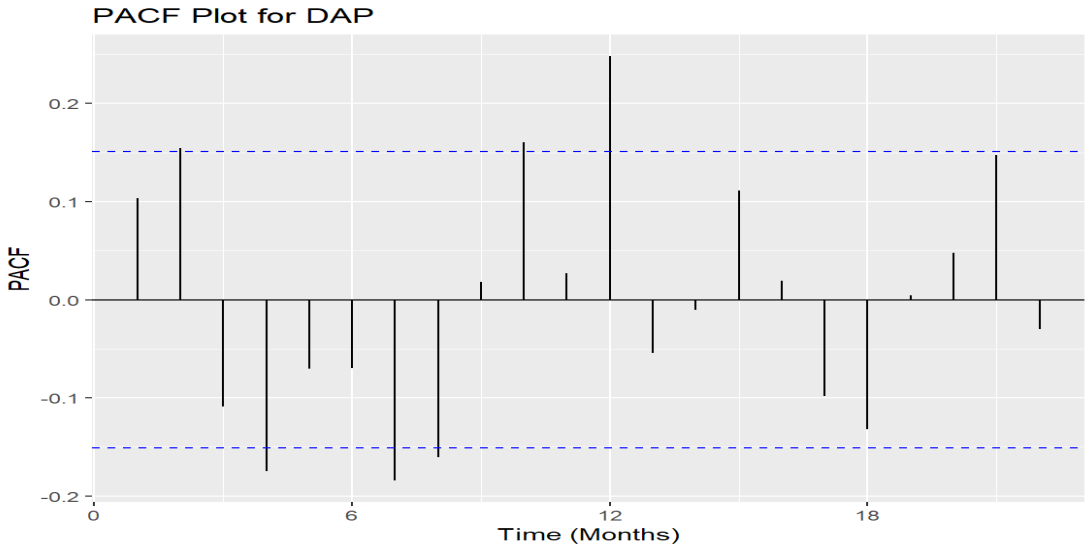


Figure 26: Partial autocorrelation function plot for diammonium phosphate

The ACF and PACF plots in Figures 25 and 26 suggest that the DAP demand series exhibits trend and seasonality. The trend is evident from the significant autocorrelation at lag 1, indicating that past demand values have a lasting effect on future demand. The spikes at lag 12 in both plots indicate the seasonality, pointing to a yearly cyclical pattern in demand. These characteristics suggest that a seasonal ARIMA model is appropriate for modelling this time series, incorporating terms that account for both the autoregressive behaviour and the seasonal effects. The significant lags observed in both ACF and PACF plots guide the selection of the order of the SARIMA model.

The seasonal patterns observed in DAP demand align with Kenya's bimodal agricultural calendar, where the long rains (March to May) and short rains (October to December) dictate planting seasons. DAP is used during the planting phase, which is why there is higher demand during these periods. The periods of zero demand likely correspond to off-seasons when top dressing is unnecessary. The National Fertiliser Subsidy Program, particularly the subsidy allocation changes, caused farmer preference shifts. The demand for DAP (Diammonium Phosphate) decreased in 2010 and 2014, likely due to shifts in government policies or cropping patterns. For instance, shifting from an e-voucher system to direct distribution through the National Cereals and Produce Board (NCPB) may have altered accessibility, affecting demand (Njagi *et al*., 2024).

DAP is primarily used during planting, especially for cereals like maize and rice (Yaseen *et al*., 2023). The high demand observed in January and March corresponds with the short rainy season, essential for crop establishment in Kenya. The consistent demand throughout the rest of the year suggests a balanced application across various crops to maintain soil fertility. Kenya's agricultural practices, which vary significantly by region and crop type, also explain the variability in fertiliser demand. DAP is essential for maize and other staple crops. For example, the high demand for DAP in January and March corresponds with the preparation for the long rains, a critical period for maize planting.

4.2.3 Visualisation of Muriate of Potash Time Series

The time series plot in Figure 27 illustrates the demand for Muriate of Potash (MP) fertiliser from 2010 to 2023. The data reveals that from 2010 to around 2021, demand levels were relatively low and stable, with occasional minor increases. However, starting in 2022, there is a sharp and significant rise in demand. This surge continues with substantial fluctuations until 2023, followed by a noticeable decline. This trend indicates a shift in demand, possibly due to changes in the market, policies, or agricultural practices. The pronounced peaks and variability in the later part of the series suggest high-demand periods and warrant further investigation into the causes. Overall, the time series shows an evident change in usage trends over the observed period. The sharp increase in demand for Muriate of Potash (MP) and other types of fertilisers can be attributed to the government's subsidy program announced in September 2022. Farmers were offered fertiliser at significantly reduced prices, with K. Sh 3.55 billion allocated to subsidise 71,000 tonnes of fertiliser, including MP. MP was available at K. Sh 1,775 per 50kg bag compared to the higher market price. This substantial price reduction, coupled with the government's directive to make 1.4 million bags of fertiliser available, drove a surge in demand as farmers sought to benefit from the subsidised rates for the short rains season (Ministry of Agriculture and Livestock Development, 2022).

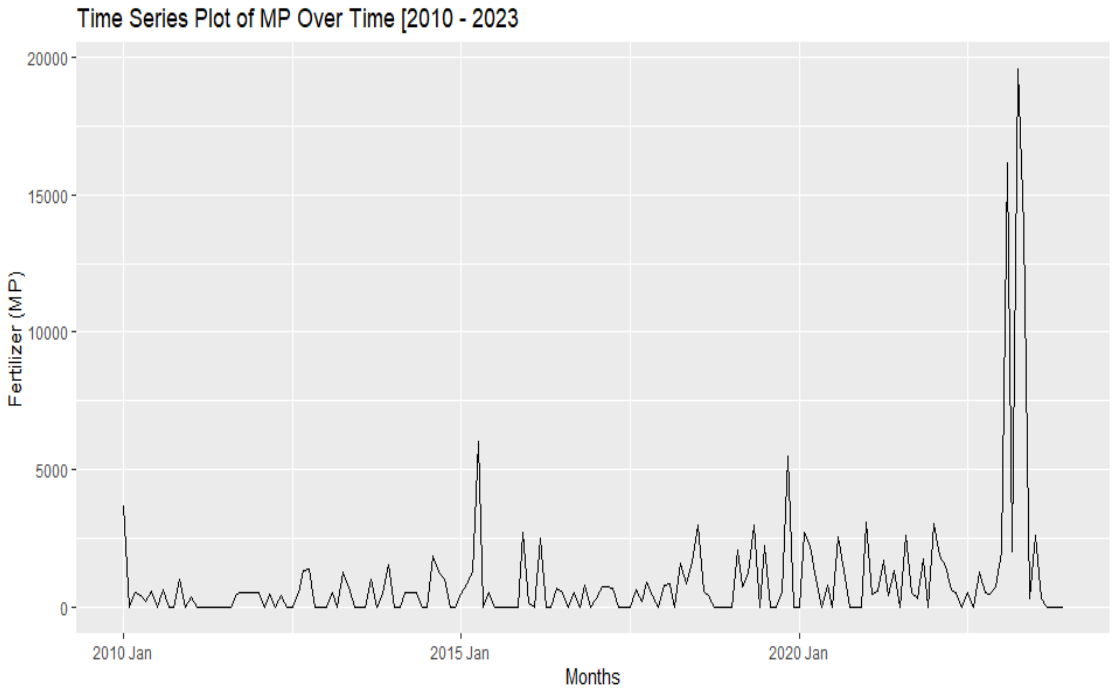


Figure 27: Muriate of potash demand time series plot

4.2.3.1 Normality Test of Muriate of Potash Time Series

In order to test if the data followed a normal distribution, the Jarque Berra test was done using the R program. The Jarque-Bera test on the Muriate of Potash fertiliser data test statistic (X-squared) was 10,690 with 2 degrees of freedom and a p-value of <0.0001. This led to rejecting the null hypothesis that the data is normally distributed. Therefore, the data does not follow a normal distribution, suggesting it may be significantly skewed or have excess kurtosis. Table 5 is a summary results for the Jarque-Bera Test for the Muriate of Potash time series data.

Table 5: Jarque Bera test for Muriate of Potash

Test	Statistic	Degrees of Freedom	p-value
Jarque-Bera Test	10690	2	< 0.0001

Figure 28 graph shows the seasonal changes in the demand for Muriate of Potash (MP) from 2010 to 2023. There are notable spikes in demand during February, April, and May, especially in 2014 and 2022. The pink line for 2014 and the blue line for 2022 highlight peaks reaching up to 20,000 units during these months. This suggests that these years saw unusually high demands in these months, likely due to factors such as agricultural cycles, market demands, or supply chain issues. The high increase in

demand for 2022 was mainly caused by government subsidies (Ministry of Agriculture and Livestock Development, 2022)

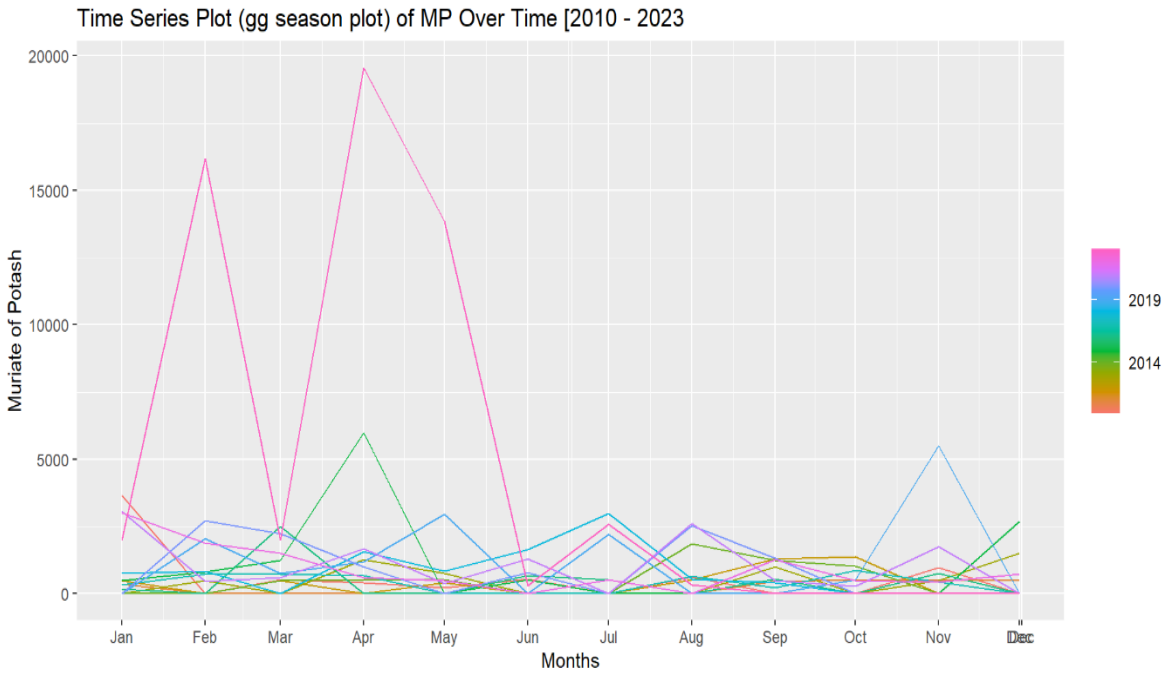


Figure 28: Time series gg season plot for Muriate of Potash 2010 - 2023

On the other hand, figure 29 offers a clearer view of the monthly distribution of MP demand over the years. By breaking down the data month by month, it becomes clear that the spikes in the Figure 27 graph are outliers compared to the overall trend. The graph shows that most months have low and stable demand, with occasional peaks. The horizontal blue lines represent the average demand for each month, indicating that the high-demand periods are anomalies rather than consistent trends. The variability within each month over the years suggests that occasional disruptions cause significant variations in MP fertiliser demand.

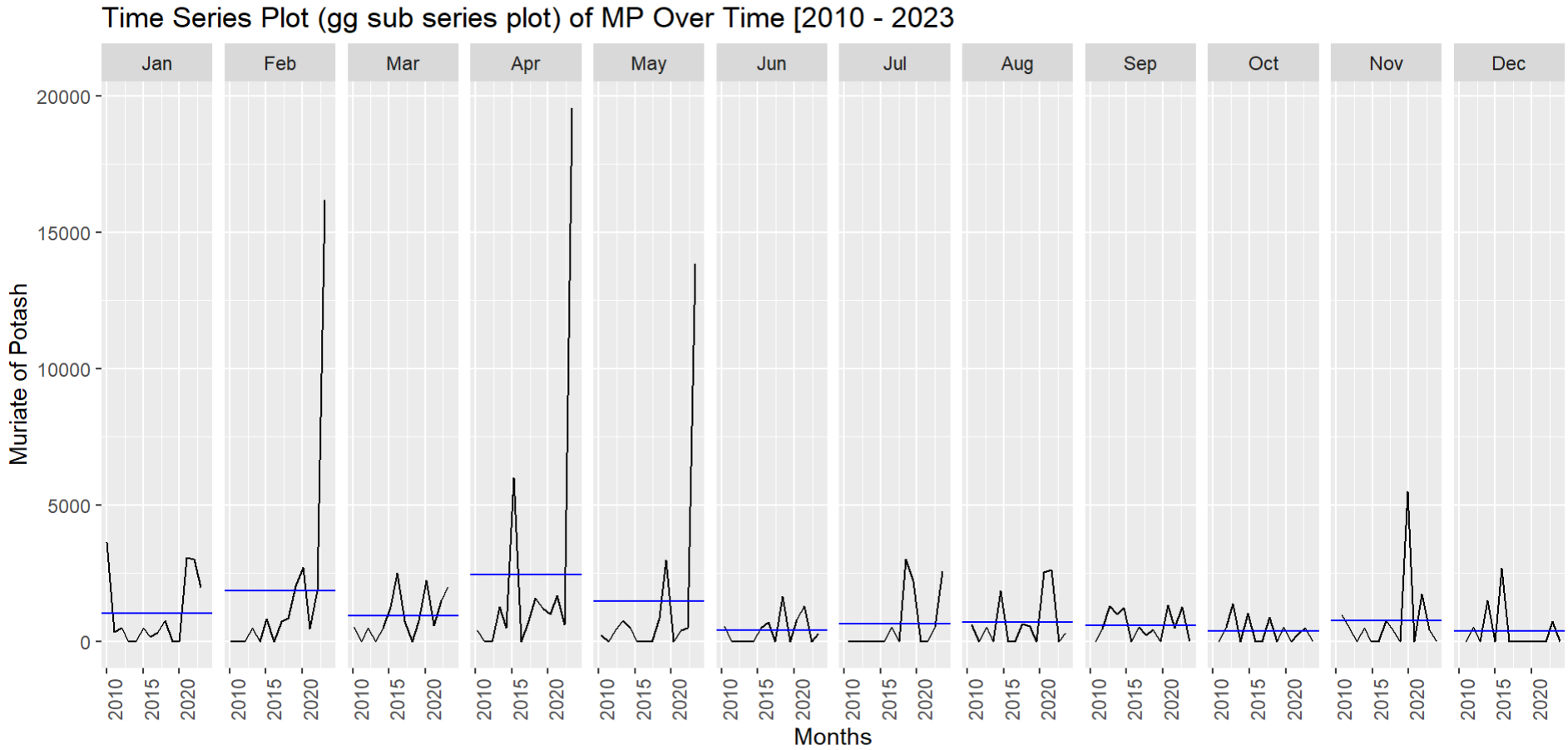


Figure 29: gg sub series plots of muriate of potash fertiliser demand

The behaviour of the plots suggests that the overall trend in MP demand is relatively stable, with occasional periods of high demand. The noticeable peaks in specific months and years imply that external factors significantly influence MP demand. In addition, the findings point out that the muriate of potash demand is high in February, April, and May. The high demand for Muriate of Potash (MP) in February, April, and May is likely due to its critical role in supporting crop growth during key agricultural periods. Farmers prepare for the upcoming planting season in February, applying MP to enhance soil potassium levels essential for early crop development. The peak in April and May aligns with the active growing season when crops require increased potassium for optimal growth and development. This timing ensures that plants have sufficient nutrients to thrive during critical stages of their growth cycle, thereby driving up the demand for MOP during these months (Njagi *et al*. 2024).

4.2.3.2 Seasonal Decomposition of Muriate of Potash Time Series

The trend component in the STL decomposition (in Figure 30) shows the long-term changes in the Muriate of Potash (MP) Time series. From 2010, the trend declined and levelled off, maintaining stability until late 2020. After that, the trend sharply rises, reaching a peak around 2022, indicating a significant value increase. Following 2023, the trend declines but stays higher than the earlier stable period, indicating an upward shift despite the recent drop. This analysis highlights the significant changes in the data, especially the steep rise and subsequent fall in recent years. The seasonality of the data is also clearly visible from 2019, with significant variability from 2019 to 2023.

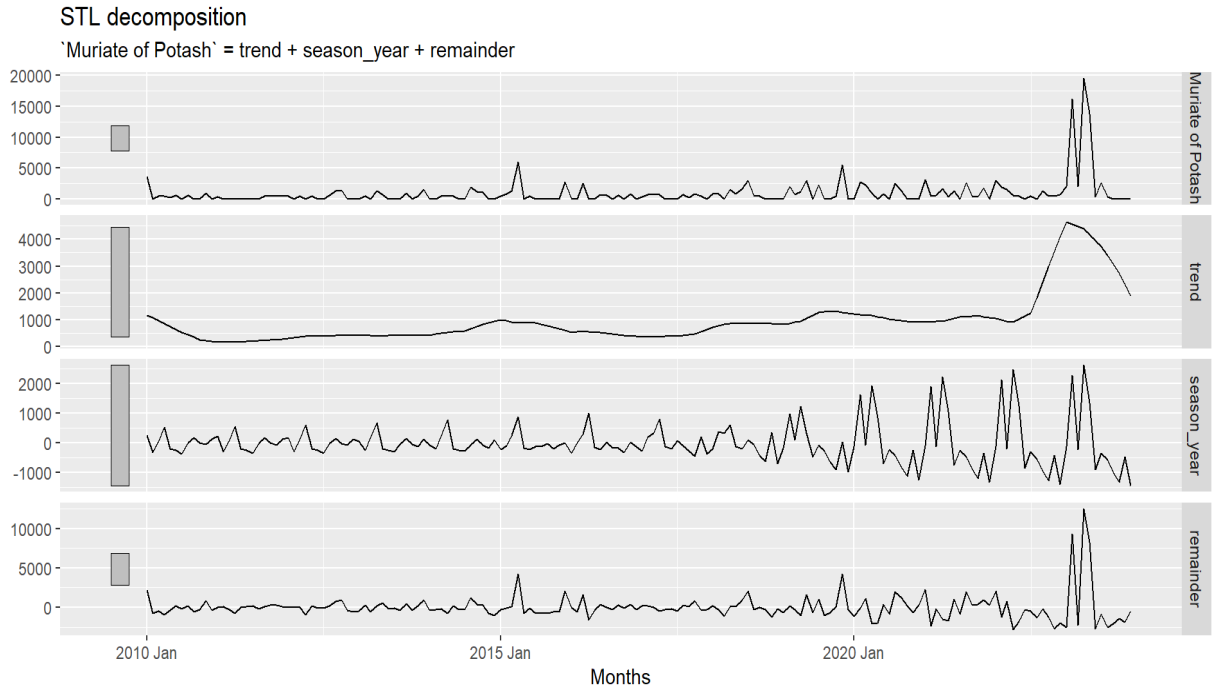


Figure 30: Seasonal-Trend Decomposition plot for of Muriate of Potash

The seasonal variation plots illustrate this well. The highest peaks of demand were experienced in 2019-2023, indicating that the demand for muriate potash fertiliser increased significantly after the COVID-19 pandemic. The increase may be due to increased production levels and government policies after introducing fertiliser subsidies.

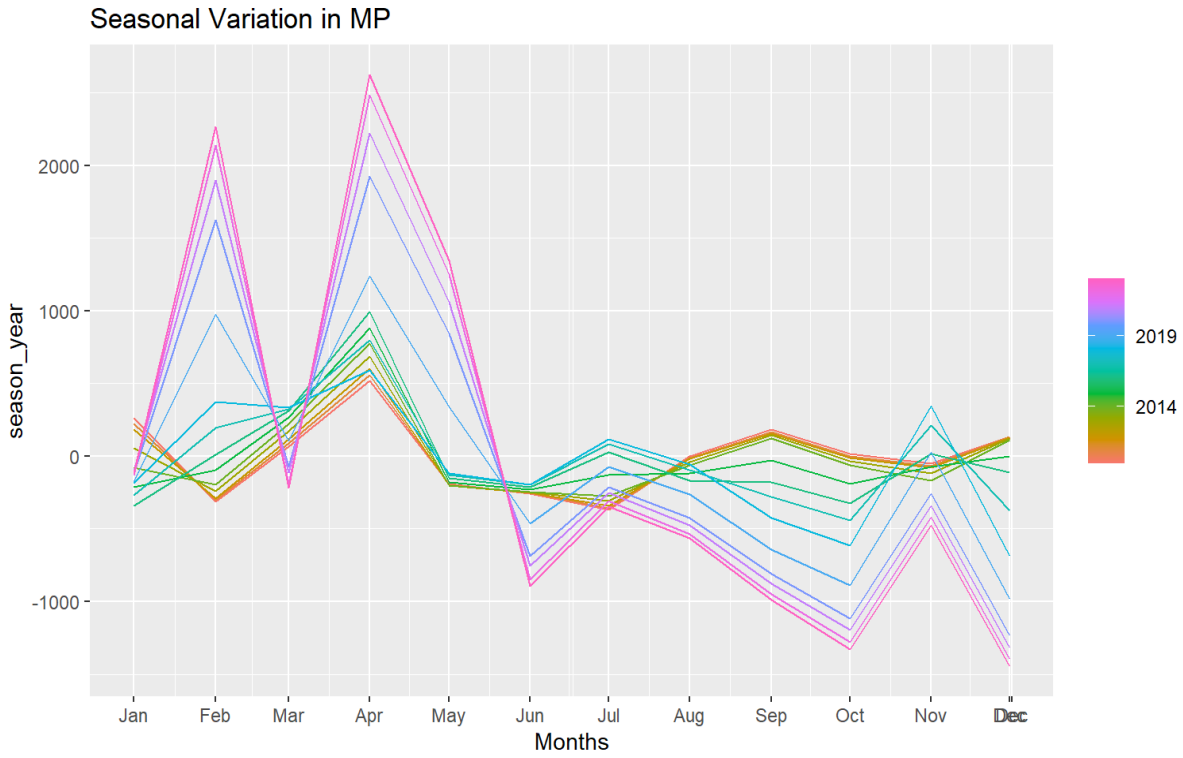


Figure 31: Seasonal variation in Muriate of Potash

The visualisation of the Muriate of Potash (MP) demand for each month indicated that the demand significantly rose during February, March, April, and May. This increased demand coincides with the agricultural planting season when farmers prepare their fields and plant potassium-dependent crops. As MP is a crucial source of potassium, it becomes essential during this time. Conversely, demand remains low in other months, likely due to the off-season when crop cultivation is minimal, and the need for fertilisers decreases. This seasonal trend illustrates how fertiliser application is synchronized with the growth cycles of various crops.

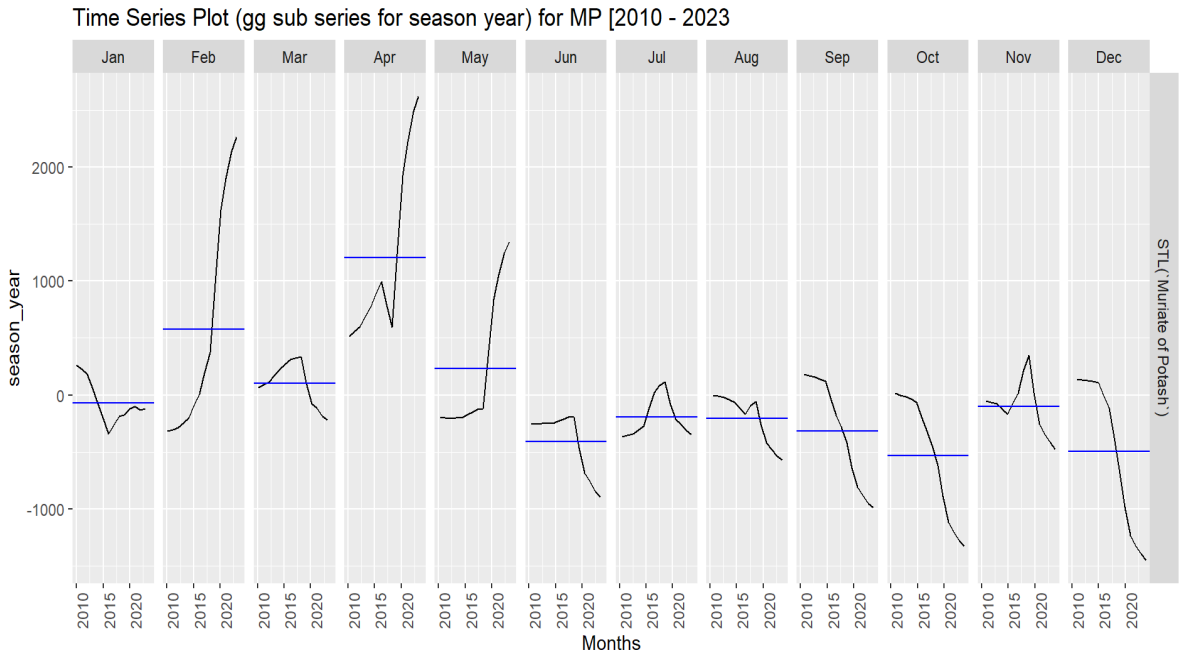


Figure 32: gg subseries plot for season year for Muriate of Potash (2010 - 2023)

The sharp increase in demand for fertilisers between 2022 and 2023 can be attributed to the government's intervention. In September 2022, the government introduced a substantial subsidy programme, providing 1.4 million bags of fertiliser at significantly reduced prices. This policy aimed to make fertilisers more affordable and accessible for farmers, resulting in a sharp rise in demand during this period as farmers took advantage of the lower costs. The findings indicated that the demand for muriate of potash increased in February and April (figure 31 and 32)

4.2.3.3 Testing for Stationarity Muriate of Potash Time Series

To determine if the data on the demand for Muriate of Potash was stationary, the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test was used. The test results indicated a statistic of approximately 0.602 and a p-value of about 0.022. In the KPSS test, a low p-value (usually less than 0.05) suggests rejecting the null hypothesis of stationarity, meaning the data series is likely not stationary. The test statistic also supports this, as higher values generally indicate non-stationarity. These findings made differencing the data necessary to make it stationary. The amount of differencing needed was determined using the 'unitroot_ndiffs' function, which showed that a first difference (d = 1) was sufficient. This process involves taking the difference between consecutive

observations, stabilizing the series' mean and variance over time, and addressing trends or seasonality in the original data.

The ACF (Autocorrelation Function) plot shows that the Muriate of Potash time series data correlates with its past values over different periods. There are notable positive correlations at lags 1 through 3, indicated by the spikes above the blue dashed significance threshold lines. This means that the data points within these lags are strongly related. Beyond lag 3, the correlations drop sharply and stay close to zero, with only a few minor spikes, suggesting that the time series has short-term dependencies up to about three months. This indicates that values beyond lag three do not significantly influence the current values.

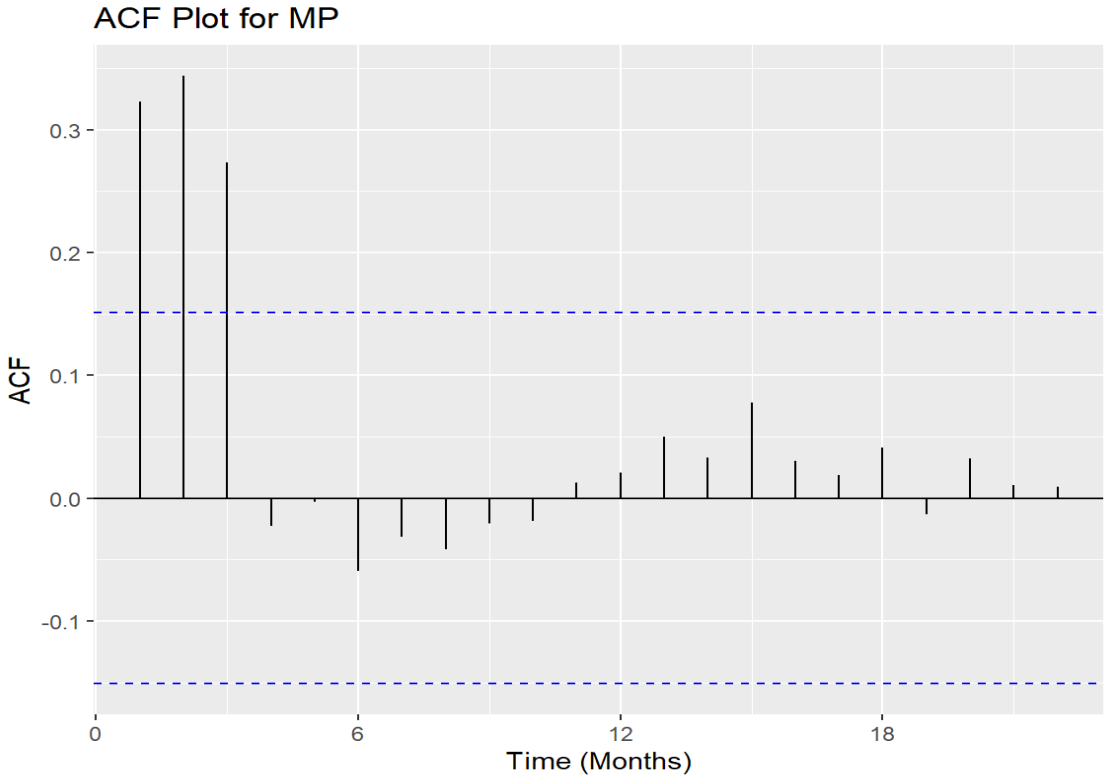


Figure 33: Autocorrelation function plot for Muriate of Potash

The PACF (Partial Autocorrelation Function) plot, on the other hand, shows the partial correlation of the time series with its lagged values, removing the influence of intermediate lags. This plot shows significant partial correlations at lags 1 through 3, with the first lag being the most significant. Beyond lag 3, the partial correlations

decrease and stay primarily within the significance thresholds. This means the time series is influenced by its recent past values, with direct dependencies up to about three months. The ACF and PACF plots suggest that the Muriate of Potash time series has notable short-term autocorrelation, essential for accurate time series forecasting models.

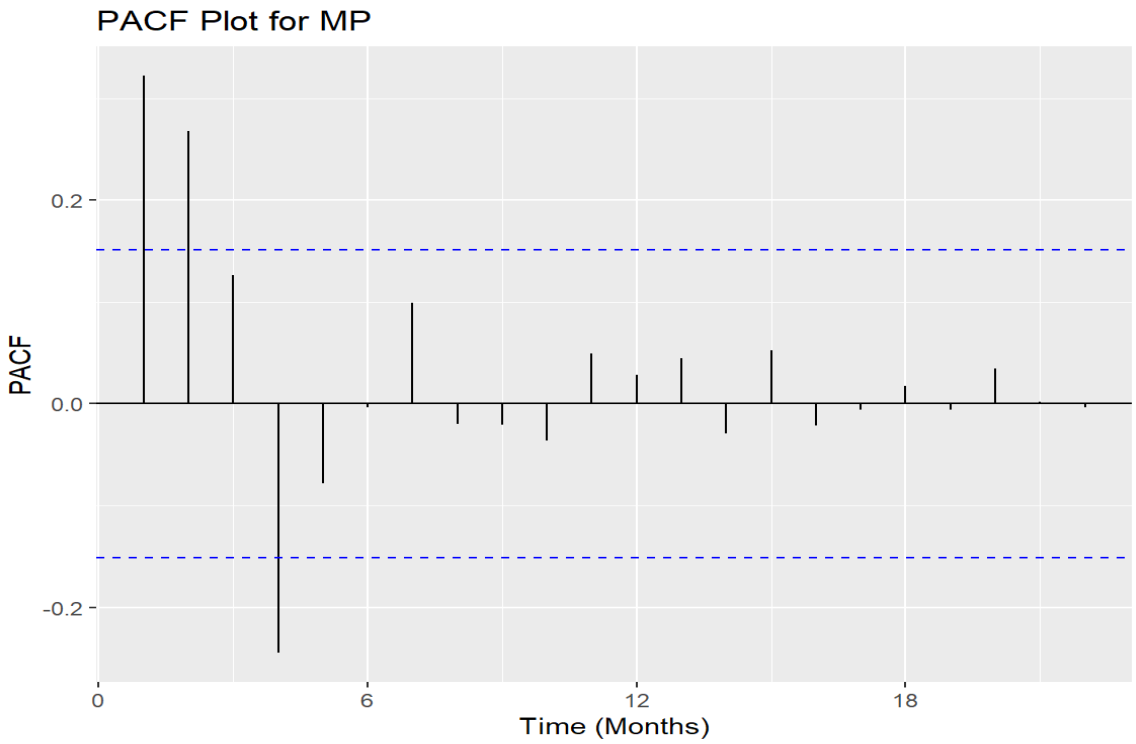


Figure 34: Partial autocorrelation function plot for Muriate of Potash

Muriate of Potash demand remained steady but saw significant peaks between 2020 and 2023, particularly in February, April, and May. These peaks might be linked to the increased production of cash crops like tea and coffee, which require potassium-rich fertilisers to enhance yield and quality (El Gayar, 2021). The timing of these peaks suggests that farmers are optimizing fertilization practices to coincide with critical growth stages before the main harvest seasons.

4.2.4 Visualisation of Nitrogen, Phosphorus, and Potassium fertiliser Time Series

The time series plot for NPK fertiliser demand from 2010 to 2023 reveals a distinct seasonal pattern with consistent peaks and troughs, indicating periodic demand fluctuations (figure 35). Each year shows multiple spikes, suggesting increased

fertiliser usage during certain times, likely aligning with planting or growing seasons. The height of these peaks has varied over the years, with significantly higher peaks around 2020 to 2022, indicating a substantial rise in demand during these years. After 2022, demand drops sharply, indicating a possible shift or decrease in fertiliser usage. The plot illustrates regular seasonal demand cycles with occasional years of heightened activity.

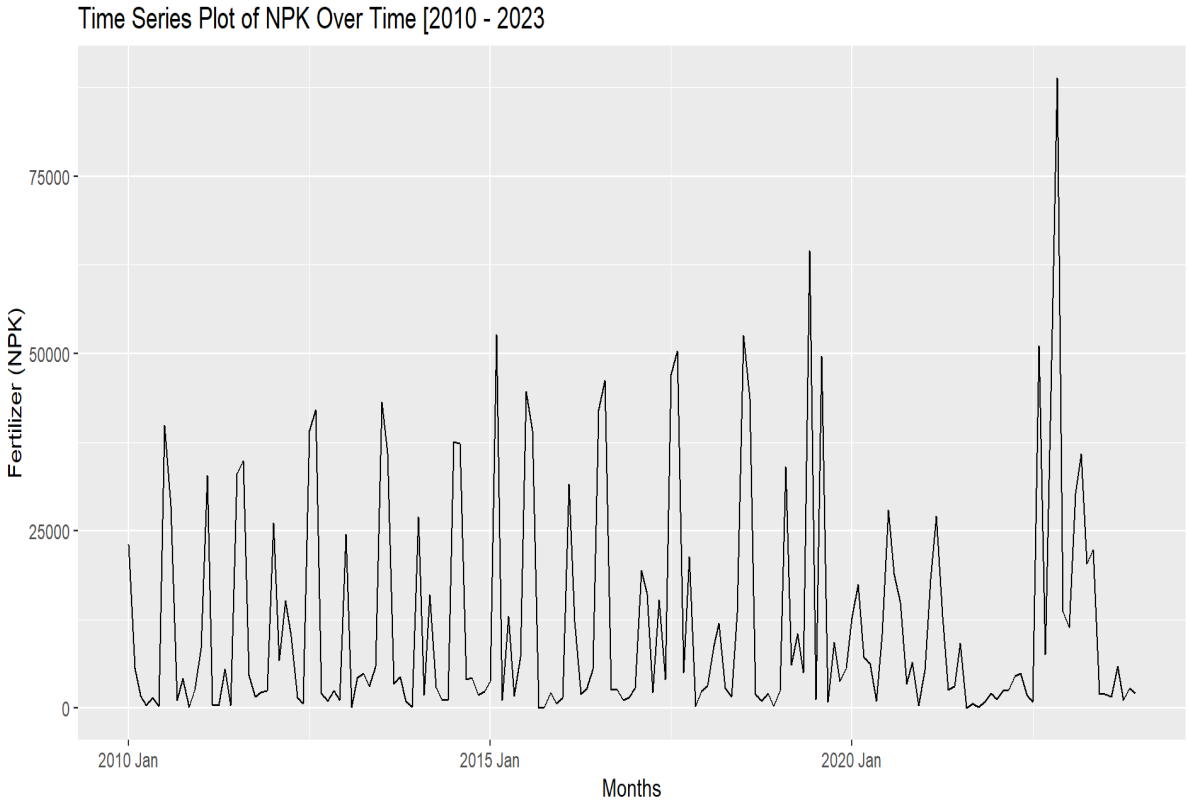


Figure 35: Time Series plot for nitrogen, phosphorus, and potassium fertiliser (2010-2023)

4.2.4.1 Seasonality Test of Nitrogen, Phosphorus, and Potassium Fertiliser Time Series

The season plot in Figure 36 illustrates the monthly demand for Nitrogen, Phosphorus, and Potassium Fertiliser (NPK) from 2010 to 2023, with different colours representing each year (Appendix 4). Key findings from this plot show notable peaks in February, June, and October across several years, indicating high demand during these months. February 2014 is particularly significant, with demand exceeding 50,000 units. This cyclical trend, with increased demand around June and July, followed by another peak in October, suggests a possible connection to agricultural cycles or planting seasons.

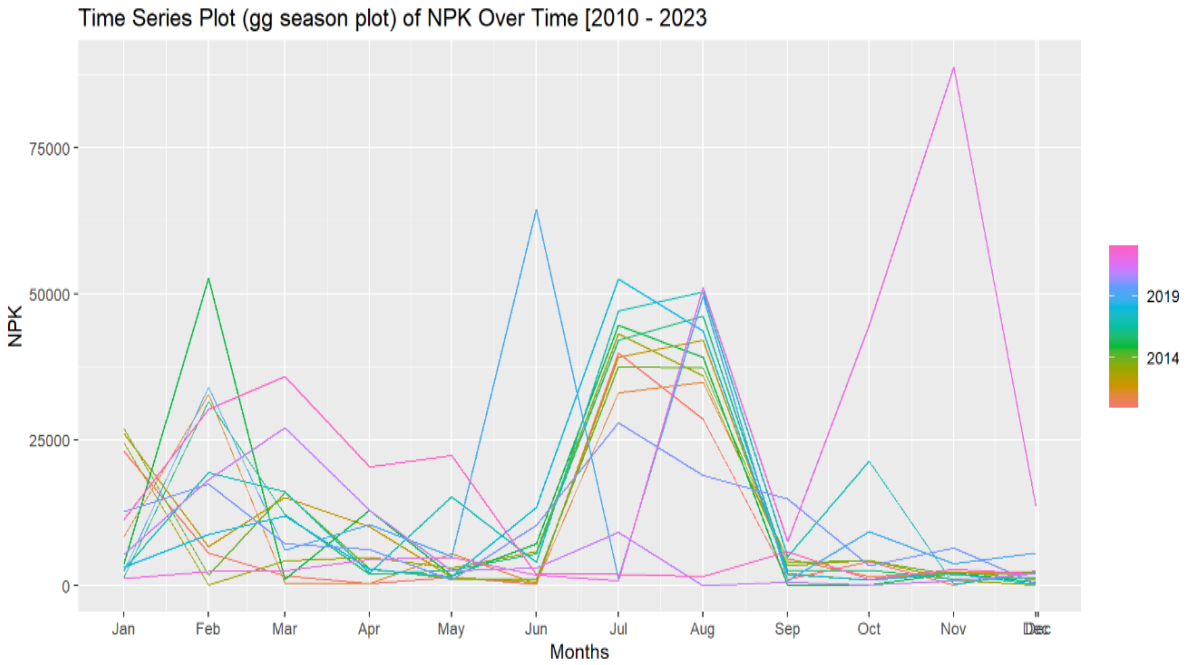


Figure 36: Season plot of nitrogen, phosphorus, and potassium fertiliser (2010 - 2023)

The subseries plot breaks down the data by month, showing each month's demand over the years. February, June, July, and August stand out with significant spikes, indicating that the demand for this fertiliser type is high in these months. November also shows notable spikes in some years, which might be linked to specific agricultural practices or seasonal events.

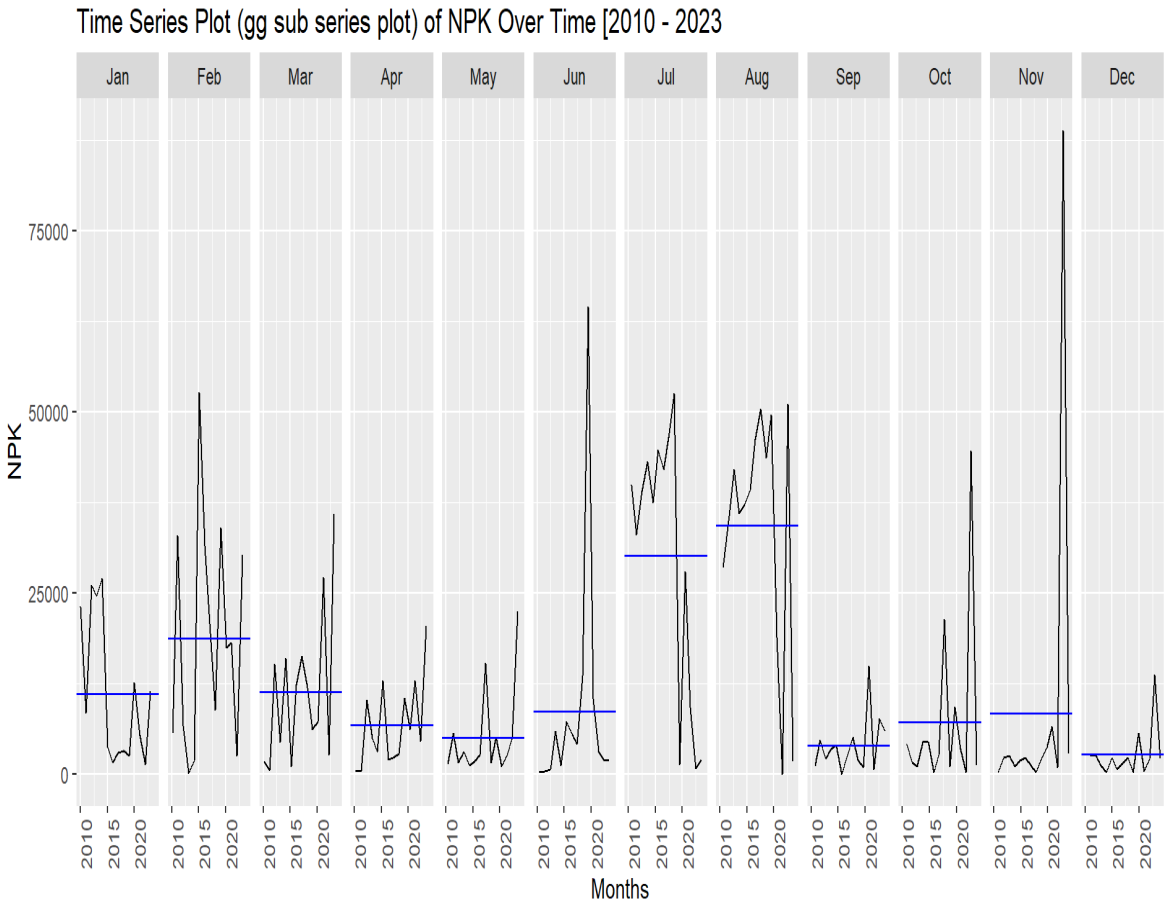


Figure 37: Subseries plot for nitrogen, phosphorus, and potassium fertiliser (2010 - 2023)

Overall, the data indicate that it is not normally distributed due to visible clear peaks and valleys, indicating high and low demand periods. The pattern is seasonal, with recurring peaks in February, June, October, and November. February, especially in 2014, shows the highest demand, potentially due to specific events or anomalies that year. These insights could help guide decisions regarding fertiliser supply chains, marketing strategies, and inventory planning to match demand patterns better.

4.2.4.2 Seasonal Decomposition of Nitrogen, Phosphorus, and Potassium Fertiliser Time Series

Figure 38 is the STL Decomposition, which breaks the NPK time series into three parts: trend, seasonal, and remainder. The first panel displays the original NPK time series, the second shows the trend component, reflecting the long-term changes in NPK, the third panel highlights the seasonal component, which captures the repeating patterns

each year, and the fourth panel illustrates the remainder, or residual component, which represents the irregular fluctuations or noise after accounting for trend and seasonality.

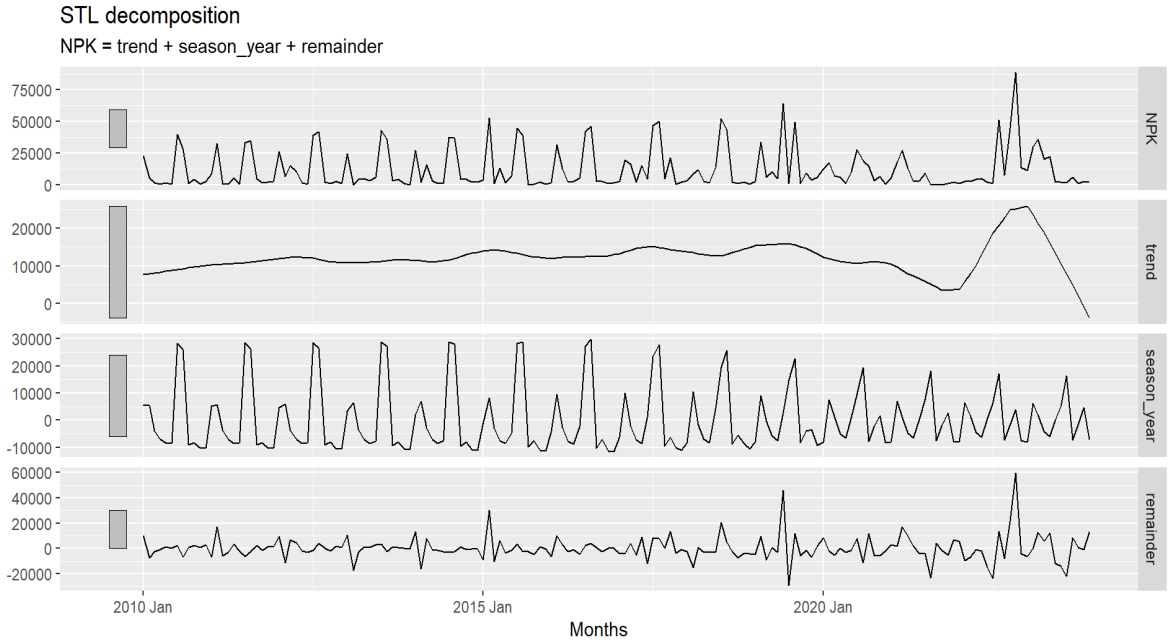


Figure 38: Seasonal-Trend Decomposition plot for nitrogen, phosphorus, and potassium fertiliser (2010 - 2023)

Figure 38 is the Time Series Plot for Nitrogen, Phosphorus, and Potassium Fertiliser (NPK) 2010 – 2023, and it illustrates how the NPK variable changes seasonally each month from 2010 to 2023. Each panel in the plot represents a different month, showing how the seasonal component of the data varies each year within that month. The plot indicates that the demand for NPK was high during July and August. The demand for NPK was at its lowest during December. Black lines illustrate the seasonal pattern for each year, while blue horizontal lines show the average seasonal value for each month over all the years. This plot highlights the recurring seasonal trends and their annual variations.

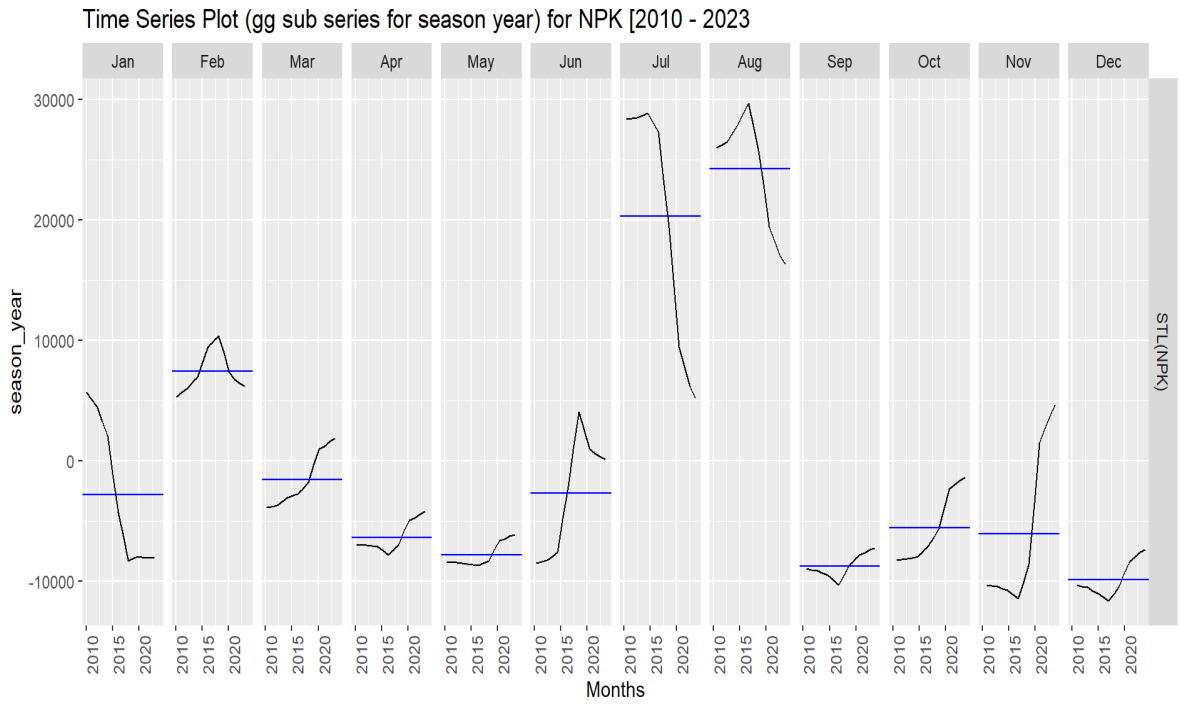


Figure 39: gg subseries plot for season year for nitrogen, phosphorus, and potassium fertiliser

Figure 40 is a presentation of Seasonal Variation in NPK. The seasonality of NPK is illustrated by plotting the seasonal component for each year on a single graph. Each line represents a different year, with colours denoting different years from 2014 to 2019 (Appendix 4). This plot allows for a clear comparison of how seasonal patterns change over time, with peaks and troughs corresponding to specific months, demonstrating the regular and varying seasonal behaviour of NPK. The trends in this figure indicate varying demand for NPK in different months, indicating that the demand for NPK fertiliser is seasonal, with high demand for it in June, July, and August.

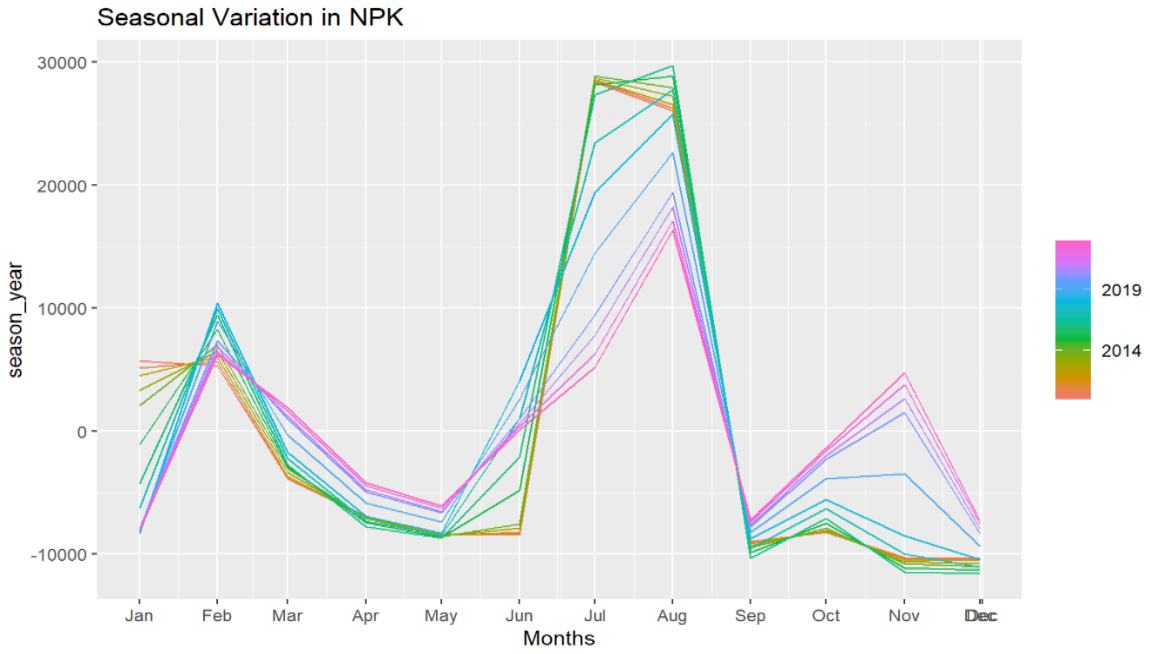


Figure 40: Seasonal variation in nitrogen, phosphorus, and potassium fertiliser (2010 - 2023)

4.2.4.3 Testing Stationarity of NPK Time Series

A KPSS test was done to test if the data was stationary. The KPSS test for the NPK data produced a test statistic of 0.0562 and a p-value of 0.1. The p-value exceeds the typical significance level of 0.05, which leads to acceptance of the null hypothesis, indicating that the data is stationary around a constant mean or trend. This means the NPK data does not show significant signs of non-stationarity or trends, so it can be regarded as stationary.

The ACF (Autocorrelation Function) plot for the NPK data shows how each value in the time series correlates with its past values over various lags. Significant spikes at lags 1, 6, and 12 indicate strong correlations at these intervals. The spike at lag 1 suggests a robust immediate relationship between each data point and its preceding value. The spikes at lags 6 and 12 suggest recurring patterns or periodic influences, possibly indicating semi-annual and annual cycles. The blue dashed lines represent the 95% confidence intervals, with any spikes outside these bounds being statistically significant, reinforcing these observations.

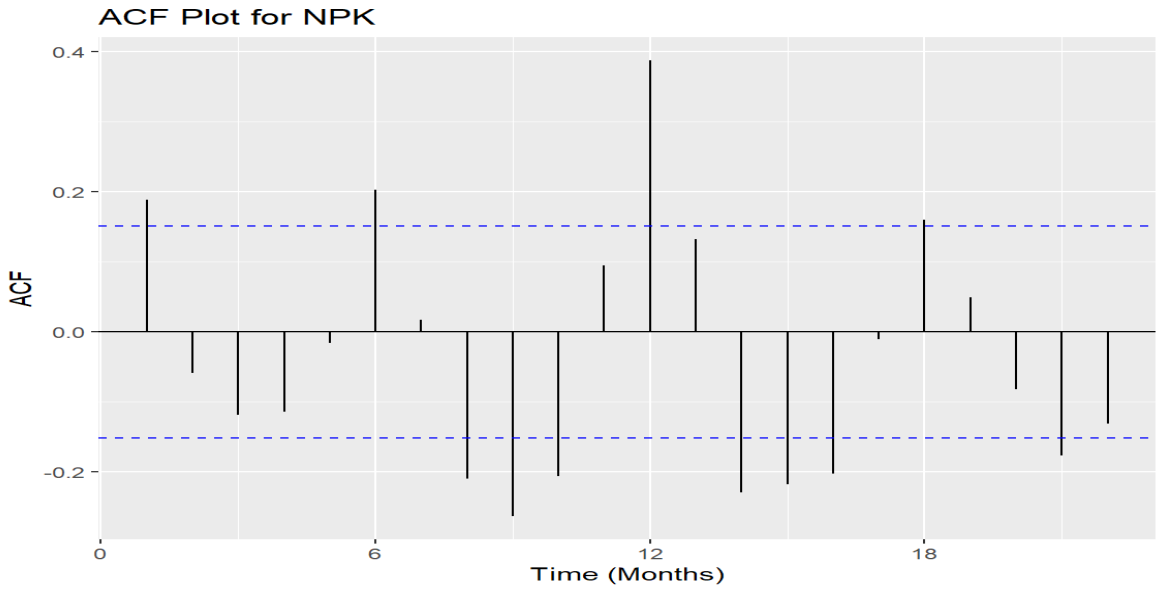


Figure 41: Autocorrelation function plot for Nitrogen, Phosphorus, and Potassium Fertiliser (2010-2023)

The PACF (Partial Autocorrelation Function) plot offers a more detailed view by measuring the correlation of data points with their lagged values while accounting for the effects of intermediate lags. Significant spikes at lags 1, 6, and 12 indicate direct relationships at these intervals, independent of shorter lag correlations. This supports the presence of periodic patterns or seasonality within the data. The spike at lag 1 confirms a strong immediate relationship, while the spikes at lags 6 and 12 indicate recurring semi-annual and annual cycles. These insights are crucial for identifying underlying trends and making accurate forecasts, highlighting the most predictable time intervals in the data.

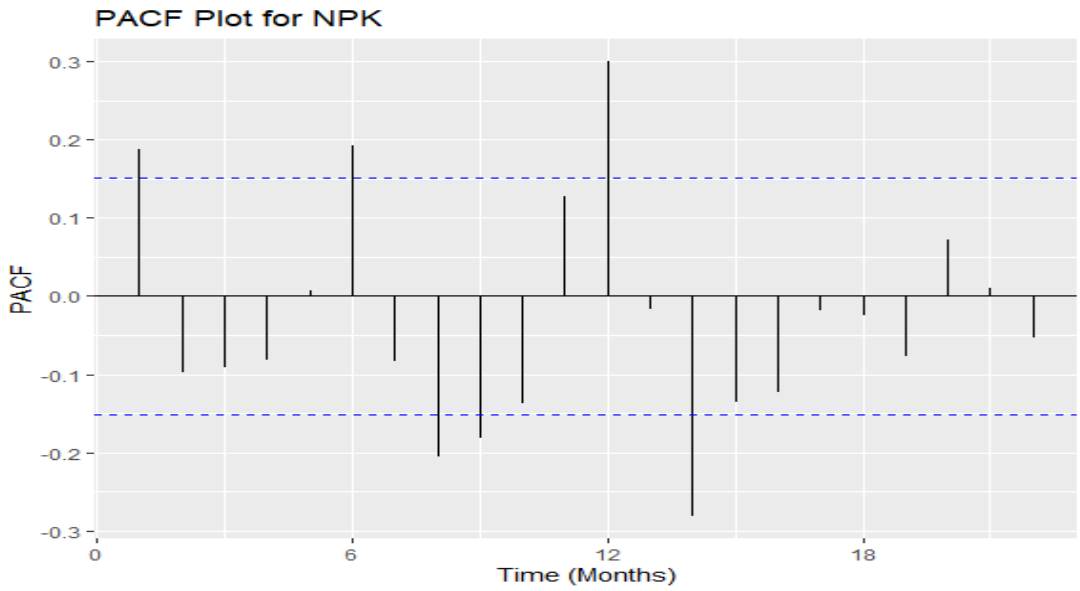


Figure 42: Partial autocorrelation function plot for nitrogen, phosphorus, and potassium fertiliser (2010-2023)

NPK (Nitrogen, Phosphorus, and Potassium) fertilisers showed notable peaks from 2020 to 2022, followed by a sharp decline after 2022. This signifies a shift in agricultural practices or the influence of external factors such as price fluctuations or changes in subsidy programs. The high demand during July and August is consistent with the long rainy season in Kenya, which is the primary planting period for many crops. The subsequent decline was due to reduced availability or increased costs of fertiliser in 2020, which prompted farmers to find alternatives or reduce application rates (GOK, 2022). The sharp peaks in demand for NPK around 2020-2022, followed by a sharp drop, reflect the influence of rising global fertiliser prices and the impact of the Kenyan shilling's depreciation (GOK, 2022).

4.2.5 Visualisation of Urea Fertiliser Demand Time Series

The time series plot in Figure 43 shows the demand for urea fertiliser from January 2010 to 2023. The data displays clear seasonal patterns, with regular highs and lows each year, likely aligning with planting and harvesting seasons. There is no noticeable long-term trend of rising or falling demand, as the variations in demand vary reasonably consistently over time. However, the size of these peaks varies, which is an indication that there are significant fluctuations in actual demand. In recent years, especially from 2020 onward, there have been more pronounced peaks, which indicate increased

demand or volatility. The plot shows urea fertiliser demand is strongly seasonal, with year-to-year differences in the demand required for production.

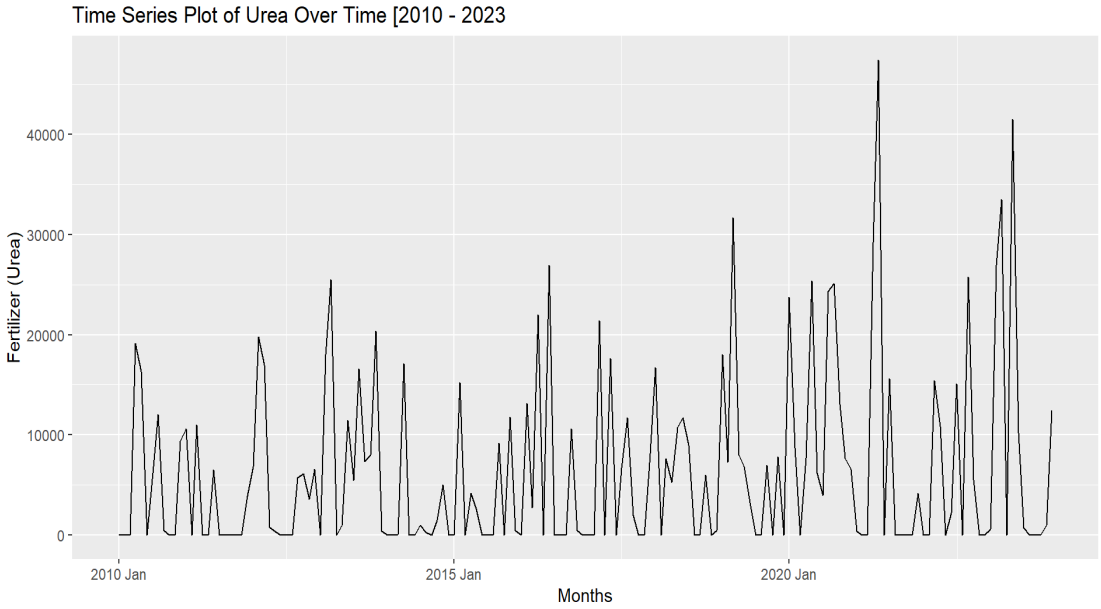


Figure 43: Time series plot for Urea (2010 - 2023)

4.2.5.1 Normality Test for Urea Time Series

In order to test whether the data was stationary, Jarque Berra test statistics were conducted. The test produced a statistic (X-squared) of 152.66 with 2 degrees of freedom and a p-value of <0.0001. These results significantly depart from normality, rejecting the null hypothesis that the data is normally distributed. Consequently, the Urea demand does not follow a normal distribution and displays notable skewness and/or kurtosis. In order to check the characteristics and trend of the demand throughout the year 2010-2023, a GG-Seasonal plot and a GG-Sub-seasonal plot were used. Figures 43 and 44 are the plots for the demand for Urea over these years. Figure 44 displays a time series of Urea data from 2010 to 2023, each year in different colours (Appendix 4). Notable peaks are observed in certain months, such as February, May, and September, suggesting possible seasonal influences on Urea levels. In Figure 45, the annual trends within each month are well visualized, confirming variability and potential long-term changes in Urea levels, as indicated in Figure 44. The blue horizontal lines indicate the monthly average for each year, providing a visual reference for comparing fluctuations within each month. The plot shows that the average demand for urea fertiliser is in February, March, April, and May. The behaviour of this demand

may be explained by changes in different factors such as planting seasons, weather conditions, or agricultural practices affecting Urea usage or availability. The second plot's focus on monthly sub-series across years helps identify any annual trends or anomalies within specific months, highlighting whether certain months have shown consistent increases or decreases. This indicates broader changes in agricultural practices, economic conditions, or policy impacts.

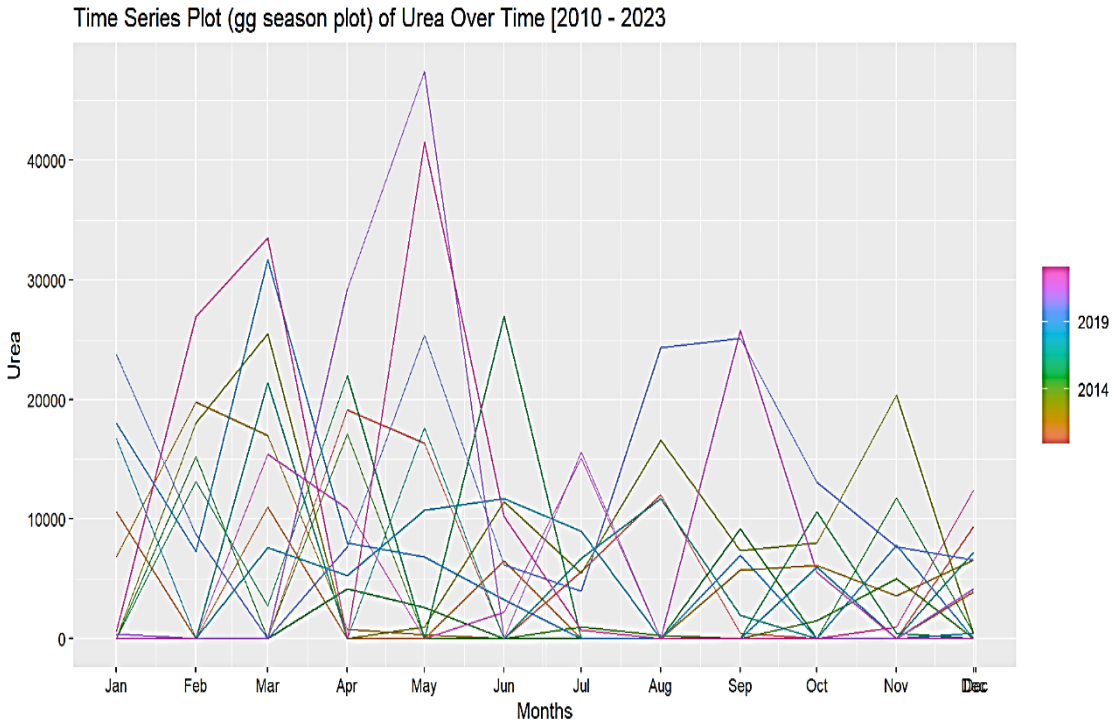


Figure 44: gg season plot for Urea fertiliser demand (2010 – 2023)

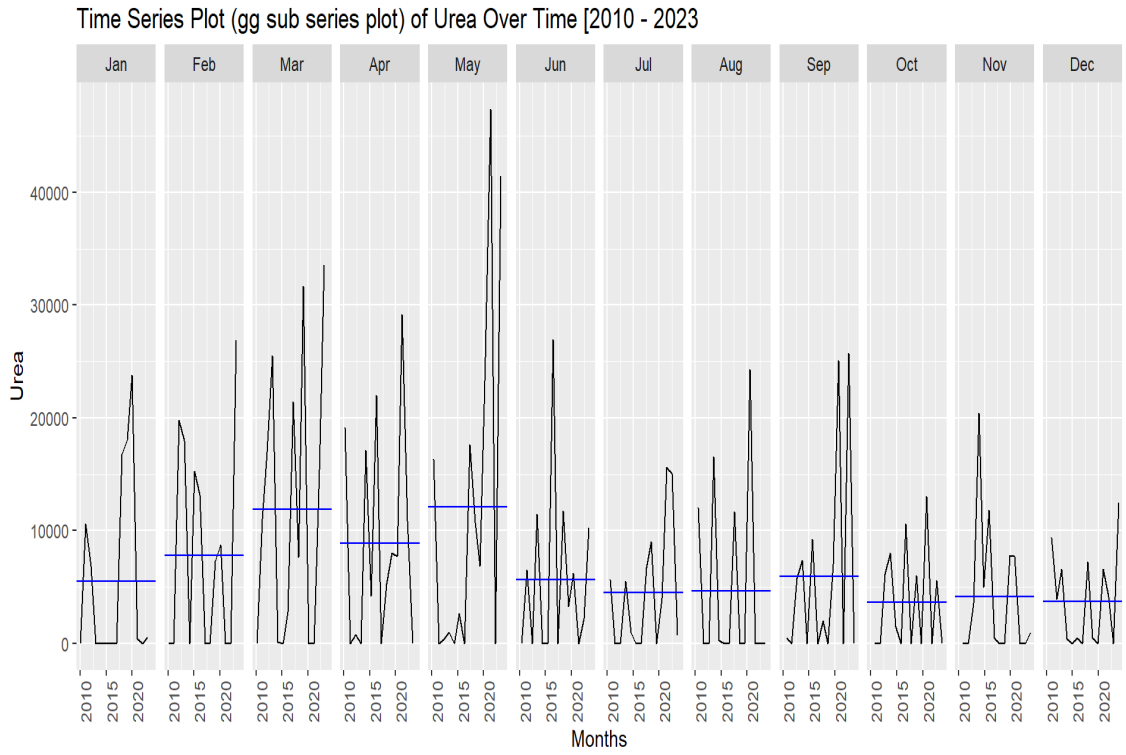


Figure 45: gg subseries plot for Urea fertiliser demand

4.2.5.2 Seasonal Decomposition of Urea Time Series

After conducting STL, the graph generated showed the time series data of Urea decomposed into three components: trend, seasonal, and remainder. The top panel presents the raw Urea data over time; the second panel displays the trend component, reflecting the long-term changes in Urea levels; the third panel illustrates the seasonal component, showcasing the recurring seasonal patterns, and the bottom panel represents the remainder, capturing the residuals after accounting for the trend and seasonality. The presence of clear and consistent seasonal patterns in the seasonal component confirms the seasonality in the data, highlighting the suitability of using the SARIMA (Seasonal Autoregressive Integrated Moving Average) model for accurate forecasting. The SARIMA model can account for seasonal and non-seasonal factors, making it ideal for this time series with evident seasonal variations.

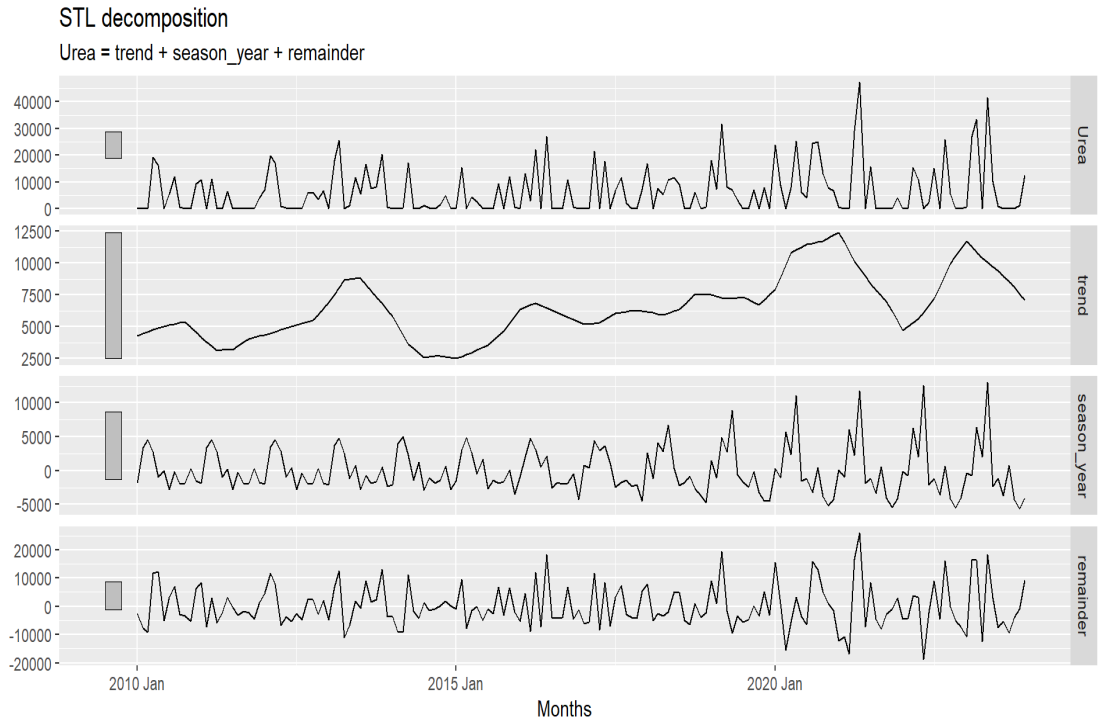


Figure 46: Seasonal-Trend Decomposition plot for Urea time series

The seasonality of the demand for urea is also made clear by the seasonal variation graph and the seasonal subseries plot. These plots indicate consistent demand variation each month. Feb, March, April, and May have the highest average demand, suggesting that more fertiliser is consumed in these months. Figures 47 and 48 represent the GG-Seasonal variation for Urea (2010-2023) and the GG-Sub-series for Urea (2010-2023).

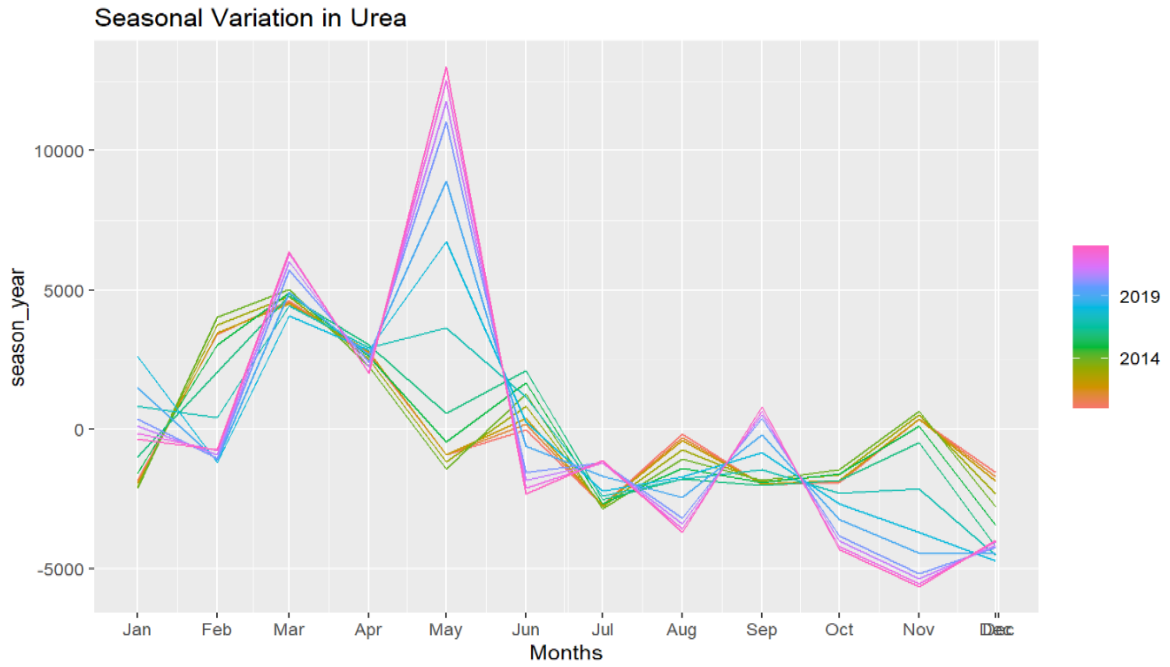


Figure 47: gg seasonal variation plot for Urea fertiliser demand

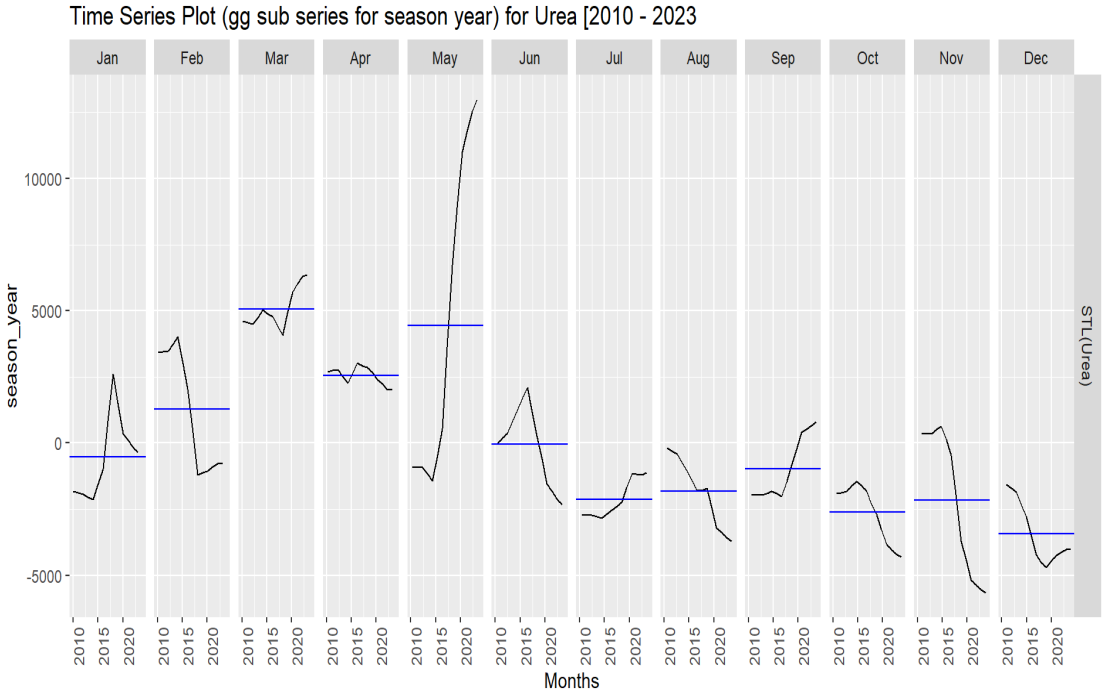


Figure 48: gg subseries plot for season year for Urea time series

4.2.5.3 Stationarity Test for Urea Demand

After conducting a stationarity test using the KPSS test, the results yielded a KPSS statistic of 0.4687569 and a p-value of 0.0487034, which led to rejection of the null hypothesis that the data was stationary and concluded that the data was not stationary

and 95% confidence level. To address this, a differencing test was performed using the R program, which determined that differencing the data once was sufficient to achieve stationarity. This adjustment ensures that the time series data is appropriate for further analysis and modelling.

The ACF (Autocorrelation function) plot in Figure 49 illustrates how the time series correlates with its past values. Significant spikes exceed the blue dashed lines, indicating lags with statistically significant autocorrelation. The plot reveals notable autocorrelations at several lags, especially at 1, 6, and 12 months, suggesting a seasonal pattern in the Urea time series. The positive autocorrelation at lag 1 implies that urea levels one month apart are positively correlated, and the significant spikes at longer lags may indicate recurring patterns annually. On the other hand, the PACF plot in Figure 50 displays the partial correlation of the time series with its past values, adjusting for all shorter lags. The PACF plot for urea also shows significant spikes at lags 1, 6, and 12 months, consistent with the ACF plot's indication of seasonality. The spike at lag 1 suggests that the value of urea one month prior is a crucial factor in the autoregressive model, while the significant partial autocorrelations at longer lags (6 and 12 months) further confirm the presence of seasonal and periodic influences in the data.

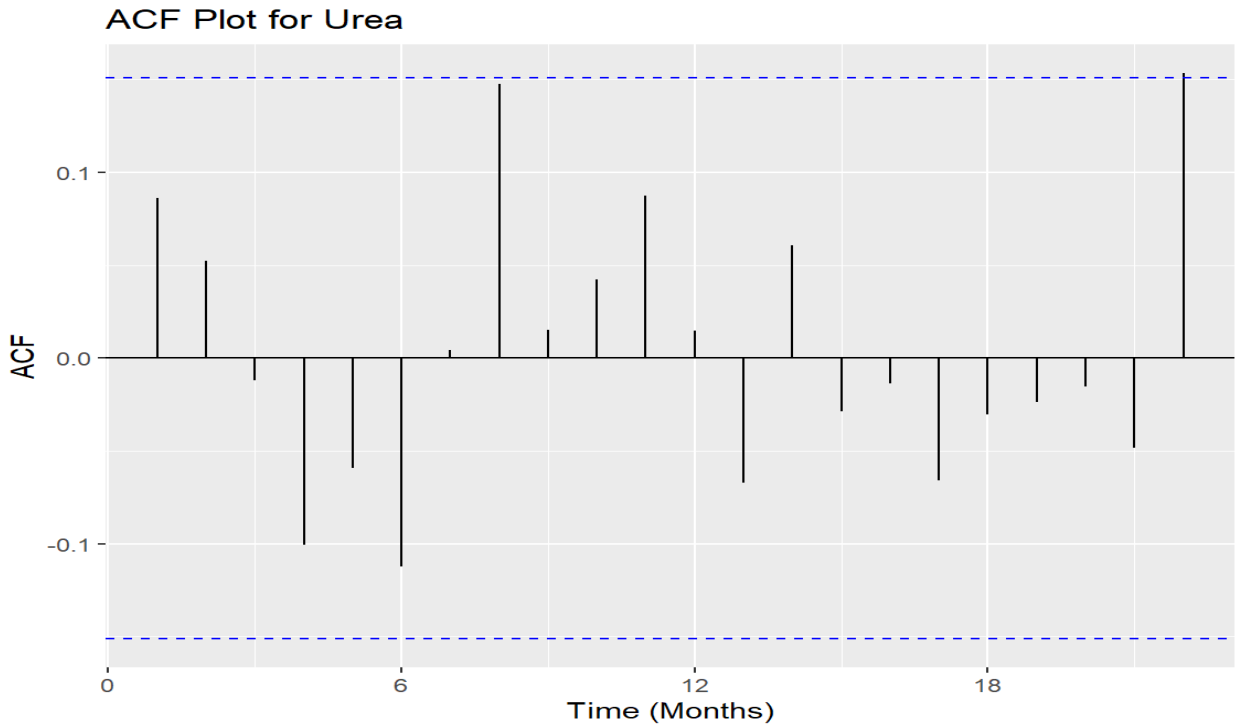


Figure 49: Autocorrelation function plot for urea fertiliser time series

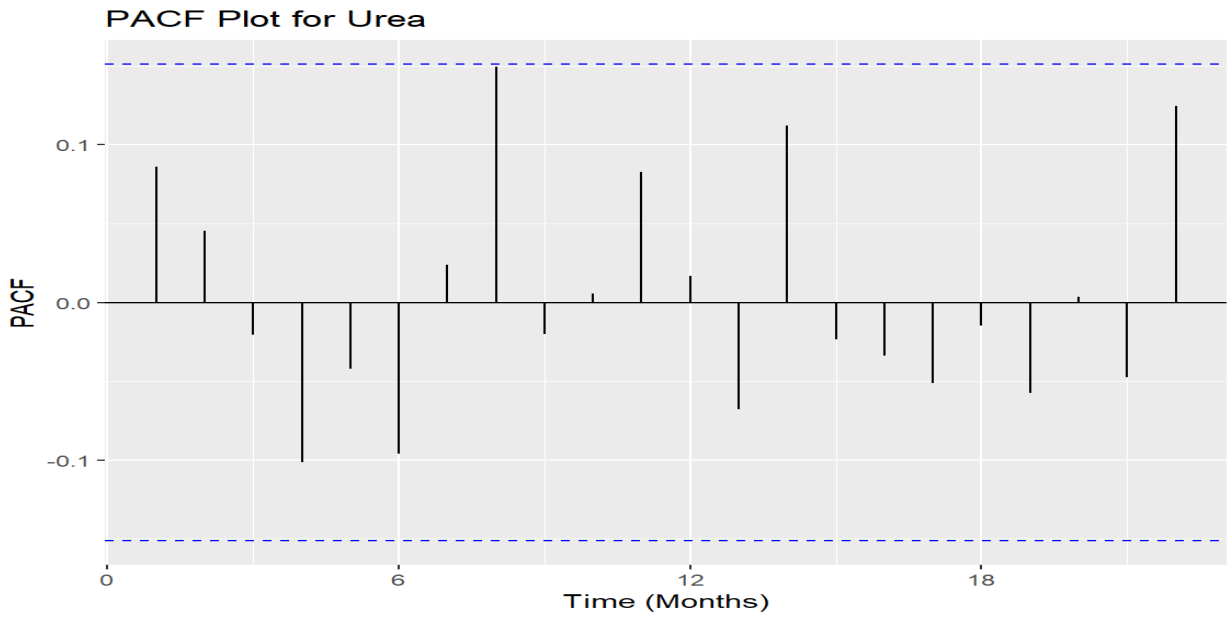


Figure 50: Partial autocorrelation function plot for urea fertiliser time series

The demand for urea showed no significant long-term trend, with only slight increases during March, April, and May. The lack of a strong trend indicates the flexible use of urea, which farmers use based on crop needs and availability rather than a fixed schedule. The slight increases in demand during certain months align with active crop

growth periods, particularly for maize and wheat, which rely heavily on nitrogen for optimal yields (Zheng *et al*., 2017). The characteristics of the shifts in demand for Urea suggest that its use is more closely tied to maintenance fertilization rather than initial planting. This stability might also reflect urea's role in top-dressing during maize cultivation, which is applied after the crops have been established.

4.2.6 Visualisation of Total Demand for Fertiliser in Kenya

The time series plot in Figure 51 represents the total fertiliser demand from 2010 to 2023. The plot reveals both a trend and seasonality of fertiliser demand. This plot indicates that total fertiliser demand has regular fluctuations in demand, with recurring peaks and valleys that suggest a seasonal pattern likely connected to agricultural cycles. These peaks typically occur annually, reflecting consistent seasonal demand for fertilisers, corresponding to planting or growing seasons. The overall trend does not exhibit a clear upward or downward trajectory, indicating that despite the seasonal variations, the average fertiliser demand has remained relatively steady over time. In addition, the plots indicate that there has been a noticeable increase in the intensity of the peaks in the last three years, suggesting that demand during peak seasons is high, which may be explained by shifts in agricultural practices, climate, or market conditions.

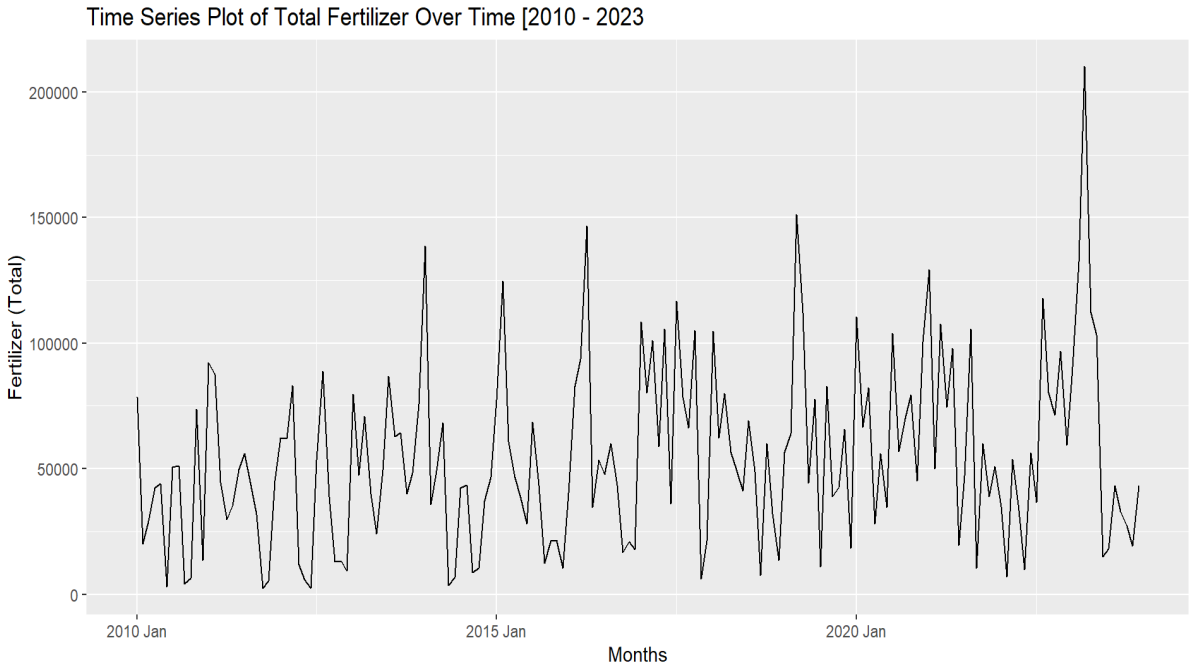


Figure 51: Time series plot for Total fertiliser demand

4.2.6.1 Testing Normality of the Total Fertiliser Time Series

In order to test the normality of the data, the Jarque-Bera test was conducted on the `total fertiliser` variable. The test produced a test statistic (X-squared) of 35.55 with 2 degrees of freedom and a p-value of 1.908e-08. These results indicate that the data does not follow a normal distribution, as the null hypothesis of normality is strongly rejected. This suggests that the distribution of the `total fertiliser` data exhibits significant skewness and/or kurtosis, deviating from the characteristics of a normal distribution.

The plot in Figure 52 is a time series graph with multiple-coloured lines representing total fertiliser usage from 2010 to 2023. Each colour corresponds to a different year (Appendix 4). The plot shows considerable variation in fertiliser usage across different months and years, with noticeable peaks around March and July, which could indicate seasonal effects or agricultural cycles influencing demand. In particular, some years, like 2019, exhibit much higher usage during peak months, possibly due to changes in agricultural practices or external factors like market demand or weather conditions.

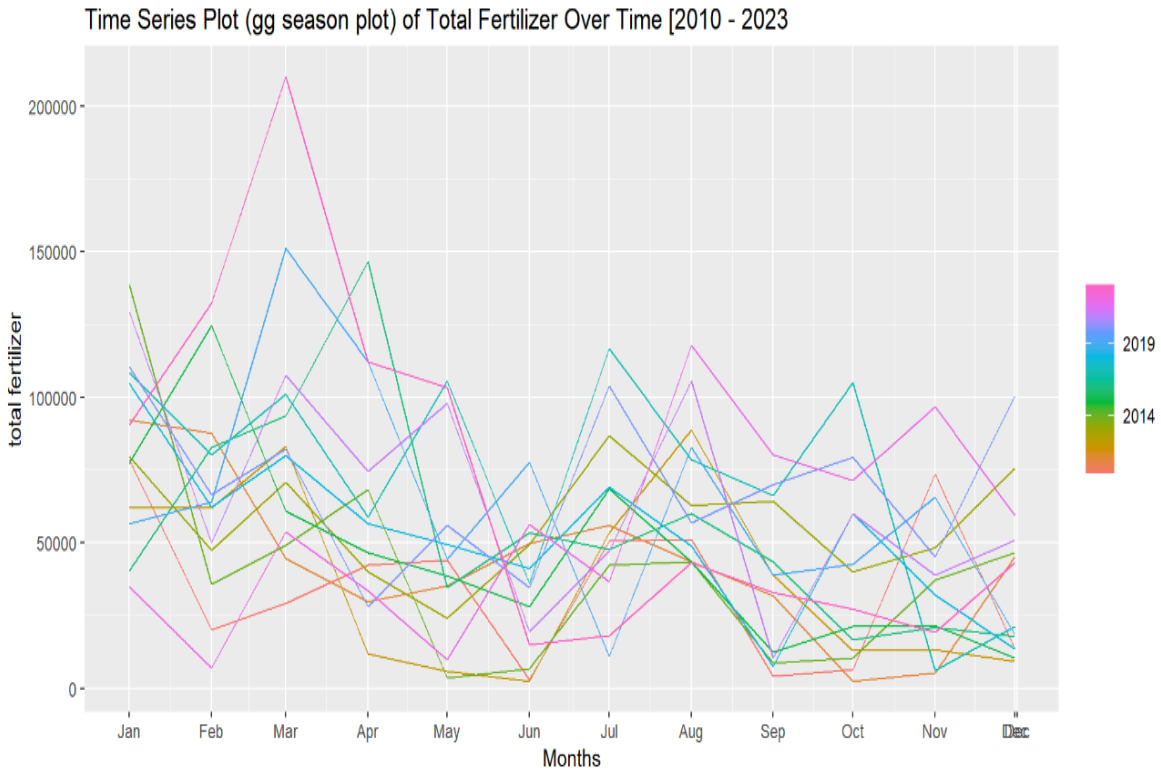


Figure 52: gg season plot for Total fertiliser time series

On the other hand, figure 53, a gg-sub-series plot, breaks down the data by month across different years into separate panels. This plot reveals more evident trends, indicating that March, July, and October generally experience higher fertiliser usage, aligning with the seasonal patterns observed in the first plot. The blue lines representing monthly mean values provide a comparison, showing that while there is variability from year to year, certain months consistently require more fertiliser. This consistency may be related to crop cycles, suggesting that more intensive fertilization is needed during specific times of the year.

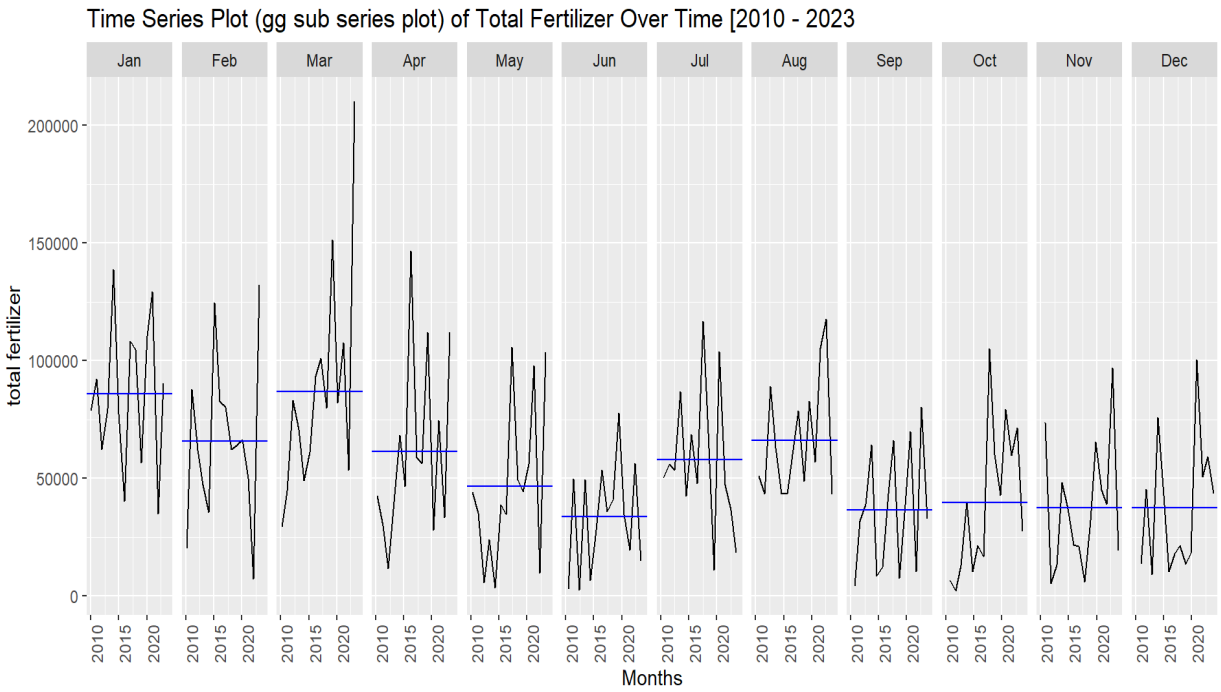


Figure 53: gg subseries plot for total fertiliser demand time series

4.2.6.2 Seasonal Decomposition of Total Fertiliser Demand

The seasonal subseries plot in Figure 54 shows the seasonal changes in total fertiliser demand from 2010 to 2023. The blue lines indicate the average monthly usage, while the black lines show year-to-year variations. From this plot, fertiliser usage tends to be high in January, February, March, and April. Total demand is low in May, June, Sep, October, Nov, and Dec. During July and August, total fertiliser demand is relatively high.

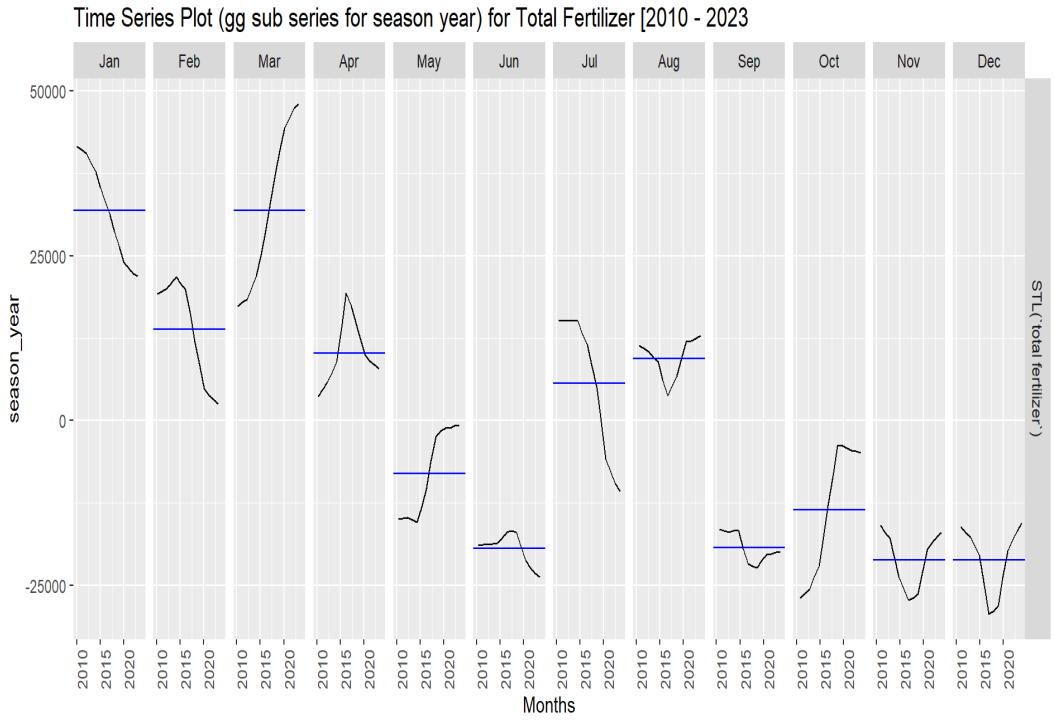


Figure 54: gg subseries plot for season year for total fertiliser demand

In Figure 55, the plot breaks down the data into the trend, seasonality, and residuals. The "trend" component illustrates the long-term changes in fertiliser usage, peaking around 2020 and then slightly declining. The "season year" component reveals consistent seasonal patterns each year, with regular peaks and troughs showing recurring changes in fertiliser demand throughout the year. The "remainder" component captures irregularities or noise, possibly due to unexpected events or outliers.

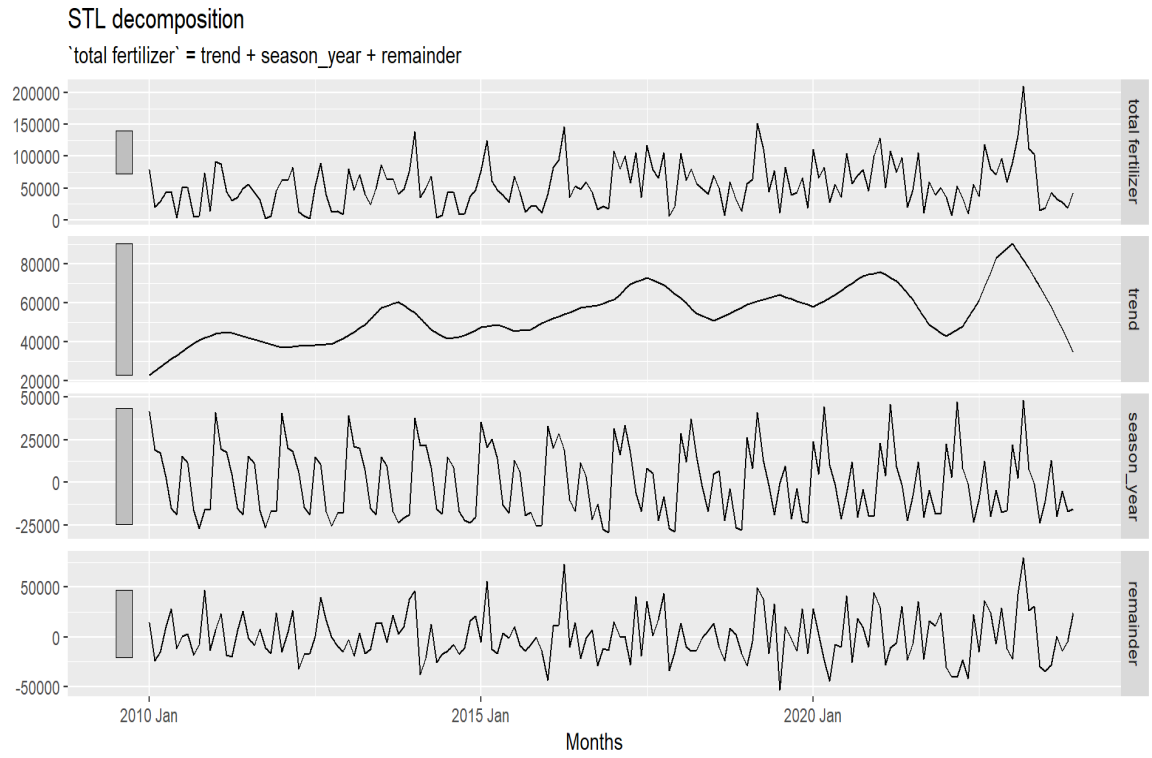


Figure 55: Seasonal-Trend Decomposition plot for total fertiliser demand

The plot in Figure 56 is a visualization that confirms the STL decomposition findings, showing that some months consistently have higher or lower fertiliser usage year after year—demand peaks around July and October and dips in March and September. The lines for different years are closely aligned, indicating a stable seasonal pattern in fertiliser application, though with some annual variation.

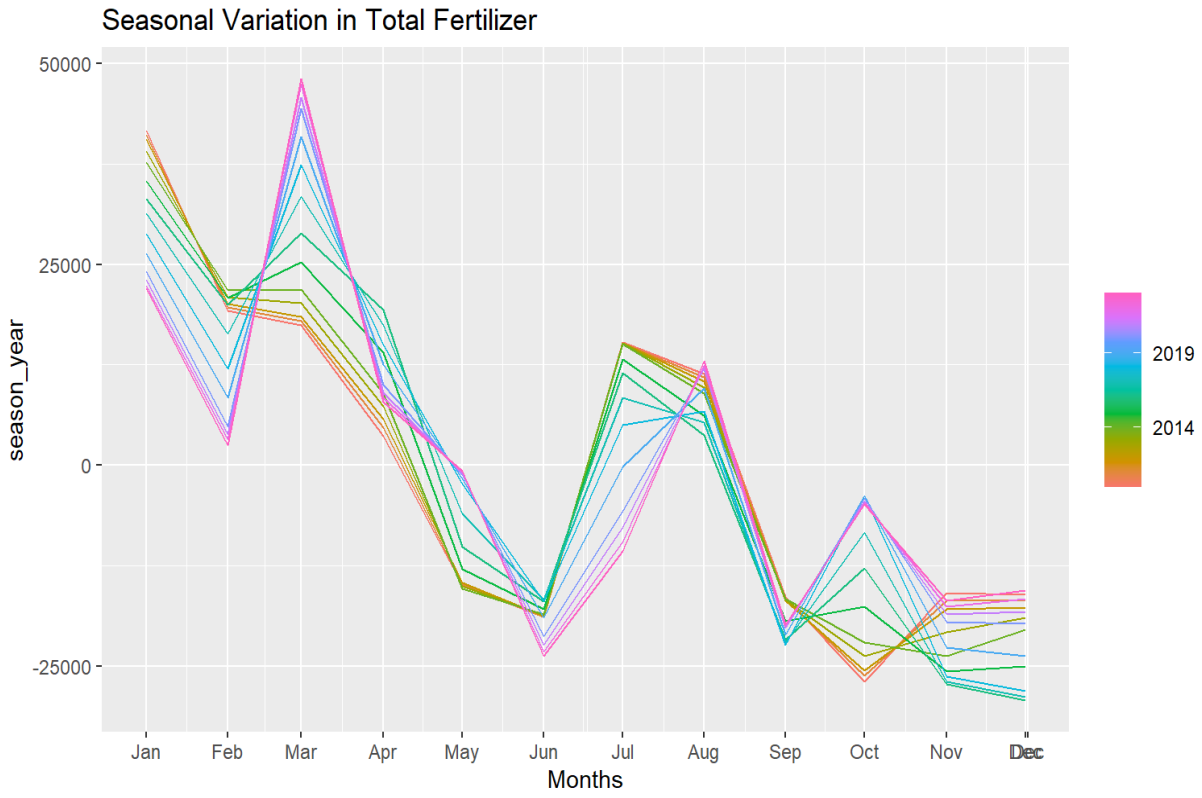


Figure 56: Plot showing seasonal changes in total fertilizer demand

4.2.6.3 Stationarity Test

The KPSS test was performed to check if the total fertiliser demand time series was stationary. This test determines whether a time series is stationary around a constant mean or trend. In this test, the null hypothesis is that the data is stationary, while the alternative hypothesis suggests that the data has a unit root, meaning it is not stationary. The test results showed a KPSS statistic of 0.623 and a p-value of 0.0205. Since the p-value is less than the standard significance level of 0.05, we reject the stationarity null hypothesis. Therefore, the data is not stationary and needs to be differenced to become stationary. A first difference was applied to achieve this, which involves subtracting each data point from the previous one to remove trends and stabilize the series.

The ACF plot for total fertiliser (Figure 57) shows the correlation between the time series and its previous values at various time lags. The plot reveals notable spikes at the first and second lags, indicating a strong correlation with values up to 2 months ago. From the plot, it is evident that as the lags increase, the correlations gradually diminish, with some falling outside the significance ACF bounds. This suggests that lags are not

statistically significant. Therefore, the time series has short-term dependencies but does not exhibit long-term autocorrelation.

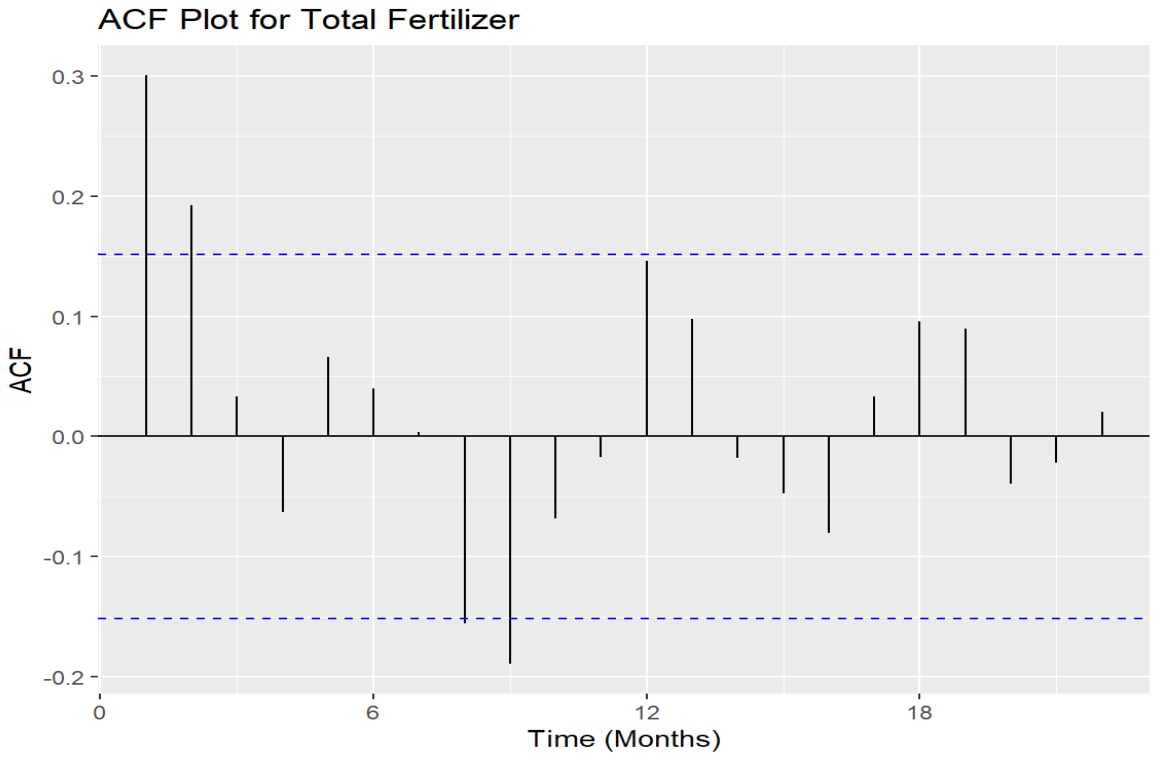


Figure 57: Autocorrelation function plot for total fertiliser demand

On the other hand, the Partial Autocorrelation Function (PACF) plot indicates a direct effect of each lag on the time series after accounting for the effects of any intervening lags. The plot displays a significant spike at the first lag and smaller spikes at a few other lags, especially at lag 2. This indicates that the first lag has a strong impact, with some additional influence from the second lag, once the effect of the first lag is removed. Beyond the second lag, the correlations are smaller and remain within the significance bounds, indicating that lags beyond the first or second have little effect. This pattern suggests an autoregressive process, potentially AR (1) or (2).

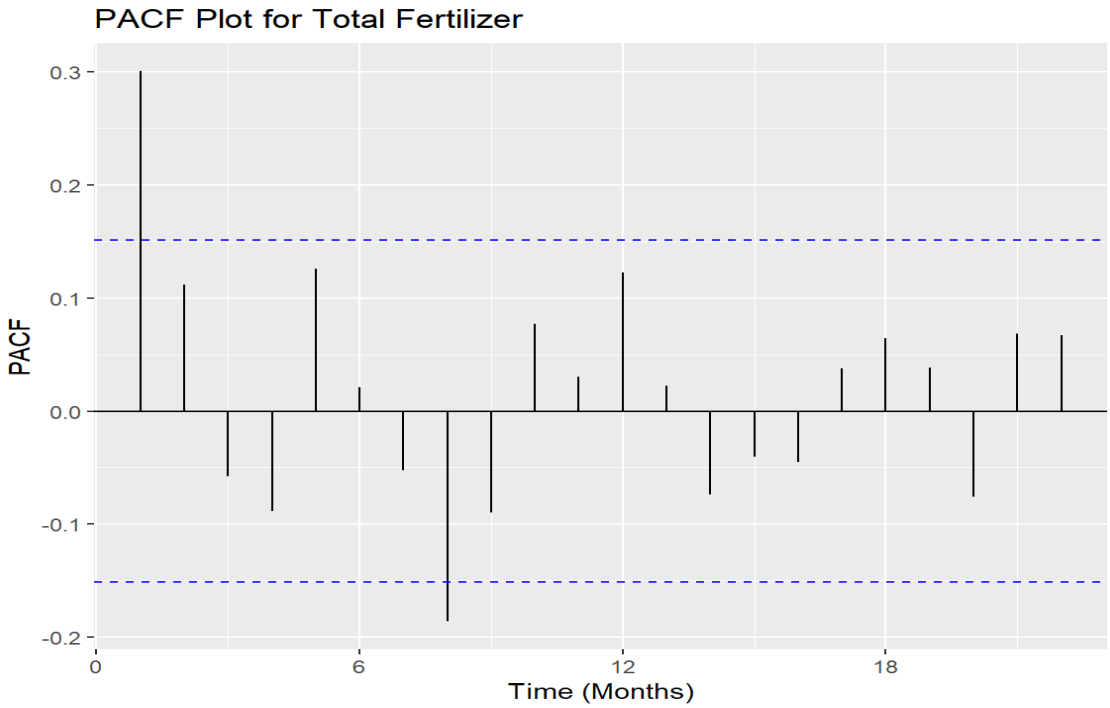


Figure 58: Partial autocorrelation function plot for total fertiliser demand

The ACF and PACF analyses of the total fertiliser time series indicate a short-term dependence on recent values. The ACF plot indicated a strong correlation at the first and second lags, suggesting that values influence the series from the past one to two months, while correlations at higher lags diminish and fall within significance bounds, indicating a lack of long-term autocorrelation. The PACF plot highlights a significant impact from the first lag and a smaller influence from the second, pointing to a potential autoregressive process, likely AR (1) or AR (2). These findings suggest that the series is primarily driven by recent trends, making it suitable for modelling using a low-order autoregressive process for effective short-term forecasting.

The visualisation of the fertiliser demand involved a thorough examination of seasonal and trend components, which are essential in determining the appropriate modelling approach for accurate forecasting. Drawing on established methodologies, such as those used in the studies by Muriithi *et al.* (2021) and Doulah (2018), the presence of seasonality and trends within the data was examined. Muriithi *et al.* (2021) highlights the suitability of SARIMA (Seasonal Autoregressive Integrated Moving Average) models for time series that exhibit a seasonal component. SARIMA models are specifically designed to capture both seasonal variations and non-seasonal trends,

making them a robust choice for such data (Out *et al*., 2014). To determine whether the time series in this study contained these components, seasonal decomposition using LOESS was used and each fertiliser showed seasonality and an increasing trend in demand.

The seasonal decomposition in this study revealed the presence of significant seasonal patterns and trends, confirming the need for a SARIMA model. This step is crucial, as the identification of these components not only informs the choice of model but also enhances the accuracy of the forecasts. The approach in this analysis aligns with the methodologies of previous studies, ensuring that the model selection process is both rigorous and evidence-based. By confirming the presence of seasonality and trends, the foundation was set for employing SARIMA models, which are particularly proficient at handling such complexities in the data.

The visualization of NPK, Urea, Calcium Nitrite, Calcium ammonium nitrate, DAP and Muriate of potash fertilizer demand reveals clear trends and seasonality. These findings are supported by several studies. A study conducted by Mathenge (2009) on "Fertilizer Types and Availability Across Regions" found that fertilizer demand varies by region and season, aligning with the seasonal fluctuations observed in this study. Similarly, Jayne *et al*. (2003) noted that maize production significantly drives fertilizer demand, with peaks during planting seasons, reflecting the seasonal patterns at different months of the year. As indicated by a study by Mutegi *et al* (2024), the type of crop to be planted is determined by season; whether it's a short rain or a long rain. This explains the reason why the visualisation results pointed high fertiliser demand during January to April. The findings of these results are also confirmed by the report by Chege and Meshack (2024). Chege and Meshack (2024) pointed out that fluctuations in fertilizer imports driven by global supply chain disruptions and seasonal local demands. Wanzala *et al*. (2013) also identified distinct seasonal patterns in smallholder farmers' fertilizer use, closely tied to crop cycles, confirming that certain months indeed show higher demand, indicated by data visualisation. These studies confirm the findings of this study that demand for the fertiliser in Kenya is seasonal and various factors have contributed to the seasonality of demand; such as logistic issues, season of the year, prices and government policies.

4.3 Developing Predictive Models

This section presents the data analysis to help determine the most suitable SARIMA model for each fertiliser type. After determining the stationarity of the data, the AR and MA process was ordered by examining ACF and PACF plots, as indicated in the last sections of stationarity. After differing in attaining stationarity and examining the ACF and PACF, it was clear that each type of fertiliser processed seasonality. An auto Sarima model was generated using the R program, and the SARIMA models generated for each fertiliser type are below.

4.3.1 Calcium ammonium nitrate (CAN); Seasonal Autoregressive Integrated Moving Average (0,0,0) w/ mean

The analysis of the Calcium Ammonium Nitrate (CAN) data pointed out that the demand can be explained by the SARIMA (0,0,0) model with a mean. This implies that the model mainly reflects the average level of the data and does not include any autoregressive, moving average, or seasonal components. The constant term, estimated at 8853.002, is highly significant with a p-value of <0.0001, indicating it accurately represents the average series value (Table 6). The variance of the residuals is 154,571,384, shedding light on the model's error variability. The model's log-likelihood is -1821.8, and the Akaike Information Criterion (AIC) is 3647.6, with the corrected AIC (AICc) at 3647.67 and the Bayesian Information Criterion (BIC) at 3653.84. These results suggest that despite its simplicity, the model provides a good fit for the data by focusing solely on its mean.

Table 6: Calcium Ammonium Nitrate Seasonal Autoregressive Integrated Moving Average Model

Model	Term	Estimate	Std. Error	t-statistic	p-value
ARIMA	constant	8853.002	956.3476	9.257096	<0.0001

sigma² estimated as 154571384: log likelihood = -1821.8, AIC = 3647.6 AICc = 3647.67 BIC = 3653.84; This Model order is autogenerated from the auto. Arima () function in R based on the minimization of AIC and BIC

From the results in Table 6, the general time series equation that can predict the demand for CAN fertiliser will be

$$Y_t = 8853.002 \quad (42)$$

The model has only the constant term, where Y_t is the CAN demand

The SARIMA (0,0,0) w/mean captures only the average level of demand over time. This model lacks autoregressive, moving average, or seasonal components, suggesting that the demand for CAN is relatively stable with no significant fluctuations or patterns. The highly significant constant term (8853.002, p-value = <0.0001) indicates the average series value, implying that the demand for CAN is consistent around its mean. The residual variance (154,571,384) highlights the error variability, but despite this, the model's log-likelihood (-1821.8) and information criteria (AIC = 3647.6, BIC = 3653.84) indicate a decent fit given the data's simplicity. The analysis, therefore, indicates that the future demand for Calcium ammonium nitrate can be predicted by its mean value only.

4.3.2 Calcium Nitrate; Seasonal Autoregressive Integrated Moving Average (1,1,1) (0,0,1) [12]

This model consists of one non-seasonal autoregressive term (AR1) estimated at 0.102946, one non-seasonal moving average term (MA1) estimated at -0.96153, and one seasonal moving average term (SMA1) estimated at 0.137274 (Table 7). The AR1 term is not statistically significant (p-value = 0.209), while the MA1 term is highly significant (p-value = <0.0001). The SMA1 term is also not statistically significant (p-value = 0.228). The residual variance is estimated at 6,118,068. The model's log-likelihood is -1541.51, with an Akaike Information Criterion (AIC) of 3091.02, a corrected AIC (AICc) of 3091.26, and a Bayesian Information Criterion (BIC) of 3103.49. These findings suggest that the MA1 term is crucial to the model, whereas the AR1 and SMA1 terms do not contribute significantly.

Table 7: Calcium nitrate Seasonal Autoregressive Integrated Moving Average model

Model	Term	Estimate	Std. error	t-statistic	p-value
ARIMA	ar1	0.102946	0.08163	1.26114	2.09E-01
ARIMA	ma1	-0.96153	0.024398	-39.4107	1.73E-86
ARIMA	sma1	0.137274	0.11341	1.210421	2.28E-01

sigma^2 estimated as 6118068: log likelihood=-1541.51, AIC=3091.02 AICc=3091.26 BIC=3103.49: This Model order is autogenerated from the auto. Arima () function in R based on the minimization of AIC and BIC

From the table 7, it is clear that the model that represents the demand for calcium nitrate will be

$$Y_t = 0.102946y_{t-1} + (-0.96153)e_{t-1} + 0.137274\,\epsilon_{t-1} \quad (43)$$

where std. error; Y_t Is the calcium nitrate series, y_{t-1} is the non-seasonal AR process of order 1, e_{t-1} is the non-seasonal MA process of order 1, and ϵ_{t-1} is the seasonal MA process of order 1.

The SARIMA (1,1,1) (0,0,1) [12] model for Calcium Nitrate includes one non-seasonal autoregressive term (AR1), one non-seasonal moving average term (MA1), and one seasonal moving average term (SMA1). The MA1 term is highly significant (p-value = 1.73E-86), suggesting it is crucial in capturing the short-term dependencies in the series. However, the AR1 and SMA1 terms are not statistically significant, indicating that they do not contribute much to explaining the time series. The model's residual variance (6,118,068) and log-likelihood (-1541.51) values suggest moderate error variability, with an AIC of 3091.02 and BIC of 3103.49 indicating a good fit. Therefore

$Y_t = 0.102946y_{t-1} + (-0.96153)e_{t-1} + 0.137274\,\epsilon_{t-1}$

Indicate that the demand model is primarily explained by the MA1 term while acknowledging the limited influence of AR1 and SMA1.

4.3.3 Diammonium Phosphate; Seasonal Autoregressive Integrated Moving A verage (0,0,0) (2,0,0) [12] w/ mean

This model includes two seasonal autoregressive terms: SAR1, estimated at 0.309194, and SAR2, estimated at 0.205886. Both terms are statistically significant, with SAR1 having a p-value of 9.74E-05 and SAR2 having a p-value of 9.75E-03 (Table 8). The constant term is estimated at 8432.586 and is highly significant, with a p-value of 7.22E-09. The variance of the residuals is estimated at 395,584,183. The model's log-likelihood is -1901.23, with an Akaike Information Criterion (AIC) of 3810.46, a corrected AIC (AICc) of 3810.7, and a Bayesian Information Criterion (BIC) of 3822.95. These results show that the seasonal autoregressive and constant terms play a significant role in the model. In addition, the model includes a mean term, which implies that the model accounts for a constant average level in the time series.

Table 8: Diammonium phosphate Seasonal Autoregressive Integrated Moving Average model

Model	Term	Estimate	Std. error	t-statistic	p-value
ARIMA	sar1	0.309194	7.74E-02	3.992806	9.74E-05
ARIMA	sar2	0.205886	7.88E-02	2.614315	9.75E-03
ARIMA	constant	8432.586	1.38E+03	6.09617	7.22E-09

sigma^2 estimated as 395584183: log likelihood=-1901.23, AIC=3810.46 AICc=3810.7 BIC=3822.95: This Model order is autogenerated from the auto. Arima () function in R based on the minimization of AIC and BIC

$$Y_t = 8432.586 + 0.309194y_{t-12} + 0.205886y_{t-24} \quad (44)$$

where y_{t-12} and y_{t-24} represents the value of the time series 12 periods (months) and the value of the time series 24 months before the current time t.

The SARIMA (0,0,0) (2,0,0) [12] model with a mean for Diammonium Phosphate incorporates two significant seasonal autoregressive terms (SAR1 and SAR2) and a constant term, which indicates a stable average level in the time series. The significance of SAR1 (p-value = 9.74E-05) and SAR2 (p-value = 9.75E-03) suggests that the demand for Diammonium Phosphate is influenced by patterns that recur yearly, with lagged effects from 12 and 24 months prior playing a critical role. The constant term (8432.586, p-value = 7.22E-09) supports the steady demand level, while the model's residual variance (395,584,183) and log-likelihood (-1901.23) implies a relatively higher error variability compared to the previous models.

4.3.4 Muriate of Potash; Seasonal Autoregressive Integrated Moving Average (1,1,4) (0,0,1) [12]

This model consists of one non-seasonal autoregressive term (AR1) estimated at -0.09014, four non-seasonal moving average terms (MA1 estimated at -0.6831, MA2 estimated at 0.080796, MA3 estimated at 0.049645, and MA4 estimated at -0.39939), and one seasonal moving average term (SMA1) estimated at -0.04284 (Table 9). The AR1 term is not statistically significant (p-value = 0.700), nor are MA2 (p-value = 0.695), MA3 (p-value = 0.646), and SMA1 (p-value = 0.757). However, the MA1 term is significant (p-value = 0.00209), and the MA4 term is highly significant (p-value = 6.34E-05). The residual variance is estimated at 4,300,024. The model's log-likelihood is -1510.62, with an Akaike Information Criterion (AIC) of 3035.23, a corrected AIC

(AICc) of 3035.94, and a Bayesian Information Criterion (BIC) of 3057.06. These results suggest that the MA1 and MA4 terms are essential to the model, while the other terms do not significantly impact.

Table 9: Muriate of Potash Seasonal Autoregressive Integrated Moving Average model

Model	Term	Estimate	Std. error	t-statistic	p-value
ARIMA	ar1	-0.09014	0.23343	-0.38617	7.00E-01
ARIMA	ma1	-0.6831	0.218518	-3.12604	2.09E-03
ARIMA	ma2	0.080796	0.206044	0.392128	6.95E-01
ARIMA	ma3	0.049645	0.107811	0.460481	6.46E-01
ARIMA	ma4	-0.39939	0.097318	-4.10394	6.34E-05
ARIMA	sma1	-0.04284	0.13814	-0.31011	7.57E-01

sigma^2 estimated as 4300024: log likelihood=-1510.62, AIC=3035.23 AICc=3035.94 BIC=3057.06: This Model order is autogenerated from the auto. Arima () function in R based on the minimization of AIC and BIC

$$Y_t = -0.09014y_{t-1} - 0.6831e_{t-1} + 0.080796e_{t-2} + 0.049645e_{t-3} - 0.39939e_{t-4} - 0.04284e_{t-12} \quad (45)$$

where Y_t is the series at time t, Y_{t-1} is the lagged value of the time series, e_{t-i} ; i=1,2,3, 4 & 12 represents lagged errors in MA and Seasonal components.

The SARIMA (1,1,4) (0,0,1) [12] model for Muriate of Potash incorporates one non-seasonal autoregressive term (AR1), four non-seasonal moving average terms (MA1, MA2, MA3, MA4), and one seasonal moving average term (SMA1). The significance of the MA1 (p-value = 0.00209) and MA4 (p-value = 6.34E-05) terms indicates that these components are crucial in capturing the demand's short-term fluctuations. However, the AR1, MA2, MA3, and SMA1 terms are not statistically significant. These findings imply that including these does not improve the model significantly. The residual variance (4,300,024) and log-likelihood (-1510.62) are relatively low, reflecting moderate error variability.

4.3.5 Nitrogen, Phosphorus, and Potassium Fertiliser Seasonal Autoregressive Integrated Moving Average model: (2,0,0) (2,0,0) [12] w/ mean

The Auto SARIMA showed that the demand for NPK has two non-seasonal autoregressive terms (AR1 at 0.045363 and AR2 at 0.177016) and two seasonal

autoregressive terms (SAR1 at 0.259051 and SAR2 at 0.39847) (Table 10). The AR1 term is not statistically significant (p-value = 0.562), while the AR2 term is significant (p-value = 0.0299). Both SAR1 (p-value = 0.000712) and SAR2 (p-value = 4.36E-06) are highly significant. The constant term is estimated at 3219.816 and highly significant (p-value = 0.000271). The residual variance is estimated at 187,363,368. The model's log-likelihood is -1839.26, with an Akaike Information Criterion (AIC) of 3690.52, a corrected AIC (AICc) of 3691.05, and a Bayesian Information Criterion (BIC) of 3709.27. These results show that the AR2, SAR1, and SAR2 terms and the constant term significantly contribute to the model, while the AR1 term does not significantly impact the model. Also, the model includes a mean term, meaning it accounts for a constant average level in the time series.

Table 10: Nitrogen, Phosphorus, and Potassium fertiliser Seasonal Autoregressive Integrated Moving Average model coefficients table

Model	Term	Estimate	Std. error	t-statistic	p-value
ARIMA	ar1	0.045363	0.078015	0.581459	5.62E-01
ARIMA	ar2	0.177016	0.080845	2.189573	2.99E-02
ARIMA	sar1	0.259051	0.075117	3.448628	7.12E-04
ARIMA	sar2	0.39847	0.083908	4.748894	4.36E-06
ARIMA	constant	3219.816	865.3579	3.720792	2.71E-04

sigma^2 estimated as 187363368: log likelihood=-1839.26, AIC=3690.52, AICc=3691.05M, BIC=3709.27: This Model order is autogenerated from the auto. Arima () function in R based on the minimization of AIC and BIC

From the coefficients, the following SARIMA model represents the demand for NPK fertiliser demand.

$$Y_t = 3219.816 + 0.045363Y_{t-1} + 0.177016Y_{t-2} + 0.259051Y_{t-12} + 0.39847Y_{t-24} \quad (46)$$

where Y_t is the series at time t, Y_{t-1} and Y_{t-2} represent the lagged values of the non-seasonal series and Y_{t-12} and Y_{t-24} represent the lagged seasonal values of the series. 3219.816 represents the constant part of the series.

4.3.6 Urea; Autoregressive Integrated Moving Average (0,1,1)

This model has only one non-seasonal moving average term (MA1) with an estimate of -0.97465, which is highly significant (p-value = <0.0001) (Table 11). The variance of the residuals is estimated at 83,726,476. The model's log-likelihood is -1761.25, with

an Akaike Information Criterion (AIC) of 3526.51, a corrected AIC (AICc) of 3526.58, and a Bayesian Information Criterion (BIC) of 3532.74. These findings show that the MA1 term is a critical component of the model and significantly improves its accuracy.

Table 11:Urea; Autoregressive Integrated Moving Average model

model	term	estimate	std. error	t-statistic	p. Value
ARIMA	ma1	-0.97465	0.017738	-54.9458	<0.0001

sigma² estimated as 83726476: log-likelihood = -1761.25, AIC = 3526.51, AICc = 3526.58, BIC = 3532.74: This Model order is autogenerated from the auto. Arima () function in R based on the minimization of AIC and BIC

$$Y_t = Y_{t-1} + 0.97465e_{t-1} \quad (47)$$

where Y_{t-1} represent the series at time t-1, Y_t is the series at time t, is the MA part at time t-1

The model indicates that the value Y_t, of Urea demand is closely related to the previous value, adjusted by the past error. This means the model heavily relies on how well it predicted in the past to make current predictions. The ARIMA (0,1,1) model for Urea is simple but effectively captures the short-term fluctuations in demand by considering the errors made in previous predictions. The significance of the MA1 term means that this model is good at accounting for short-term dependencies in the data.

4.3.7 Total Fertiliser Seasonal Autoregressive Integrated Moving Average model: (1,1,4) (0,0,1) [12]

This model features one non-seasonal autoregressive term (AR1) with an estimate of -0.57506, and four non-seasonal moving average terms (MA1 at -0.13842, MA2 at -0.42886, MA3 at -0.15199, and MA4 at -0.23263). Additionally, it includes one seasonal moving average term (SMA1) with an estimate of 0.130772. The AR1 term is not statistically significant (p-value = 0.162), nor are MA1 (p-value = 0.728) and MA2 (p-value = 0.172). However, MA3 (p-value = 0.055) and MA4 (p-value = 0.015) are significant, and SMA1 is nearly significant (p-value = 0.085). The variance of the residuals is estimated at 1.149e+09. The model's log-likelihood is -1977.17, with an Akaike Information Criterion (AIC) of 3968.33, a corrected AIC (AICc) of 3969.04, and a Bayesian Information Criterion (BIC) of 3990.16. These findings suggest that the

MA4 term, and possibly MA3 and SMA1 terms, significantly contribute to the model, while the others do not.

Table 12: Total fertiliser Seasonal Autoregressive Integrated Moving Average model

Model	Term	Estimate	Std. error	t-statistic	p-value
ARIMA	ar1	-0.57506	0.409179	-1.40541	0.161758
ARIMA	ma1	-0.13842	0.396684	-0.34895	0.727563
ARIMA	ma2	-0.42886	0.312842	-1.37085	0.172261
ARIMA	ma3	-0.15199	0.078585	-1.93411	0.05479
ARIMA	ma4	-0.23263	0.094572	-2.45984	0.014919
ARIMA	sma1	0.130772	0.075481	1.732516	0.085028

sigma^2 estimated as 1.149e+09: log likelihood=-1977.17, AIC=3968.33 AICc=3969.04 BIC=3990.16: This Model order is autogenerated from the auto. Arima () function in R based on the minimization of AIC and BIC

The SARIMA model that represents the fertiliser demand is, therefore

$$Y_t = (1 + 0.57506)Y_{t-1} - 0.57506Y_{t-2} + 0.13842e_{t-1} + 0.42886e_{t-2} + 0.15199e_{t-3} + 0.23263e_{t-4} - 0.130772\mathcal{E}_{t-12} \quad (48)$$

where Y_t is the series at time t, Y_{t-1} and Y_{t-2} are the lagged values of the series, e_{t-i} ; i=1,2,3,4, and 12 represent the lagged terms error terms (MA and seasonal component)

The SARIMA (1,1,4) (0,0,1) [12] model for Total Fertiliser demand is a complex model that includes one non-seasonal autoregressive term (AR1), four non-seasonal moving average terms (MA1, MA2, MA3, MA4), and one seasonal moving average term (SMA1). The significance of MA4 (p-value = 0.015) and near-significance of MA3 (p-value = 0.055) and SMA1 (p-value = 0.085) indicate that these terms are significant in capturing the demand's short-term fluctuations and seasonal effects. However, AR1, MA1, and MA2 are not statistically significant, which implies that their inclusion has an insignificant effect on the model.

As indicated by Shumway and Stoffer (2017), the process of model fitting involves several key steps: data visualization, data transformation where necessary, model identification of dependence order, parameter estimation, diagnostic analysis, and the selection of an appropriate model. The principle of parsimony necessitates researchers to choose the simplest model that adequately explains the behaviour of the series (Chen *et al*., 2015). This study used a network search approach, which is common in the

reviewed literature. The model order was autogenerated from the '**auto.arima()'** function in R, which selects the model that minimizes AIC and BIC, with the model having the lowest values of these criteria considered the best (Akaike, 1974; Akaike, 1979). The efficiency of the model generated from this function is attributed to its ability to identify the best model from a series of potential models by minimizing these evaluation criteria (Hyndman & Athanasopoulos, 2018).

According to Box and Jenkins (1976), the Box-Jenkins approach recommends a minimum of 50 data points to ensure reliable and accurate model estimation. The models generated for each type of fertilizer are based on 168 data points, covering the period from January 2010 to December 2023, which corresponds to 168 months. Given that this study utilized over three times that amount, the dataset is more than adequate for the application of SARIMA and ARIMA models. Pisuttinusart's (2022) analysis entailed 168 months to model SARIMA models that forecasted fertiliser imports in Thailand. This data set is equal to this study confirming the applicability of SARIMA modelling in this study. The extensive dataset allows for capturing long-term trends and seasonal patterns with greater precision, thereby enhancing the reliability and accuracy of the models generated. The robustness of the modelling approach is further supported by the principle of parsimony, ensuring that the models selected are both simple and effective in explaining the behaviour of the time series data (Chen *et al*., 2015).

The analysis determined the optimal models for each type of fertilizer as follows: for Calcium Nitrate, SARIMA (1,1,1) (0,0,1) [12]; for Diammonium Phosphate, SARIMA (0,0,0) (2,0,0) [12]; for Muriate of Potash, SARIMA (1,1,4) (0,0,1) [12]; for Nitrogen, Phosphorus, and Potassium Fertiliser, SARIMA (2,0,0) (2,0,0) [12]; for Calcium Ammonium Nitrate, ARIMA (0,0,0) with mean; for Urea, ARIMA (0,1,1); and for Total Fertiliser Demand, SARIMA (1,1,4) (0,0,1) [12]. These models were chosen in accordance with the principle of parsimony, which advocates for selecting the simplest model that sufficiently explains the data (Chen *et al*., 2015). Due to the nature of the data, some time series exhibited non-seasonal patterns, leading to the application of ARIMA models where SARIMA was not appropriate. ARIMA models are suited for time series that do not display seasonality, as they focus on capturing non-seasonal

trends and patterns in the data (Hyndman & Athanasopoulos, 2018). Therefore, CAN and Urea demand are explained by ARIMA model.

4.4 Models' Accuracy Test

In order to evaluate the accuracy of SARIMA models, several necessary steps have been taken. First, the residuals are analysed to ensure they are randomly distributed and resemble white noise, which suggests the model has effectively captured the underlying patterns. This includes plotting the residuals, checking their autocorrelation using ACF and PACF functions, and assessing their normality with statistical tests and Q-Q plots. The model's forecasting accuracy is assessed by comparing its predictions to a hold-out sample or using cross-validation. Lastly, comparing forecast plots to actual values provides additional insight into the model's performance.

4.4.1 Forecasted Data Demand and Forecast Plots

Using the model generated from R, the following forecasts are made. From the previous ACF and PACF plots, it is clear that there are short-term dependencies. A 24-month forecast is therefore selected to predict the demand for each type of fertiliser. This decision is supported because the SARIMA model captures significant correlations primarily over shorter time lags. Forecasting over a longer horizon, such as 60 months, may lead to less accurate predictions because the model's effectiveness diminishes as it attempts to project further into the future, where the influence of short-term patterns becomes less reliable. Longer-term forecasts often suffer from increased uncertainty and reduced precision as they extrapolate the model's short-term behaviour into periods where underlying patterns and external factors may shift, potentially leading to misleading or inaccurate predictions.

4.4.1.1 Calcium Ammonium Nitrate Forecast

This plot suggests that the series can be accurately described solely by a constant term, with no significant trends or seasonal patterns affecting the data. The constant value of 8853.002 implies that the demand for CAN fertiliser remains stable over time, and the model's predictions do not vary from this constant level. This simple forecast reflects that, according to the model, there are no substantial fluctuations or changes in the demand for CAN fertiliser over the forecast period.

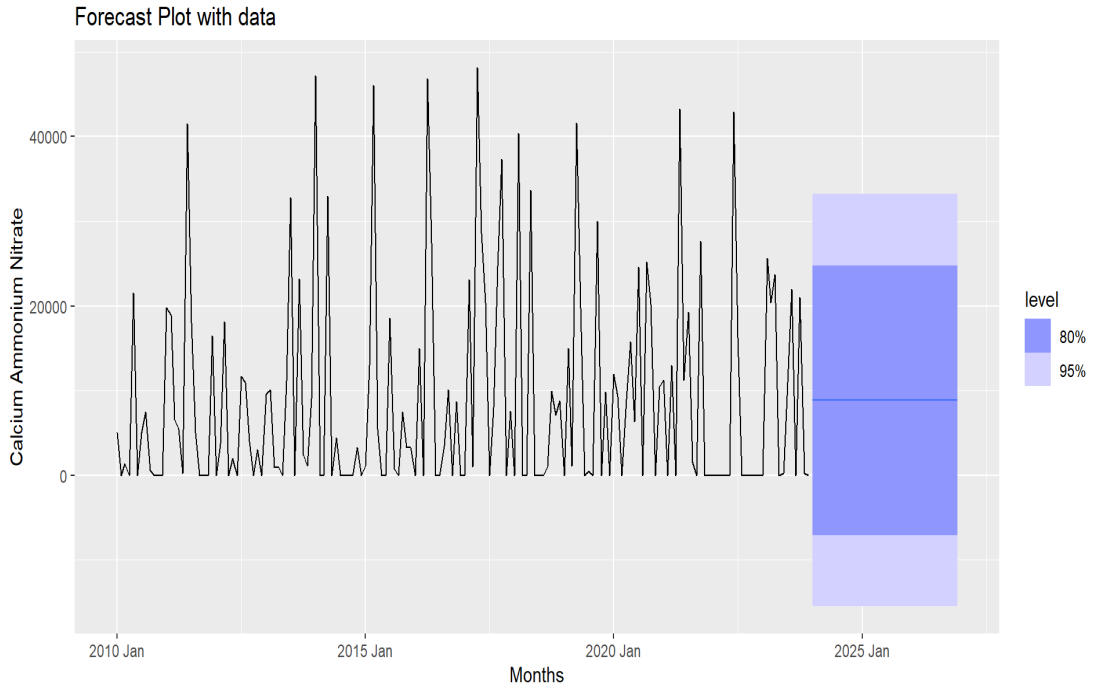


Figure 59: Forecast plot for calcium ammonium nitrate

Figure 59 is a plot of the forecasted demand for CAN fertiliser. The line within the blue-shaded area in the forecast section of the graph represents the point forecast, which is the best estimate of the future values of CAN based on the forecasting model. This solid blue line indicates the predicted values at each future time point. The blue-shaded area represents the range of uncertainty around this forecast. From the graph, all the forecasted points lie within the bounds, confirming that the predictions are within the acceptable range. Table 13 presents the forecast for CAN demand from August 2024 to July 2026, covering 24 months.

Table 13: Forecasted demand for calcium ammonium nitrate

Month	Qty (in metric tons)	Month	Qty (in metric tons)
2024 Aug	N (8853, 1.5e+08)	2025 Aug	N (8853, 1.5e+08)
2024 Sep	N (8853, 1.5e+08)	2025 Sep	N (8853, 1.5e+08)
2024 Oct	N (8853, 1.5e+08)	2025 Oct	N (8853, 1.5e+08)
2024 Nov	N (8853, 1.5e+08)	2025 Nov	N (8853, 1.5e+08)
2024 Dec	N (8853, 1.5e+08)	2025 Dec	N (8853, 1.5e+08)
2025 Jan	N (8853, 1.5e+08)	2026 Jan	N (8853, 1.5e+08)
2025 Feb	N (8853, 1.5e+08)	2026 Feb	N (8853, 1.5e+08)
2025 Mar	N (8853, 1.5e+08)	2026 Mar	N (8853, 1.5e+08)
2025 Apr	N (8853, 1.5e+08)	2026 Apr	N (8853, 1.5e+08)
2025 May	N (8853, 1.5e+08)	2026 May	N (8853, 1.5e+08)
2025 Jun	N (8853, 1.5e+08)	2026 Jun	N (8853, 1.5e+08)
2025 Jul	N (8853, 1.5e+08)	2026 Jul	N (8853, 1.5e+08)

4.4.1.2 Calcium Nitrate Forecast

The graph in figure 60 illustrates the actual and fitted values of Calcium Nitrate demand from January 2010 to December 2023. The red line represents the actual data, while the blue line represents the fitted model. While the lines closely follow each other throughout most of the period, slight deviations exist between the fitted values and the actual data, particularly during peaks. However, these deviations are not substantial, suggesting that the fitted model effectively captures the overall trend and variability in the data. This close alignment of the trend pattern confirms that the model provides a reliable prediction of Calcium Nitrate demand, with only minor discrepancies in more volatile periods. The model's ability to follow the data closely throughout most of the time series supports its reliability for forecasting future demand. Table 14 on the other hand represent the demand forecast for calcium nitrate fertiliser (in metric tonnes).

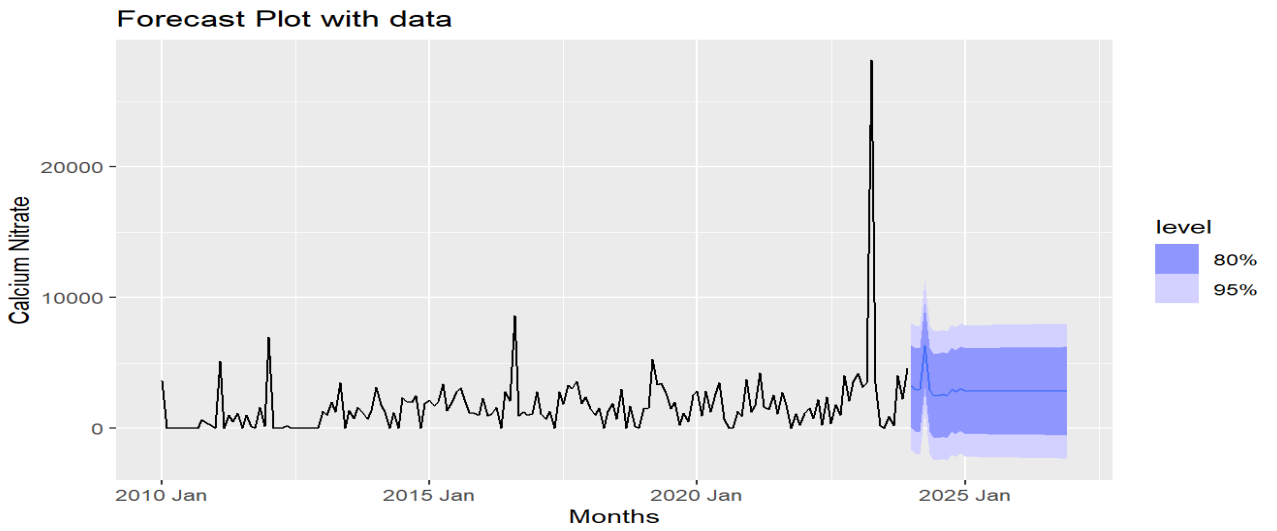


Figure 60: Forecast plot for calcium nitrate fertiliser

Table 14: Demand forecast points for calcium nitrate fertiliser

Month	Qty (in metric tons)	Month	Qty (in metric tons)
2024 Aug	N (2605, 6314507)	2025 Aug	N (2852, 6670099)
2024 Sep	N (2521, 6325759)	2025 Sep	N (2852, 6684653)
2024 Oct	N (2985, 6337011)	2025 Oct	N (2852, 6699207)
2024 Nov	N (2782, 6348264)	2025 Nov	N (2852, 6713760)
2024 Dec	N (3073, 6359516)	2025 Dec	N (2852, 6728314)
2025 Jan	N (2875, 6558094)	2026 Jan	N (2852, 6742868)
2025 Feb	N (2854, 6581839)	2026 Feb	N (2852, 6757421)
2025 Mar	N (2852, 6597236)	2026 Mar	N (2852, 6771975)
2025 Apr	N (2852, 6611875)	2026 Apr	N (2852, 6786529)

2025 May	N (2852, 6626437)	2026 May	N (2852, 6801082)
2025 Jun	N (2852, 6640992)	2026 Jun	N (2852, 6815636)
2025 Jul	N (2852, 6655546)	2026 Jul	N (2852, 6830190)

4.4.1.3 Diammonium Phosphate Demand Forecast

Figure 61 presents the forecasted demand for DAP fertiliser, with Table 15 displaying the 24-month forecast points. The forecasted values are all within the required bounds, indicating reliability in the predictions. The deep blue shaded area represents the 80% confidence level, while the light blue shaded area corresponds to the 95% confidence level. Notably, all forecasted points fall within the 95% confidence bounds, demonstrating that the predictions are within an acceptable range and, therefore, reliable.

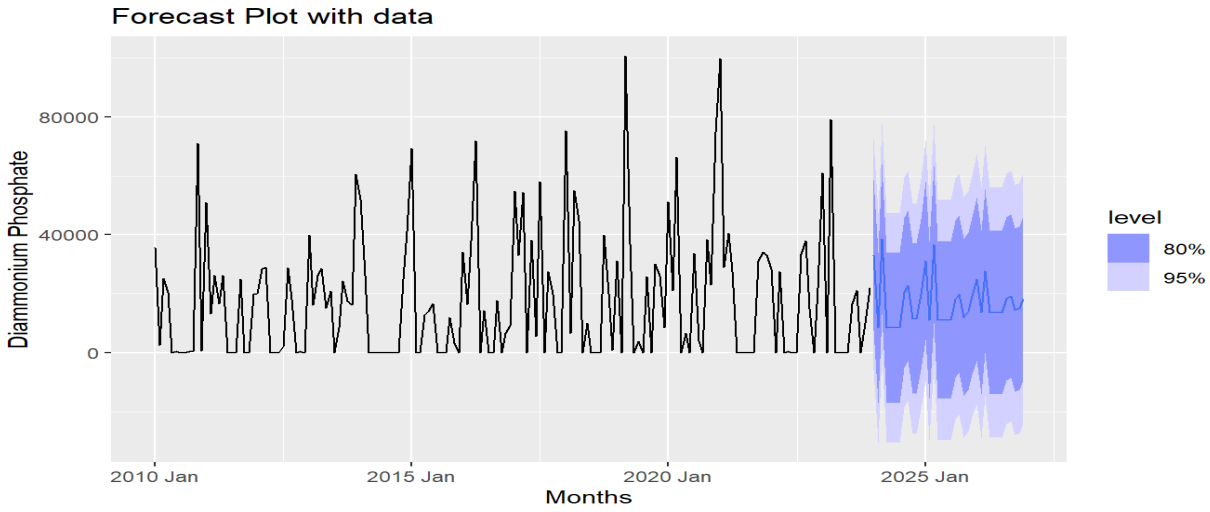


Figure 61: Forecasts plot for diammonium phosphate fertiliser

Table 15: Demand forecasts for diammonium phosphate fertiliser

Month	Qty (in metric tons)	Month	Qty (in metric tons)
2024 Aug	N (20405, 4e+08)	2025 Aug	N (18139, 4.3e+08)
2024 Sep	N (22718, 4e+08)	2025 Sep	N (19780, 4.3e+08)
2024 Oct	N (11500, 4e+08)	2025 Oct	N (11988, 4.3e+08)
2024 Nov	N (11518, 4e+08)	2025 Nov	N (14048, 4.3e+08)
2024 Dec	N (19764, 4e+08)	2025 Dec	N (19073, 4.3e+08)
2025 Jan	N (31203, 4.3e+08)	2026 Jan	N (24896, 4.7e+08)
2025 Feb	N (11040, 4.3e+08)	2026 Feb	N (13582, 4.7e+08)
2025 Mar	N (36638, 4.3e+08)	2026 Mar	N (27698, 4.7e+08)
2025 Apr	N (11040, 4.3e+08)	2026 Apr	N (13582, 4.7e+08)
2025 May	N (11068, 4.3e+08)	2026 May	N (13610, 4.7e+08)
2025 Jun	N (11040, 4.3e+08)	2026 Jun	N (13582, 4.7e+08)
2025 Jul	N (11040, 4.3e+08)	2026 Jul	N (13582, 4.7e+08)

4.4.1.4 Forecast on Muriate of Potash (24 months)

Figure 62 presents the forecasted demand for Muriate of Potash fertiliser, with Table 16 displaying the 24-month forecast points. The forecasted values are all within the required bounds, with the deep blue shaded area representing the 80% confidence level and the light blue shaded area representing the 95% confidence level. Notably, all forecasted points fall within the 95% confidence bounds, indicating that the predictions are reliable. The forecasts suggest a slight variation in demand during the first ten months (August 2024 to May 2025), after which the demand remains constant at 1,602 metric tonnes. This stability in demand reflects the model's short-term predictive ability.

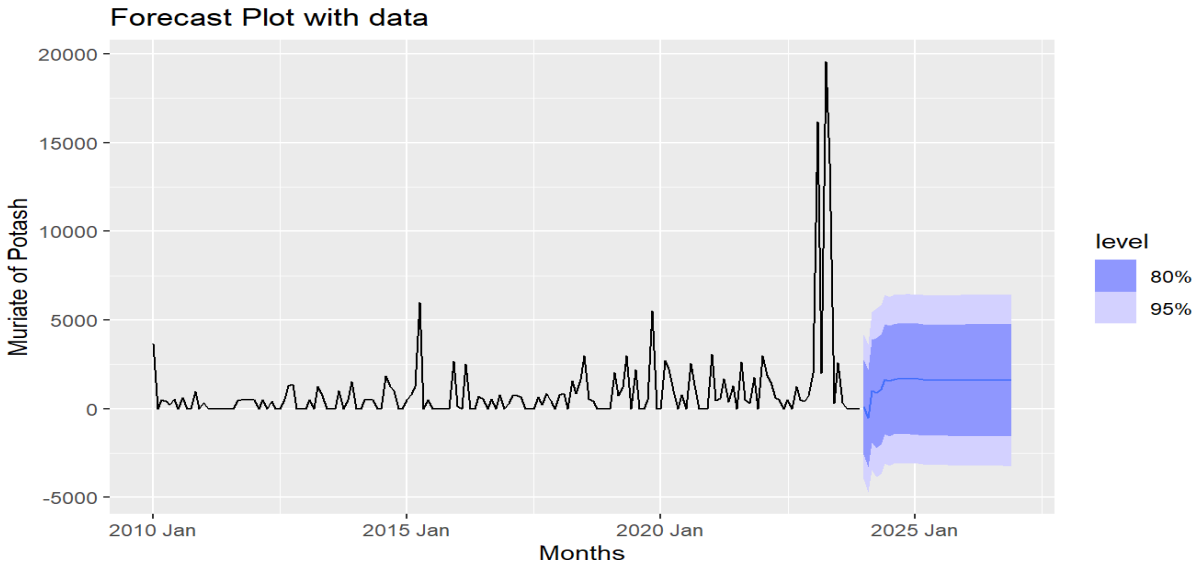


Figure 62: Forecast plot for muriate of potash demand

Table 16: Forecasted demand for muriate of potash fertiliser

Month	Qty (in metric tons)	Month	Qty (in metric tons)
2024 Aug	N (1660, 5894573)	2025 Aug	N (1602, 6e+06)
2024 Sep	N (1672, 5902894)	2025 Sep	N (1602, 6e+06)
2024 Oct	N (1673, 5911216)	2025 Oct	N (1602, 6e+06)
2024 Nov	N (1673, 5919538)	2025 Nov	N (1602, 6e+06)
2024 Dec	N (1673, 5927860)	2025 Dec	N (1602, 6e+06)
2025 Jan	N (1666, 5927854)	2026 Jan	N (1602, 6e+06)
2025 Feb	N (1667, 5932904)	2026 Feb	N (1602, 6e+06)
2025 Mar	N (1627, 5936230)	2026 Mar	N (1602, 6e+06)
2025 Apr	N (1600, 5939197)	2026 Apr	N (1602, 6e+06)
2025 May	N (1602, 5947347)	2026 May	N (1602, 6e+06)
2025 Jun	N (1602, 6e+06)	2026 Jun	N (1602, 6e+06)
2025 Jul	N (1602, 6e+06)	2026 Jul	N (1602, 6054042)

4.4.1.5 Forecast of Nitrogen, Phosphorus, and Potassium fertiliser Demand (24 months)

Figure 63 and Table 17 present the forecasted demand for the next 24 months, with all forecasted values falling within the required confidence bounds. The deep blue shaded area on the plot represents the 80% confidence level, while the light blue shaded area represents the 95% confidence level, indicating that the predictions are reliable. The forecasted quantities vary significantly from month to month, particularly in the first year, with peaks in November 2024 (40,289 metric tonnes) and October 2024 (22,210 metric tonnes). In contrast, the second year shows more stability, with lower demand fluctuations. This trend suggests that while demand may be volatile in the short term, it stabilizes over time. The consistent adherence to the 95% confidence bounds throughout the forecast period highlights the reliability of these predictions.

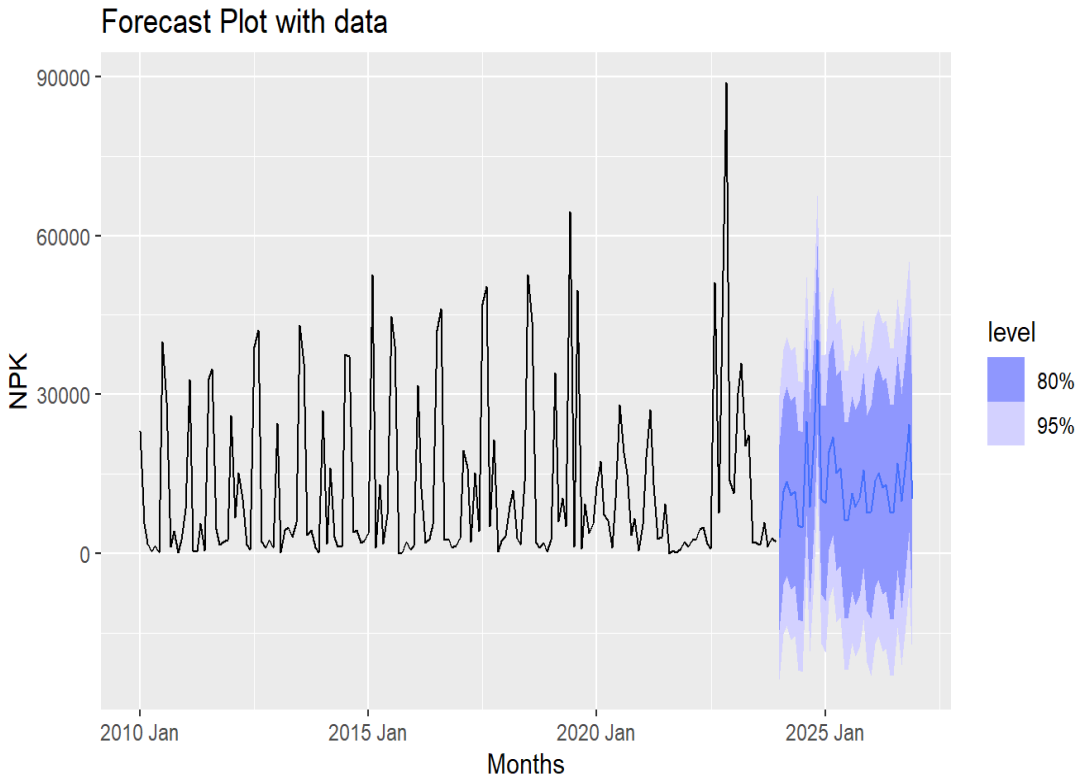


Figure 63: Forecast plot nitrogen, phosphorus, and potassium fertiliser demand

Table 17: Demand forecast points for nitrogen, phosphorus, potassium fertiliser

Month	Qty (in metric tons)	Month	Qty (in metric tons)
2024 Aug	N (24918, 1.9e+08)	2025 Aug	N (11261, 2.1e+08)
2024 Sep	N (8679, 1.9e+08)	2025 Sep	N (8722, 2.1e+08)
2024 Oct	N (22210, 1.9e+08)	2025 Oct	N (10357, 2.1e+08)
2024 Nov	N (40289, 1.9e+08)	2025 Nov	N (15709, 2.1e+08)
2024 Dec	N (10143, 1.9e+08)	2025 Dec	N (7619, 2.1e+08)
2025 Jan	N (9449, 2.1e+08)	2026 Jan	N (7758, 2.5e+08)
2025 Feb	N (19190, 2.1e+08)	2026 Feb	N (13743, 2.5e+08)
2025 Mar	N (21957, 2.1e+08)	2026 Mar	N (15241, 2.5e+08)
2025 Apr	N (15090, 2.1e+08)	2026 Apr	N (12412, 2.5e+08)
2025 May	N (16096, 2.1e+08)	2026 May	N (12990, 2.5e+08)
2025 Jun	N (6314, 2.1e+08)	2026 Jun	N (7902, 2.5e+08)
2025 Jul	N (6193, 2.1e+08)	2026 Jul	N (7714, 2.5e+08)

4.4.1.6 Urea Demand Forecast (24 months)

Figure 64 and Table 18 present the 24-month forecasted demand for Urea fertiliser, showing a consistent quantity of 7,994 metric tons each month from August 2024 to July 2026. All forecasted values fall within the required confidence bounds, with the deep blue shaded area representing the 80% confidence level and the light blue shaded area representing the 95% confidence level. The fact that every forecasted point lies within the 95% confidence bounds indicates the reliability of these predictions. The forecast suggests a stable and unchanging demand for Urea fertiliser over the entire period, reflecting high predictability and consistency in the model's outputs.

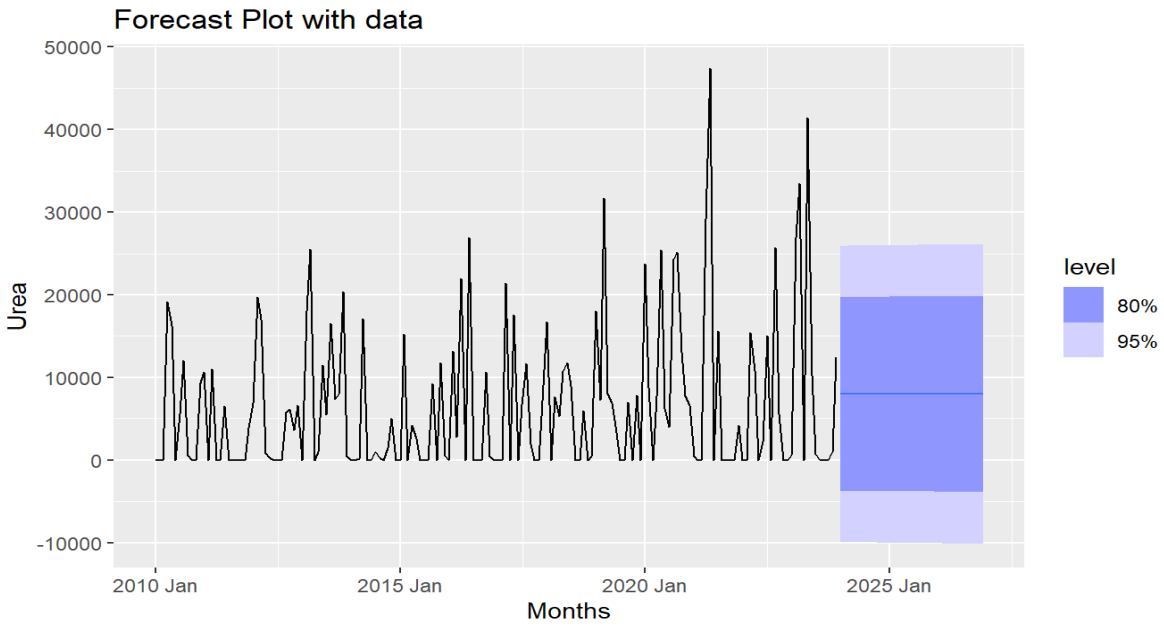


Figure 64: Forecast plot for demand for Urea

Table 18: Forecast demand points for urea

Month	Qty (in metric tons)	Month	Qty (in metric tons)
2024 Aug	N (7994, 8.4e+07)	2025 Aug	N (7994, 8.4e+07)
2024 Sep	N (7994, 8.4e+07)	2025 Sep	N (7994, 8.4e+07)
2024 Oct	N (7994, 8.4e+07)	2025 Oct	N (7994, 8.4e+07)
2024 Nov	N (7994, 8.4e+07)	2025 Nov	N (7994, 8.4e+07)
2024 Dec	N (7994, 8.4e+07)	2025 Dec	N (7994, 8.4e+07)
2025 Jan	N (7994, 8.4e+07)	2026 Jan	N (7994, 8.4e+07)
2025 Feb	N (7994, 8.4e+07)	2026 Feb	N (7994, 8.4e+07)
2025 Mar	N (7994, 8.4e+07)	2026 Mar	N (7994, 8.4e+07)
2025 Apr	N (7994, 8.4e+07)	2026 Apr	N (7994, 8.4e+07)
2025 May	N (7994, 8.4e+07)	2026 May	N (7994, 8.4e+07)
2025 Jun	N (7994, 8.4e+07)	2026 Jun	N (7994, 8.4e+07)
2025 Jul	N (7994, 8.4e+07)	2026 Jul	N (7994, 8.4e+07)

4.4.1.7 Forecast of Total Fertiliser Demand (24 months)

Figure 65 and Table 19 present the 24-month forecasted demand for Total Fertiliser. The forecasted values vary from August 2024 to July 2025, with quantities ranging between 54,207 and 60,338 metric tonnes. From August 2025 onwards, the demand stabilises at 60,243 metric tonnes per month. All forecasted values are within the required bounds, with the deep blue shaded area representing the 80% confidence level and the light blue shaded area representing the 95% confidence level. Notably, every forecasted point falls within the 95% confidence bounds, indicating that the predictions are reliable. The initial variation followed by stabilisation suggests that while short-term fluctuations may occur, the long-term demand is predictable and consistent.

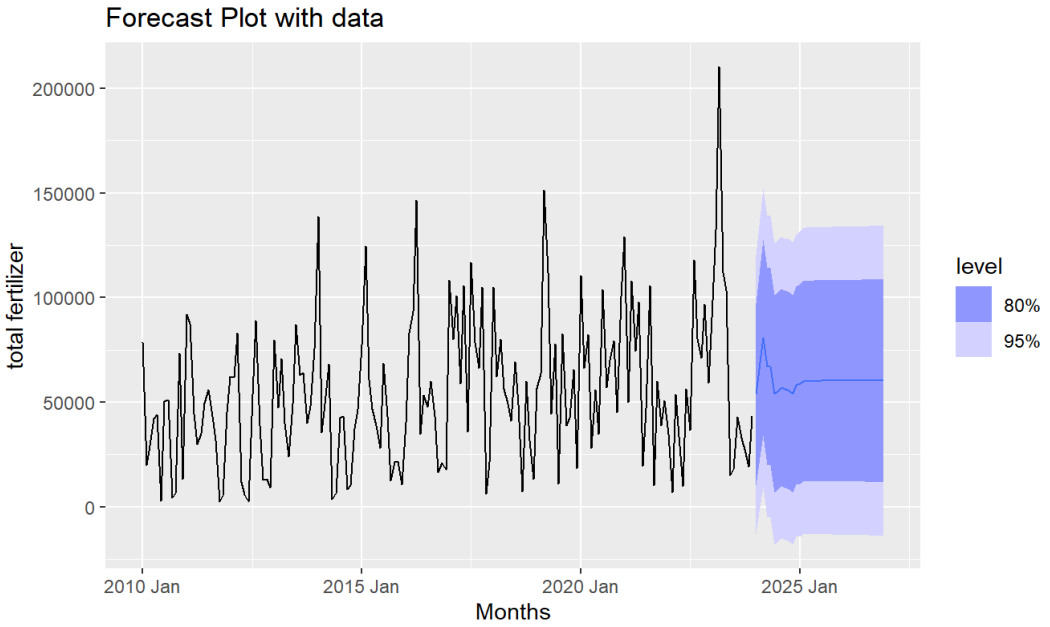


Figure 65: Forecast plot total fertiliser demand

Table 19: Forecasted demand for total fertiliser

Month	Qty (in metric tons)	Month	Qty (in metric tonnes)
2024 Aug	N (57070, 1.4e+09)	2025 Aug	N (60237, 1.4e+09)
2024 Sep	N (56238, 1.4e+09)	2025 Sep	N (60246, 1.4e+09)
2024 Oct	N (55731, 1.4e+09)	2025 Oct	N (60241, 1.4e+09)
2024 Nov	N (54207, 1.4e+09)	2025 Nov	N (60244, 1.4e+09)
2024 Dec	N (58008, 1.4e+09)	2025 Dec	N (60242, 1.4e+09)
2025 Jan	N (58813, 1.4e+09)	2026 Jan	N (60243, 1.4e+09)
2025 Feb	N (60030, 1.4e+09)	2026 Feb	N (60243, 1.4e+09)
2025 Mar	N (60338, 1.4e+09)	2026 Mar	N (60243, 1.4e+09)
2025 Apr	N (60188, 1.4e+09)	2026 Apr	N (60243, 1.4e+09)
2025 May	N (60274, 1.4e+09)	2026 May	N (60243, 1.4e+09)
2025 Jun	N (60225, 1.4e+09)	2026 Jun	N (60243, 1.4e+09)
2025 Jul	N (60253, 1.4e+09)	2026 Jul	N (60243, 1.4e+09)

One effective method for assessing forecast accuracy is by comparing actual versus fitted data, a principle that involves evaluating how closely forecasted values align with observed outcomes (Hyndman and Athanasopoulos, 2018). This approach provides insights into the reliability of the forecasting model and its predictions. Hyndman and Athanasopoulos (2018) used this method to validate their model's performance by juxtaposing predicted values against actual sales figures, thereby confirming the accuracy of their forecasts (Hyndman & Athanasopoulos, 2018). Similarly, Gneiting and Raftery (2007) demonstrated the effectiveness of this approach in weather

forecasting, where comparing forecasted temperatures with actual observations provided a robust measure of forecast accuracy (Gneiting & Raftery, 2007). In this analysis, the comparison between forecasted and actual data revealed insignificant discrepancies, supporting the accuracy of the forecast, as corroborated by the study by Hyndman and Athanasopoulos (2018) and Gneiting and Raftery (2007).

4.4.2 Evaluation of the Forecast Accuracy

The accuracy test is done to test the accuracy of the forecast of each fertiliser type. The outputs below show each model's point focus and forecast distribution. Each plot indicates that the data have a normal distribution with mean and variance. This kind of focus assumes that residuals are following a normal distribution. If we assume that the residuals are not following normal distribution, we have to use bootstrapping to test whether the residues are normally distributed. Below are the forecast distributions and the actual and fitted data comparisons.

4.4.2.1 Evaluation of the Forecast Accuracy for Calcium Ammonium Nitrate

The plots of the forecast distribution for CAN indicate that the forecasts are normally distributed. The graph in Figure 66 indicates that the forecasted values follow a normal distribution, as suggested by the symmetry and distribution of the residuals around the predicted points. This normality is crucial, as it validates the model's underlying assumptions and enhances the reliability of the forecast. This has also been confirmed by the plot of the residue's normality in Figure 68. The graph in Figure 67 compares the ARIMA model's fitted values with the actual historical CAN data, highlighting how the model captures the data's trends. The fitted line is straight because the constant term explains the model well, as indicated in Equation 42.

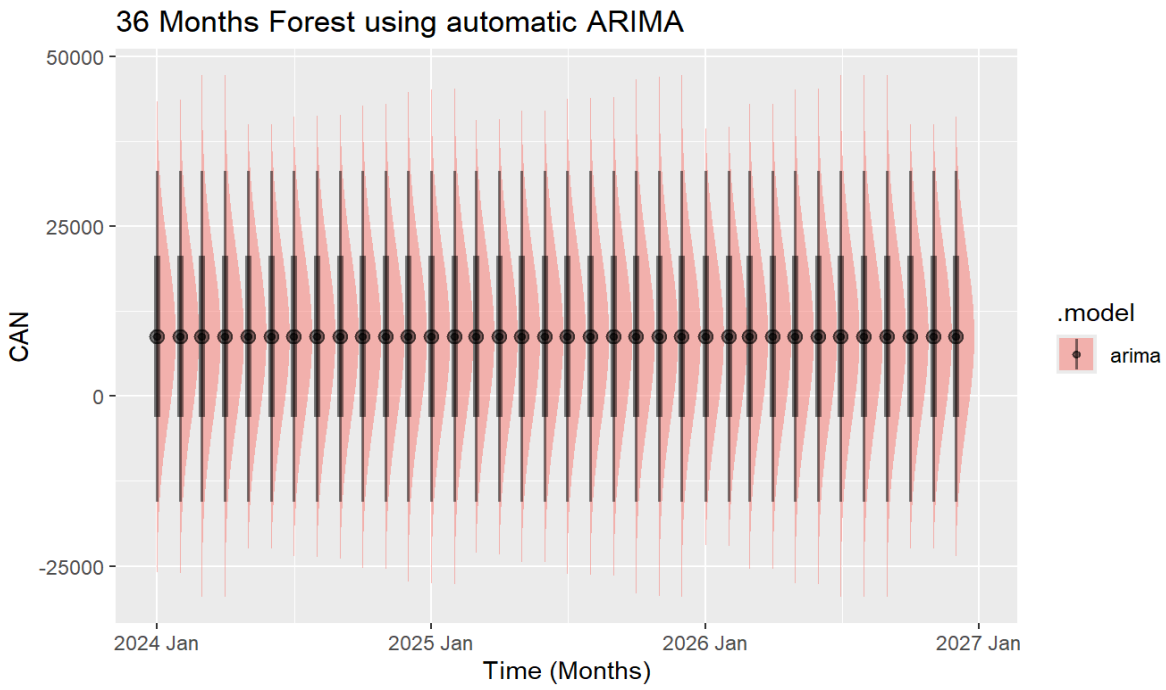


Figure 66: Distribution of the forecast for calcium ammonium nitrate.

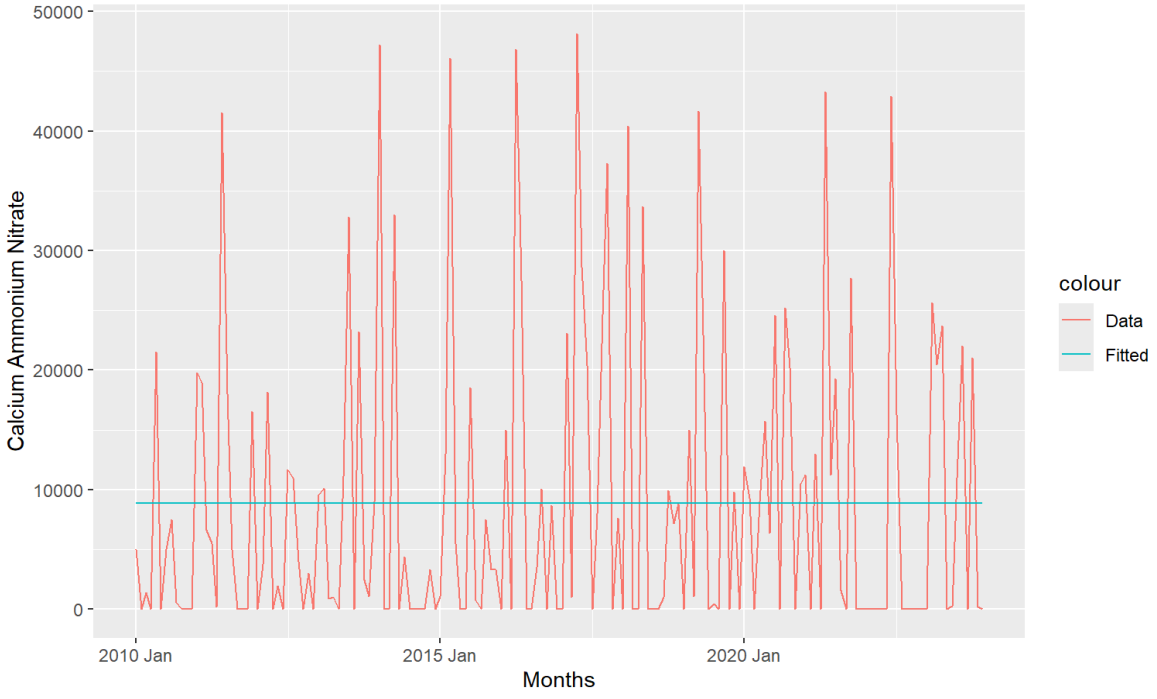


Figure 67: Fitted vs. actual time series for calcium ammonium nitrate

The Shapiro-Wilk normality test was conducted to assess the normality of the residuals from the SARIMA model. The test yielded a W = 0.74617 and a p-value of <0.0001. The closer the value of W, it is an indication that the residues approach normality.

Therefore, the tests indicated that the residues approach normality but not a perfect normal distribution. Figure 68 also confirms that the residues slightly approach the normal distribution.

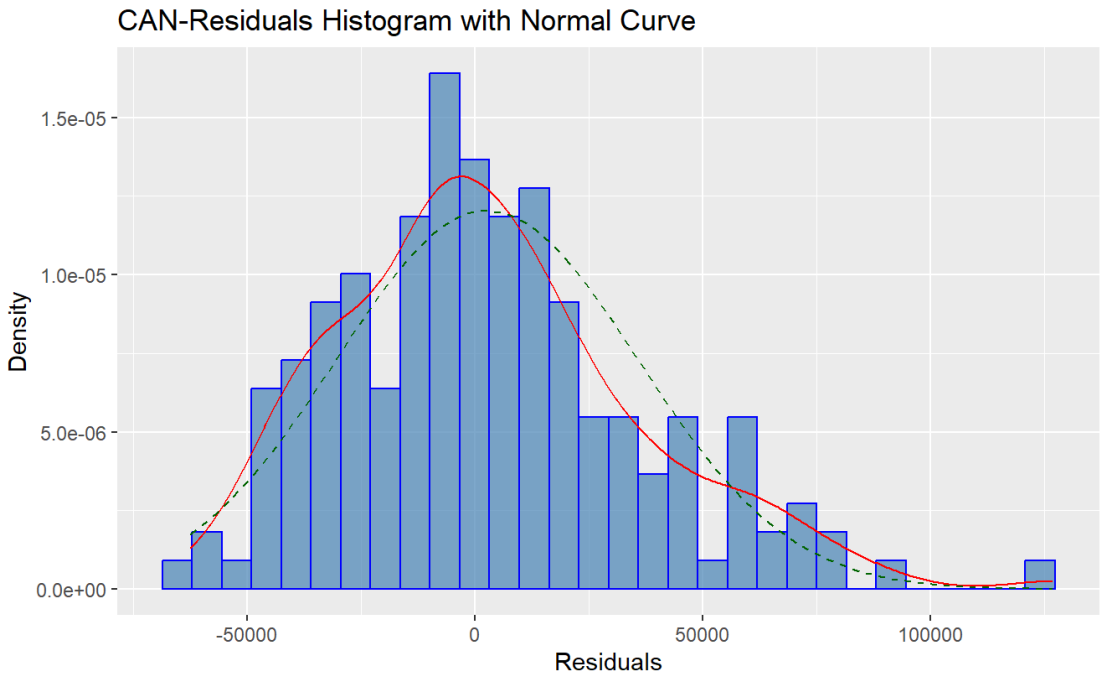


Figure 68: Residue normality test for calcium ammonium nitrate time series

The figure below is Q-Q plots that point out the standardised deviations of the residues from the normality. The plot indicates that the residues slightly deviate from the normal, indicating that the model's accuracy is reliable in the short term.

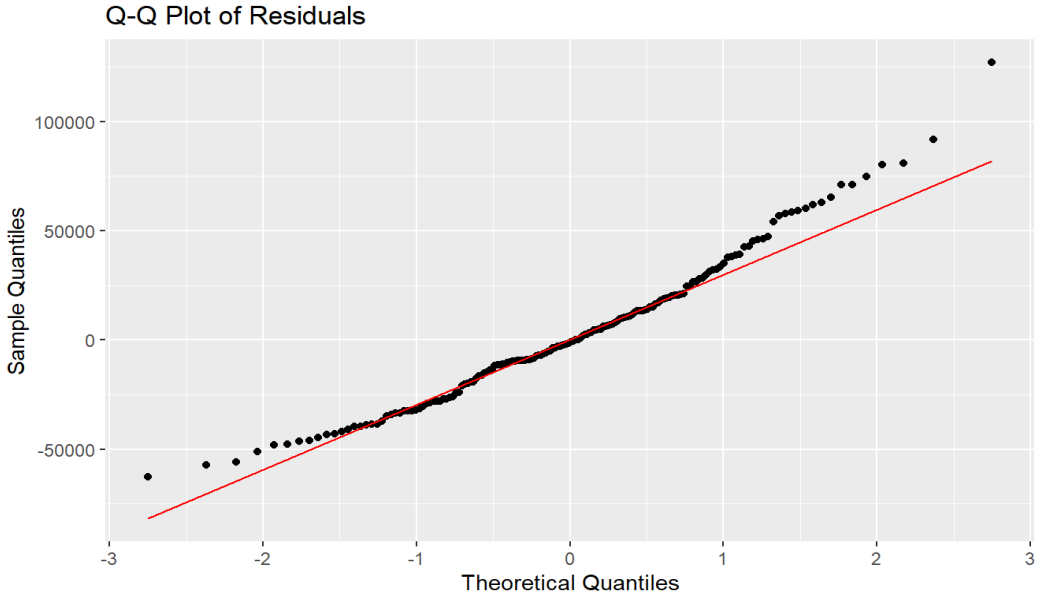


Figure 69: Standardised deviations from normality for calcium ammonium nitrate

4.4.2.2 Evaluation of the Forecast Accuracy for Calcium Nitrate

Figure 70 is a plot of the forecast of the Calcium nitrate forecast points. The forecast point indicates that they are normally distributed and that the model is a good fit. The graph in Figure 70 illustrates the actual and fitted values of Calcium Nitrate demand from January 2010 to December 2023. The red line represents the actual data, while the blue line represents the fitted model. While the lines closely follow each other throughout most of the period, slight deviations exist between the fitted values and the actual data, particularly during peaks. However, these deviations are not substantial, suggesting that the fitted model effectively captures the overall trend and variability in the data. This close alignment indicates that the model provides a reliable prediction of Calcium Nitrate demand, with only minor discrepancies in more volatile periods. The model's ability to follow the data closely throughout most time series supports its reliability for forecasting future demand.

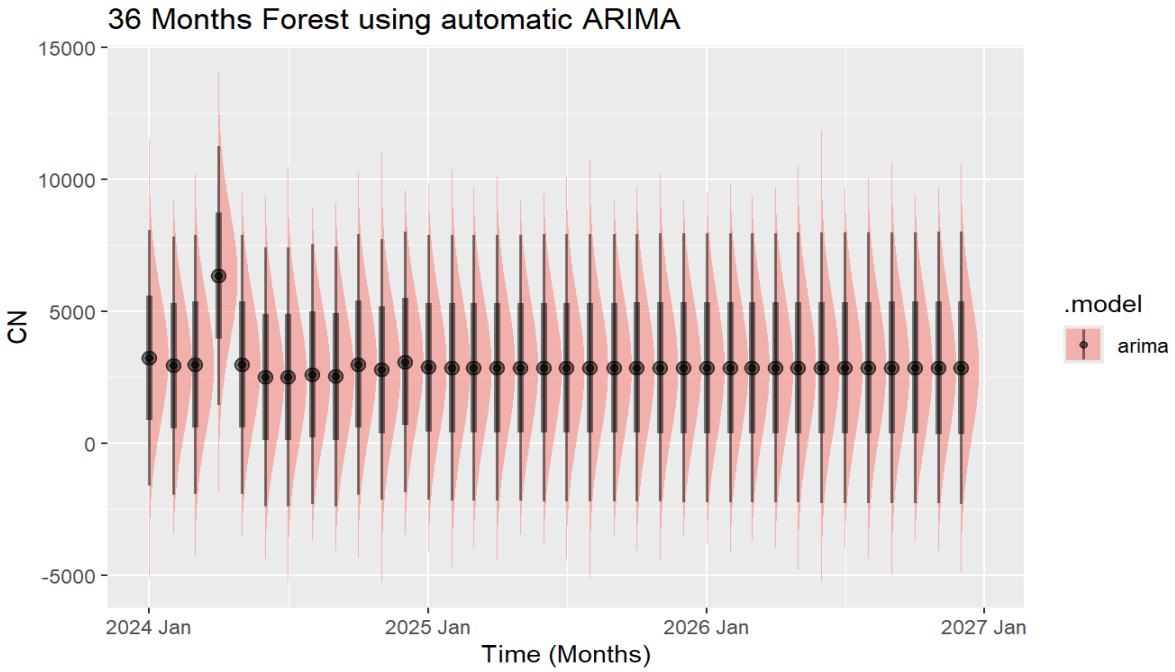


Figure 70: Distribution of the forecast for calcium nitrate.

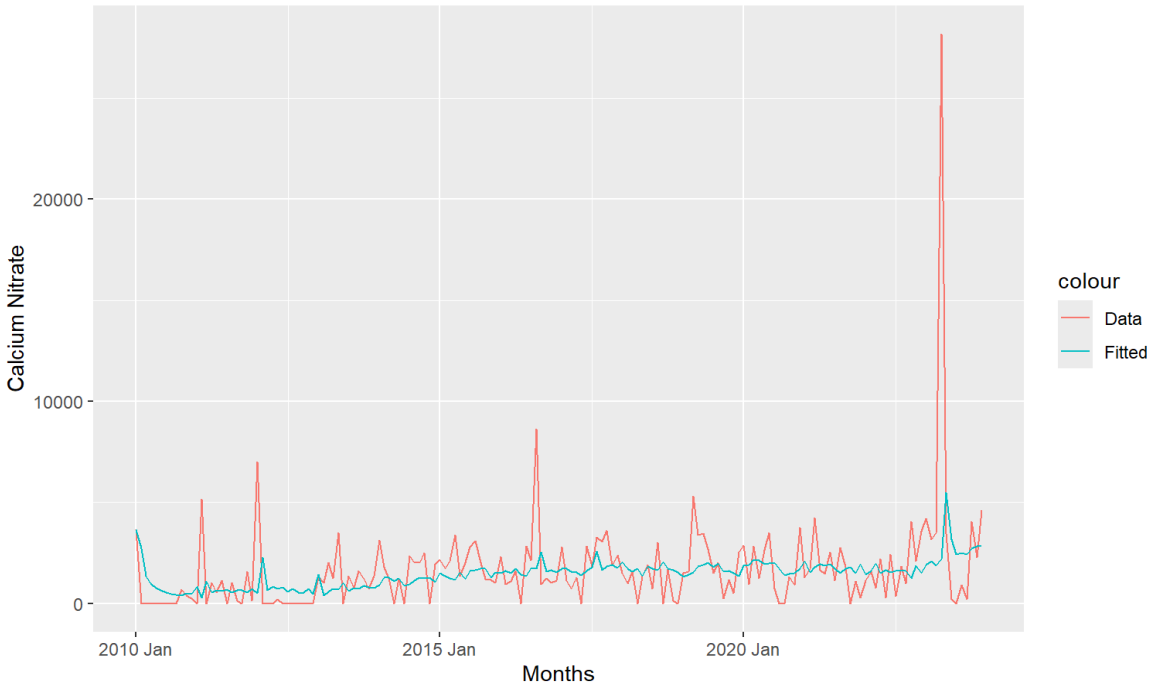


Figure 71: Actual vs. fitted time series for calcium nitrate

The Shapiro-Wilk normality test also indicated that W = 0.50569: This test statistic is significantly less than 1. W much lower than 1 suggests that the residuals are not normally distributed. The results pointed out that the residues did not follow normal distribution. A gg plot was done to confirm these results and point out that the closely attained normal distribution. Figure 72 shows the graphical plot of the residues and the normality curve. The graph indicates that the residues slightly approach normal distribution. The deviation from normal is illustrated in the graph in Figure 73.

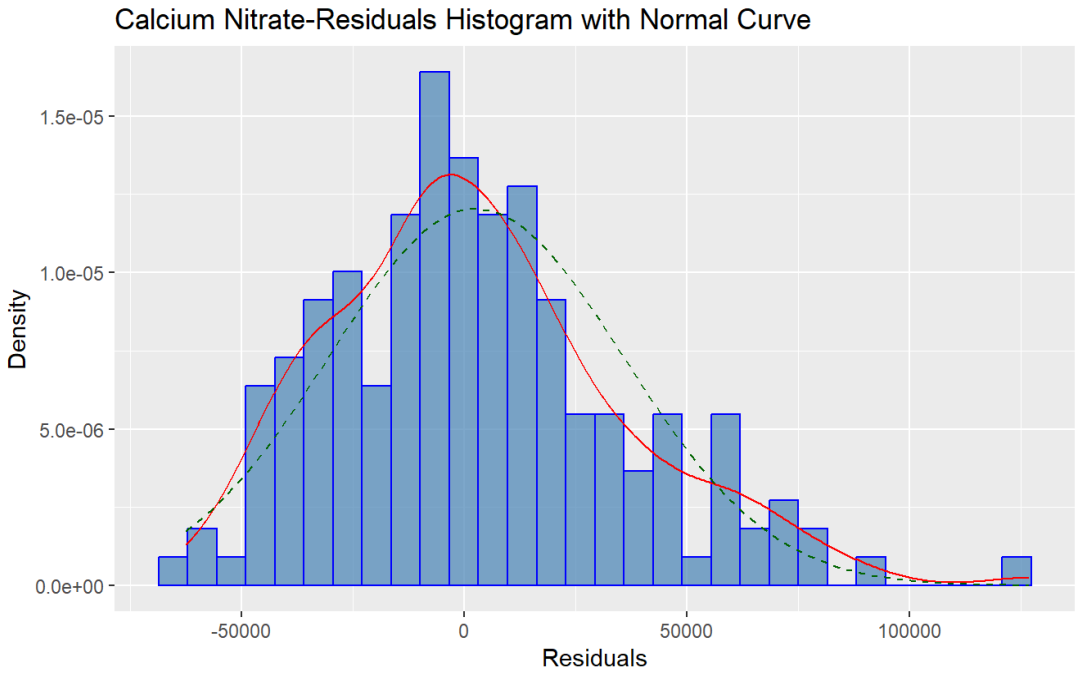


Figure 72: Residuals and normality for calcium nitrate

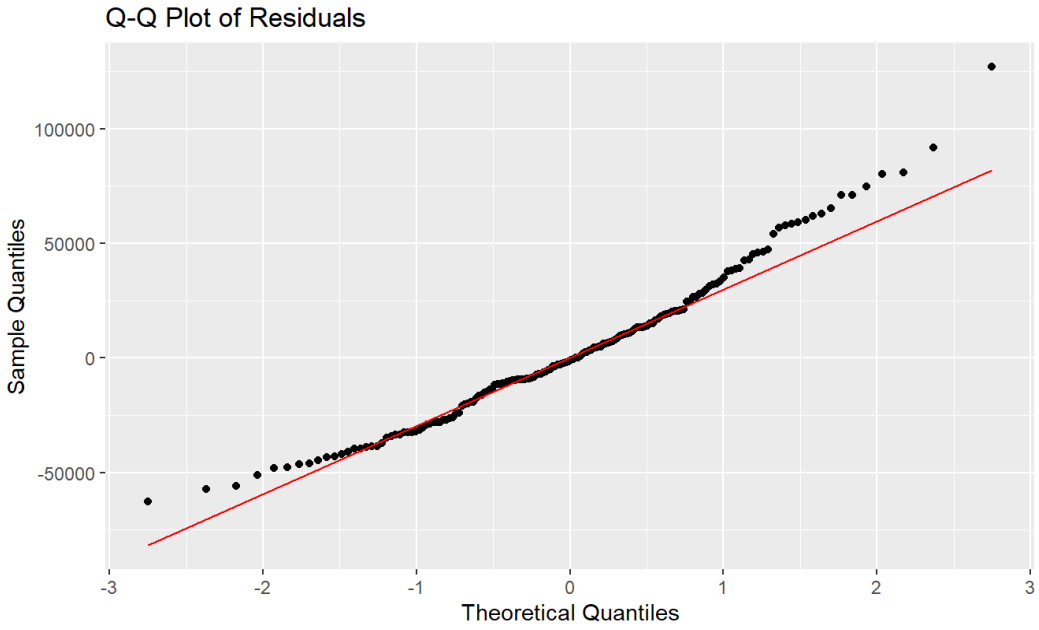


Figure 73: Standardised deviations from normality for calcium nitrate

4.4.2.3 Evaluation of the Forecast Accuracy for Diammonium Phosphate

The plots to test the normality of the forecasts indicated that the forecasts are normally distributed (as shown in Figure 74). A plot to check the deviation between the fitted and actual data also indicates slight deviations, confirming that the model was a good fit for

forecasting. The Shapiro-Wilk normality test yielded W = 0.90764 and a p-value = 8.636e-09. W is close to 1, indicating that the residues closely approached normal distribution. The graph in Figure 77 shows standardized deviations of residuals from normality

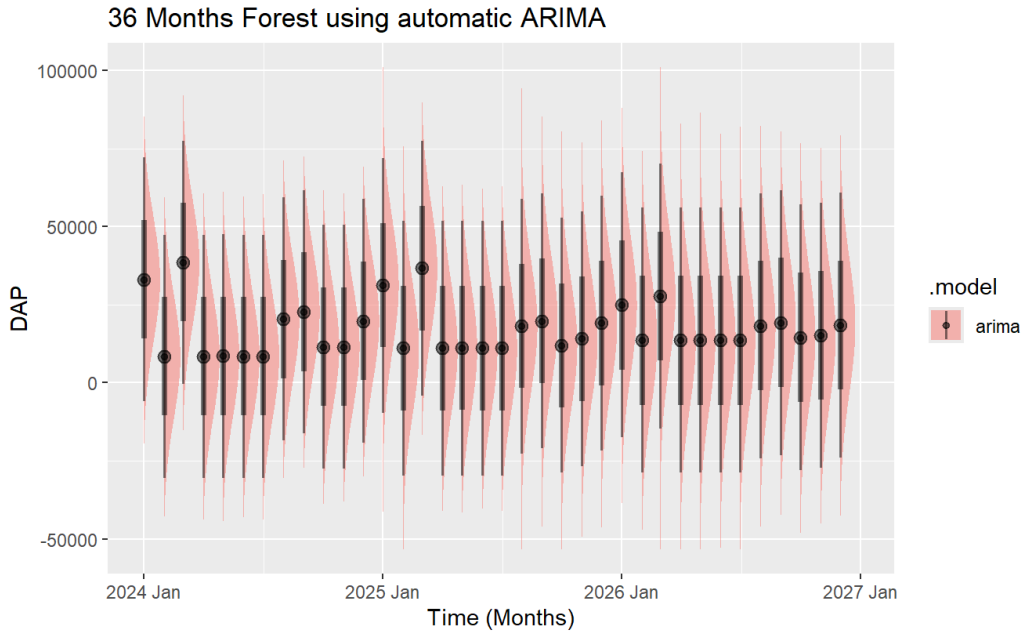


Figure 74: Distribution of the forecast for diammonium phosphate.

Figure 75 presents the actual and fitted values of the Diammonium Phosphate demand from January 2010 to Dec 2023. The red line indicates the actual data, while the blue line represents the fitted model. The graph indicates pronounced variability, with frequent and sharp spikes in demand. Although the fitted values follow the general pattern of the actual data, there are noticeable deviations, particularly during periods of high demand. Despite these discrepancies, the fitted model still captures the data's overall trend and seasonal fluctuations. The model's ability to generally track the actual demand, even with some deviations during peak periods, suggests that the predictions remain reliable. A plot to check the normality of the residues (in figure 76) indicated that the residues slightly follow normality, a confirmation of the model's accuracy.

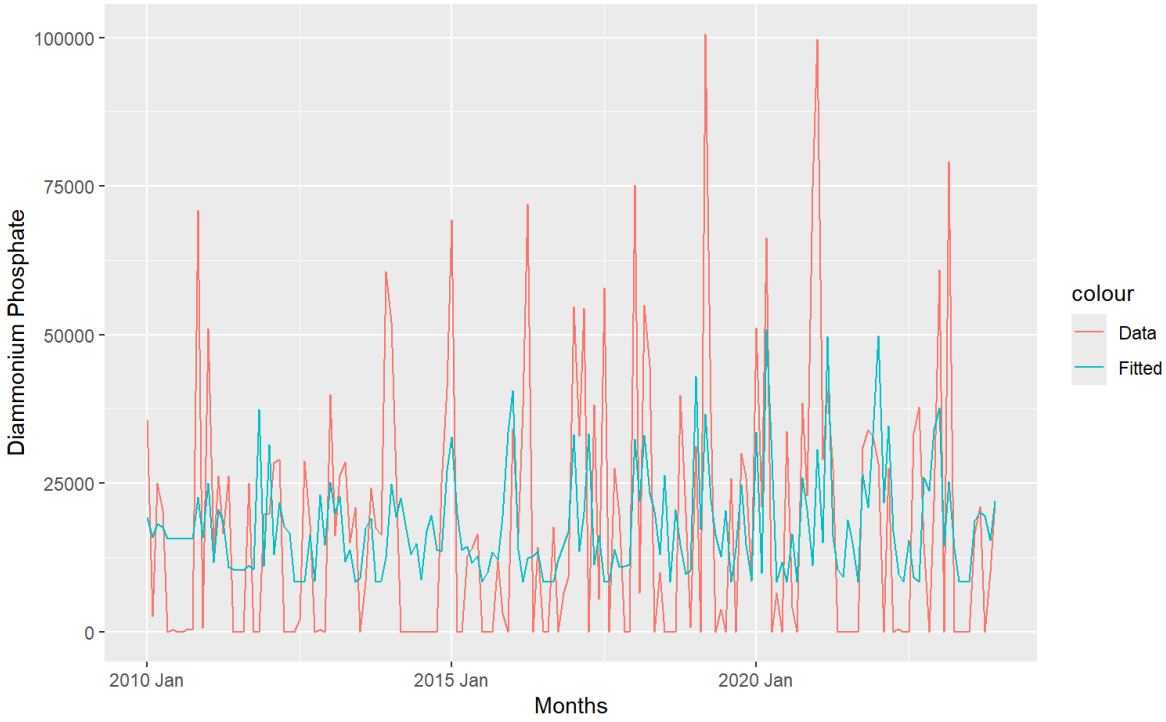


Figure 75: Fitted vs. actual time series for diammonium phosphate

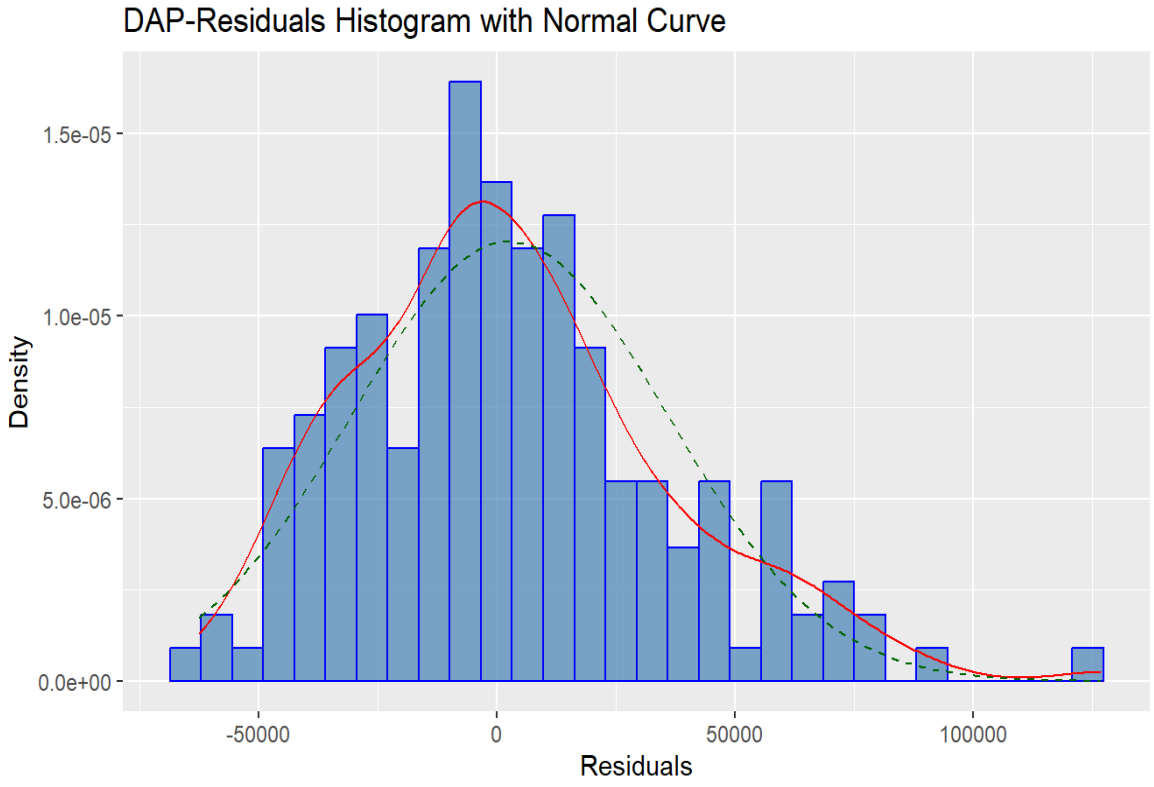


Figure 76: Residuals vs. normal for diammonium phosphate

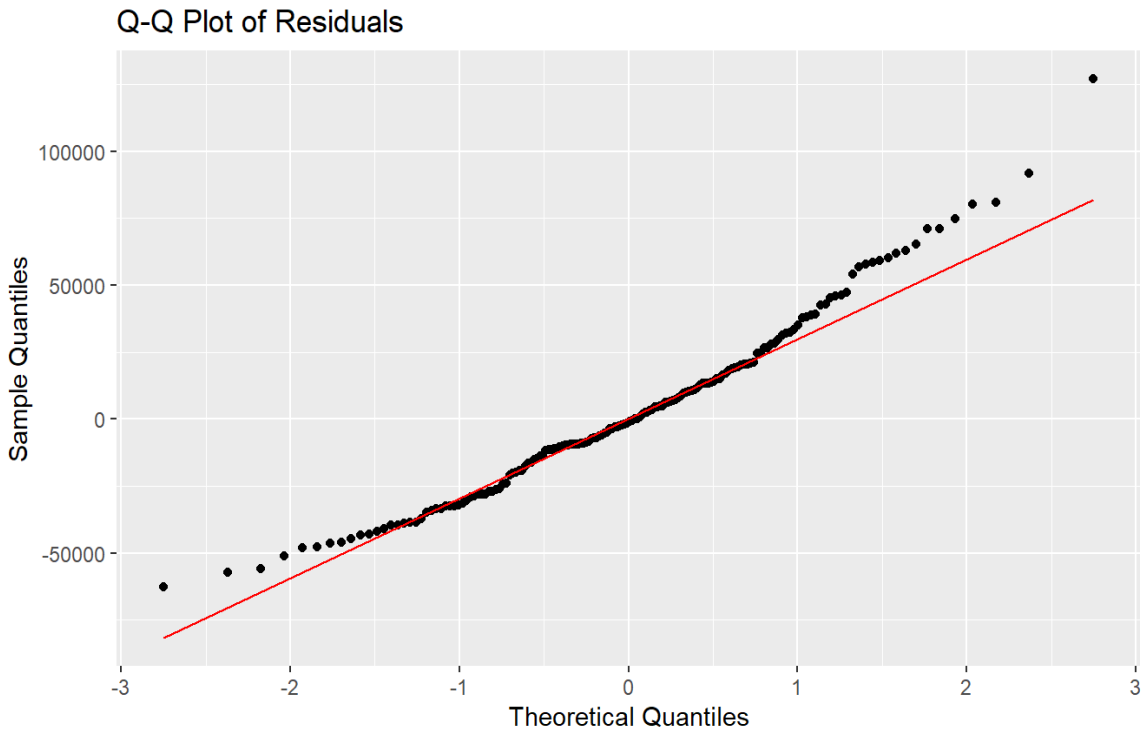


Figure 77: Standardised residuals plot for diammonium phosphate

4.4.2.4 Evaluation of the Forecast Accuracy for Muriate of Potash

The plots of the SARIMA points indicated that the forecasted points are normally distributed (figure 78). In addition, the plot of the forecasted points and the actual data point (figure 79) indicated a slight difference, confirming the accuracy of the fitted model. To test the normality of the data, the Shapiro-Wilk normality test was also done and yielded W= W = 0.6356, p-value <0.0001, which is also toward one, indicating that the residuals approached normal distribution. To visualize the normality of the residues, a histogram plot with a standard curve was fitted, and it ascertained that the residues differed slightly from the normal distribution (Figure 80).

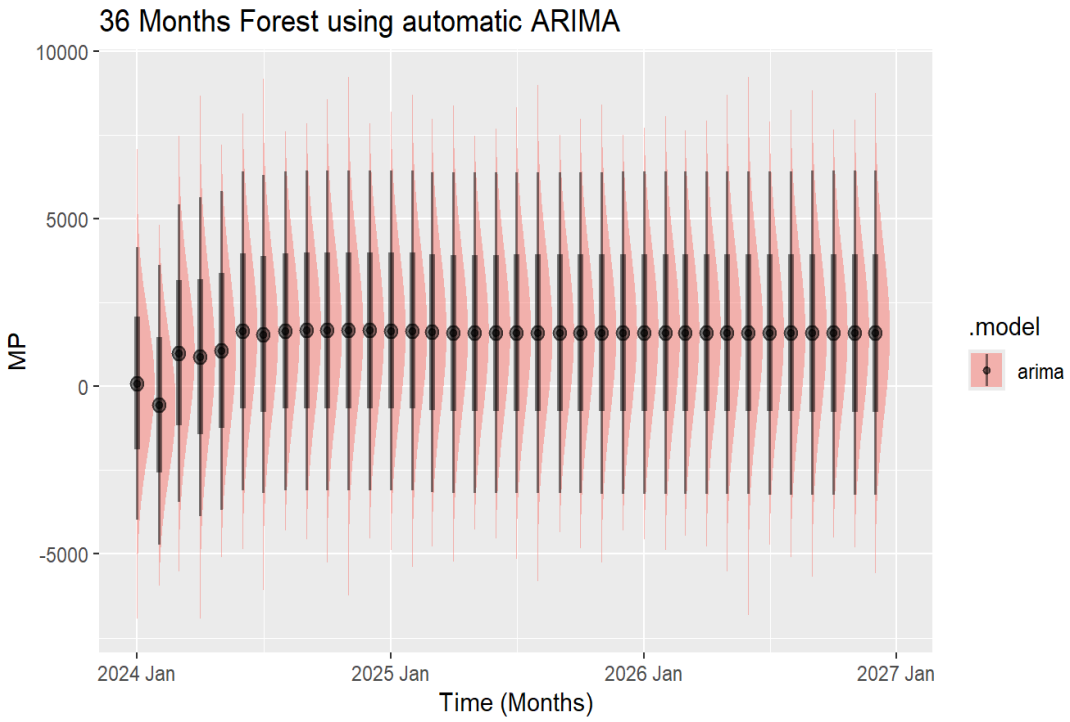


Figure 78: Distribution of the forecast Muriate of Potash.

The graph in figure 79 reveals high variability in the actual demand, with frequent and significant spikes, particularly towards the later part of the time series. The fitted model follows the overall trend and seasonal patterns of the data, although there are some discrepancies, especially during periods of peak demand. Despite these deviations, the fitted model has effectively captured the general movement and fluctuations in demand closely. This close alignment suggests that the predictions are reliable.

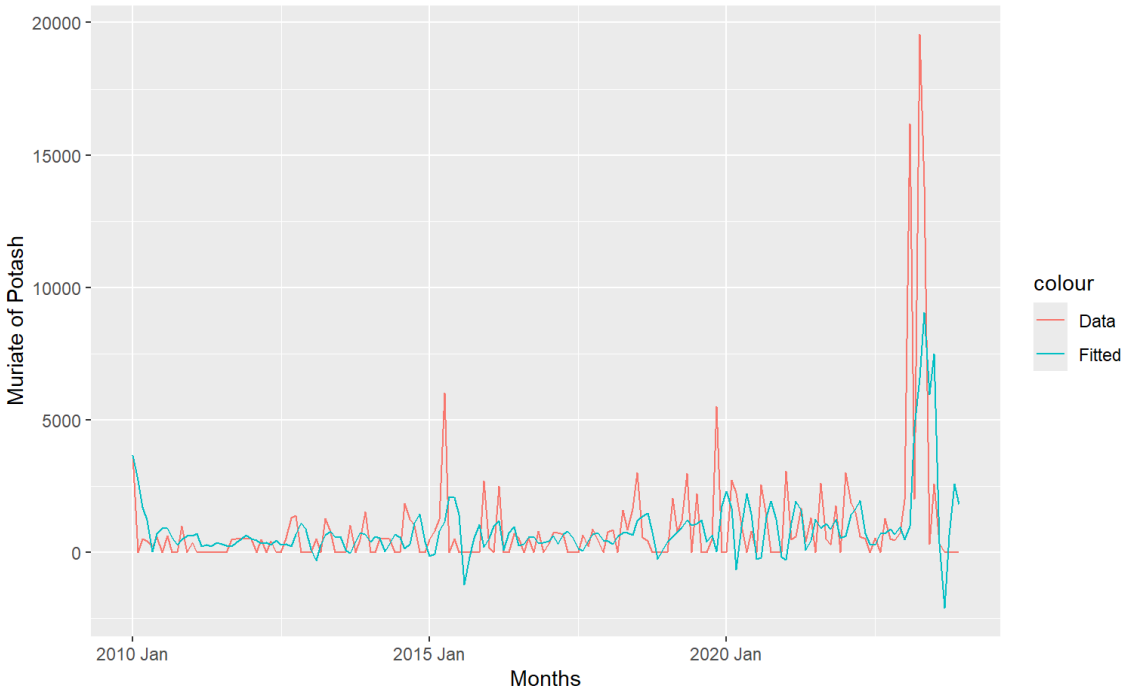


Figure 79: Fitted Vs. Actual fertiliser demand for muriate of potash

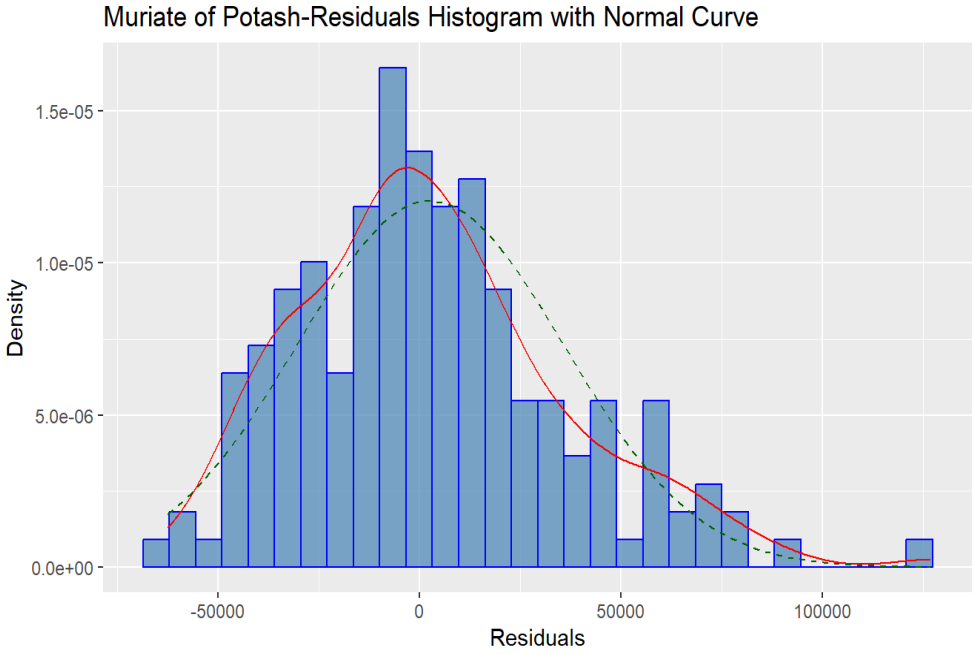


Figure 80: Residuals and normality for muriate of potash time series

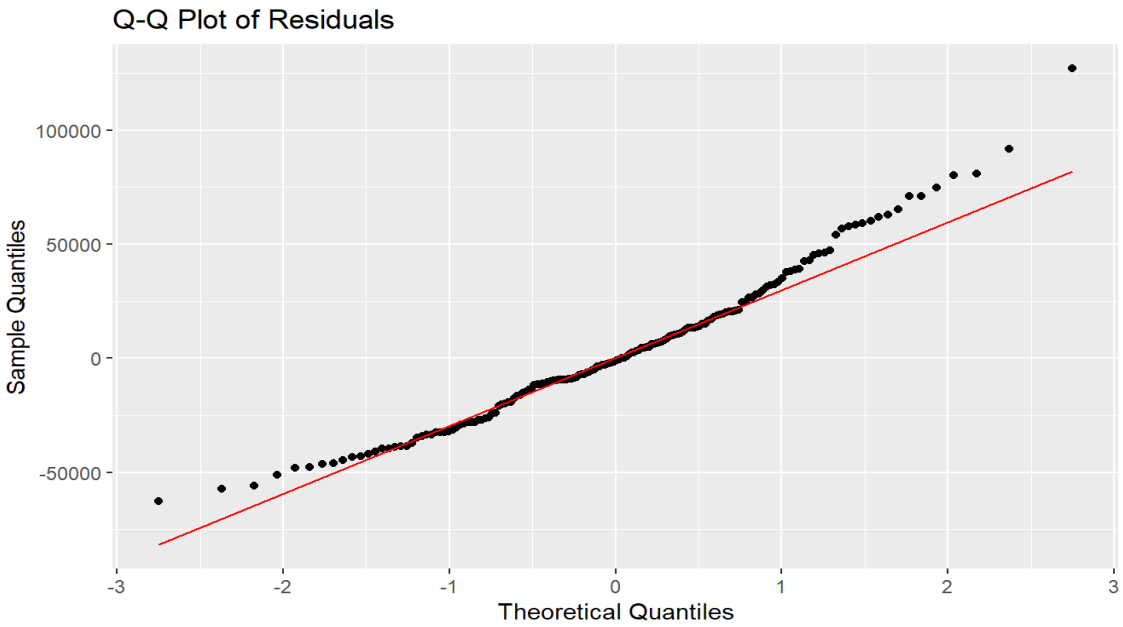


Figure 81: Standardised residuals for muriate of potash

4.4.2.5 Evaluation of the Forecast Accuracy for Nitrogen, Phosphorus and Potassium Fertiliser

A plot of the SARIMA model forecast points indicated that the fitted points were normally distributed, as shown in graph 82. A plot of fitted data and the actual data points (figure 83) showed slight deviations, confirming that the model was a good fit. The normality of the residues was also determined using the Shapiro-Wilk normality test, which yielded the values W= W = 0.83217 and p-value = <0.0001. This value is close to 1, confirming that the residues followed a normal distribution. The normality of the distribution is also confirmed by the histogram of residues in Figures 84 and 85.

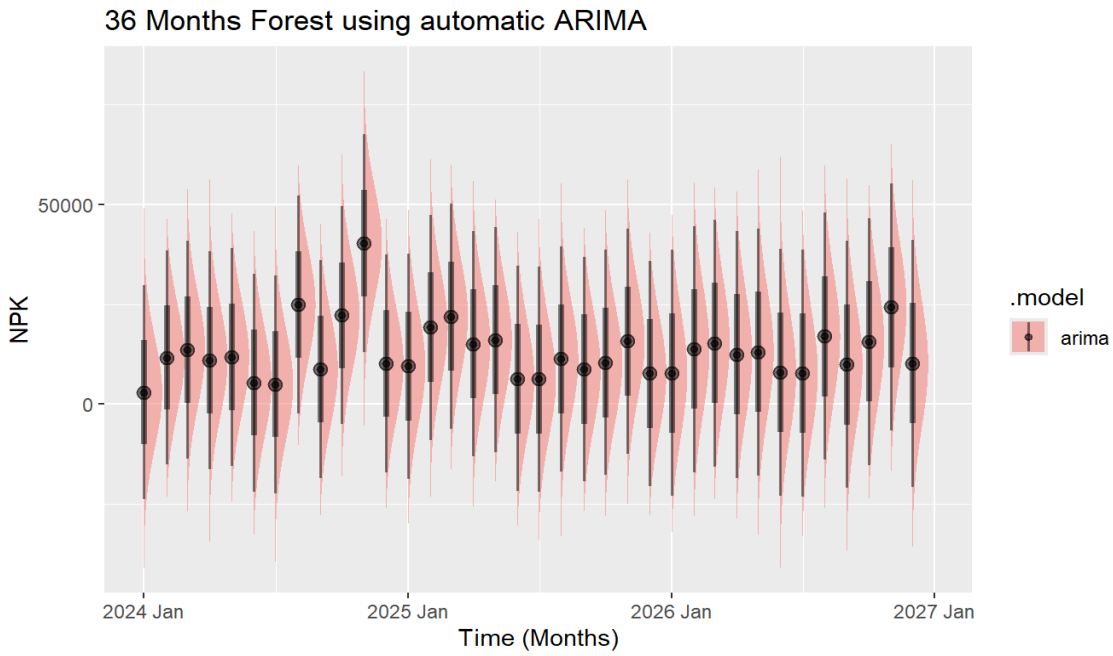


Figure 82: Distribution of the nitrogen, phosphorus and potassium fertiliser forecast.

The graph in figure 83 reveals high variability in the actual demand, with frequent and significant spikes, particularly toward 2020-2023. The fitted model follows the overall trend and seasonal patterns of the data, although there are some discrepancies, especially during periods of peak demand. The fitted model effectively captures the general movement and fluctuations in demand. This close alignment indicates that the predictions are reliable. The standardised residue plots in figure 85 also confirm a small deviation of residues from the normality, confirming the reliability of the model predictions.

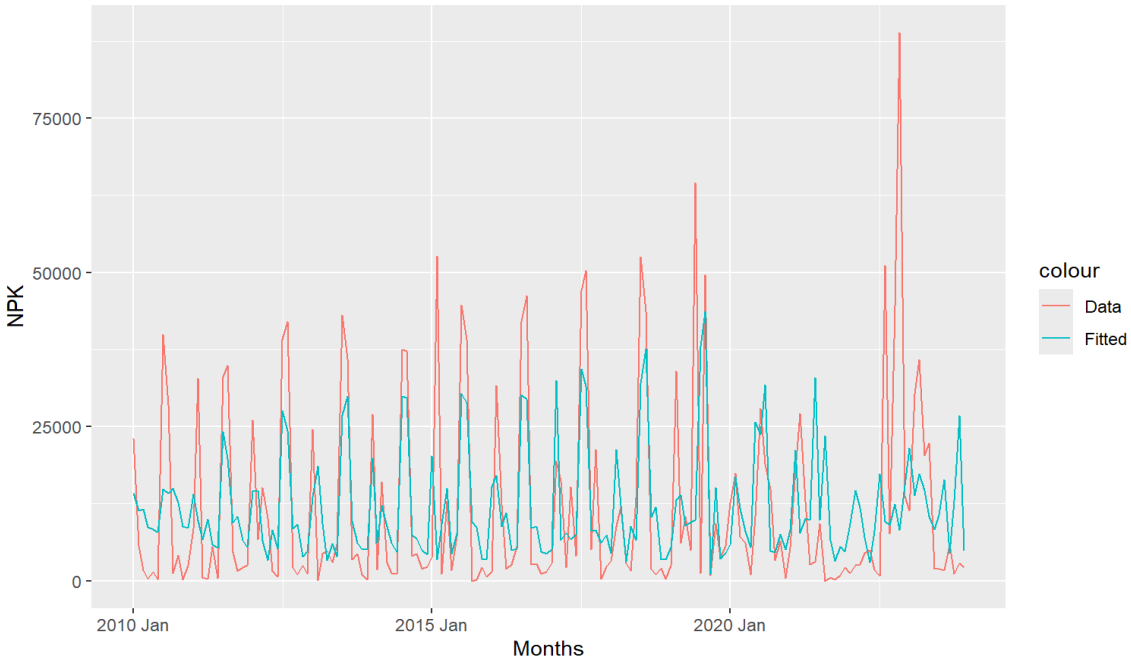


Figure 83: Fitted vs. Actual data points for nitrogen, phosphorus, and potassium fertiliser

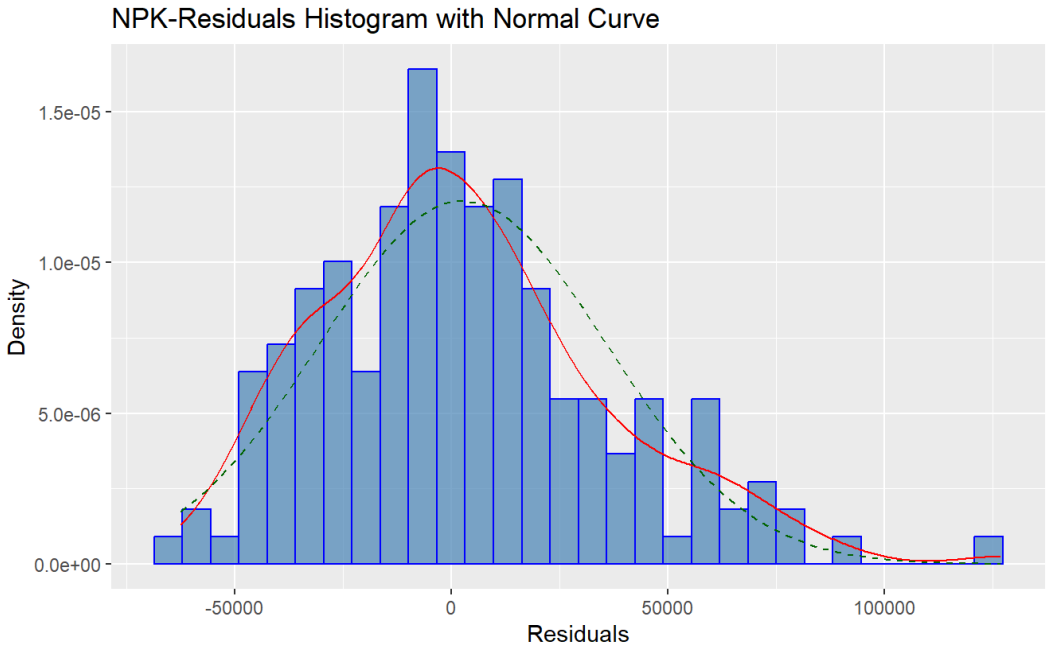


Figure 84: Residuals and normal curve for nitrogen, phosphorus, and potassium fertiliser time series

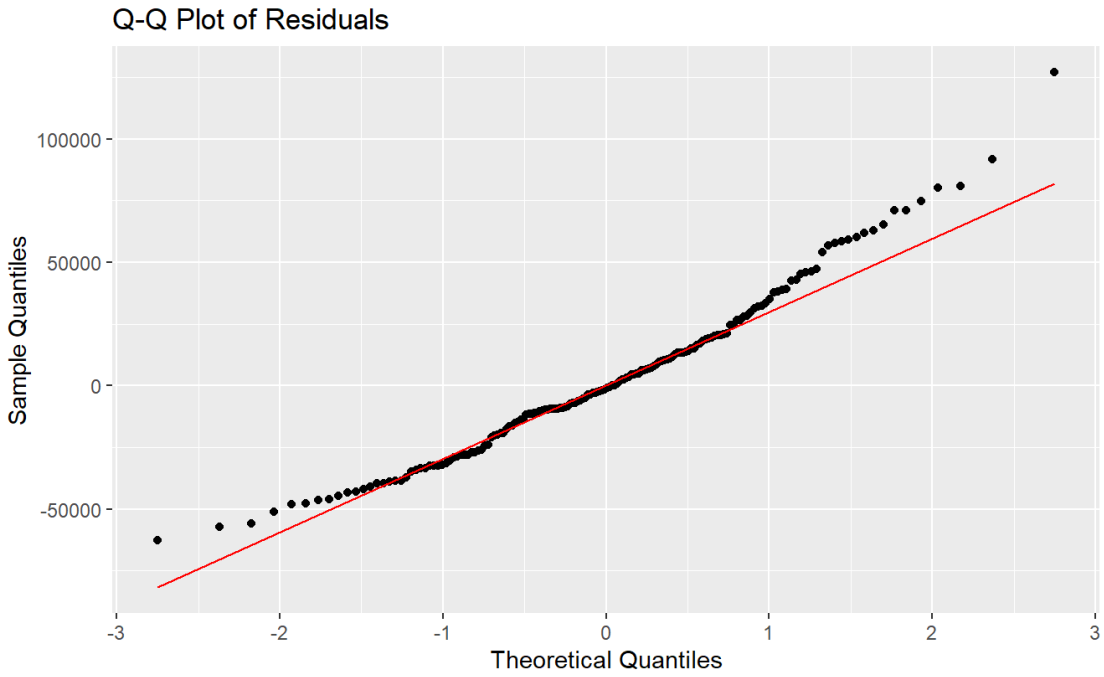


Figure 85: Standardised residual plot nitrogen, phosphorus, and potassium fertiliser

4.4.2.6 Evaluation of the Forecast Accuracy for Urea

When the plot of the forecasted values of urea was done, the forecast point appeared normally distributed (figure 86). The graph of the fitted data also indicated that the forecast was within the actual data point, demonstrating its accuracy. In order to test the normality of the residues, the Shapiro-Wilk normality test was done, and the test results were W = 0.82573, p-value = <0.0001, and W is close to 1. This is confirmation that the residues were normally distributed. Figures 88 and 89 are residual plots that confirm the normality of the residuals.

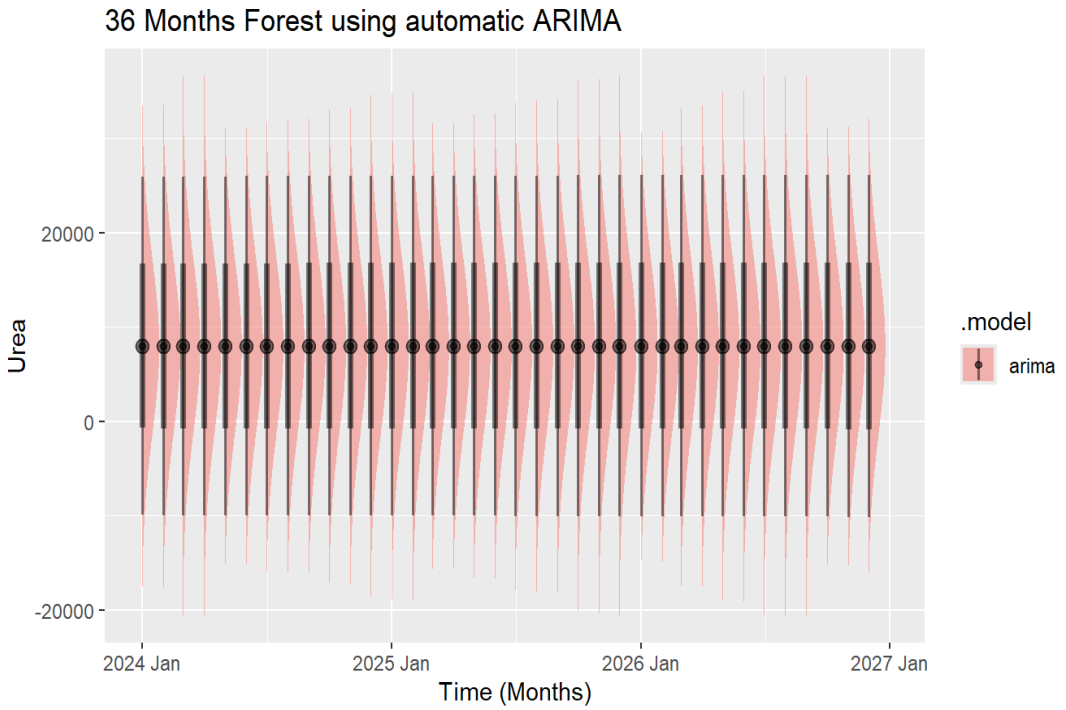


Figure 86: Distribution of the forecast for Urea.

The plot in Figure 87 demonstrates a close alignment between the actual and fitted data, confirming the forecasting model's accuracy. The fitted line, while smoother, consistently follows the general pattern of the actual data, capturing the underlying trend and the long-term movements in the observed values. The model effectively reduces the noise and short-term fluctuations visible in the data, suggesting that it is adequate in identifying the core dynamics of the series. The occasional deviations between the actual and fitted values are expected in any time series data, especially with high variability, but the overall fit suggests that the model is reliable and accurately forecasts the trend in the data. This alignment supports the model's credibility in making accurate predictions based on historical data, which is critical for decision-making and future planning.

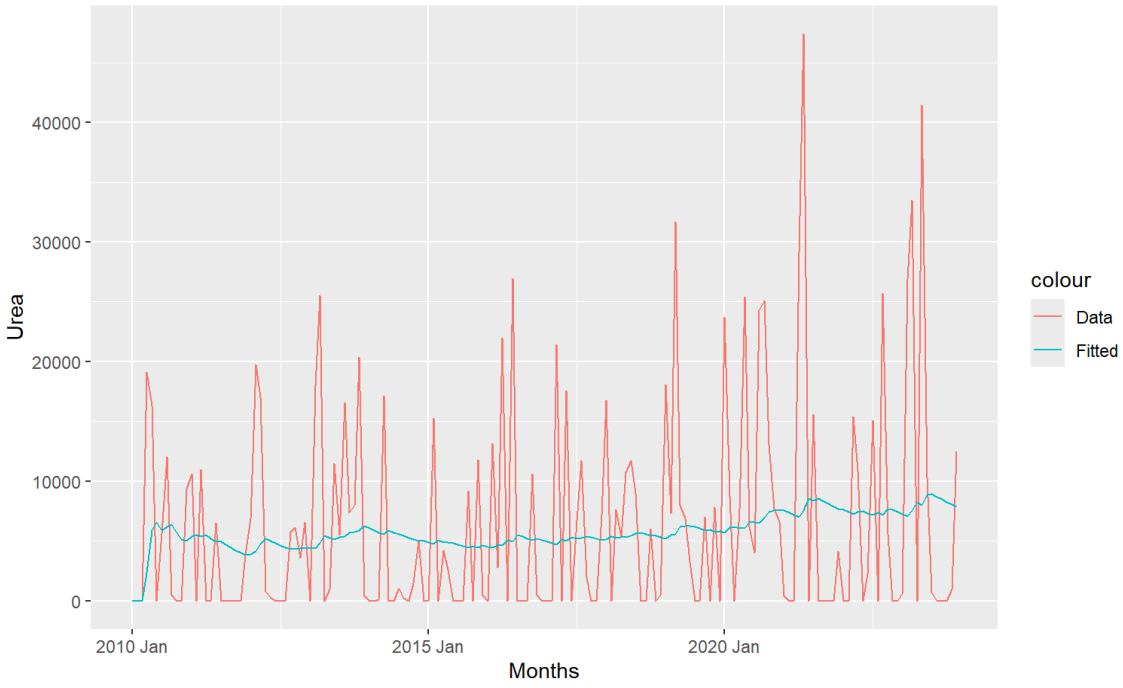


Figure 87: Fitted vs. actual data for Urea

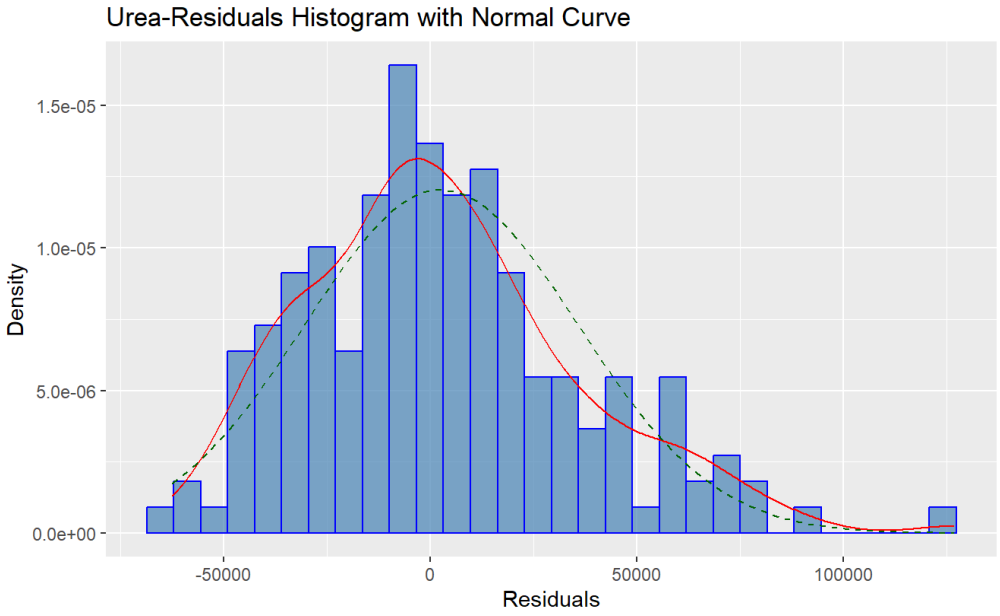


Figure 88: Histogram of residuals and normal curve for Urea time series

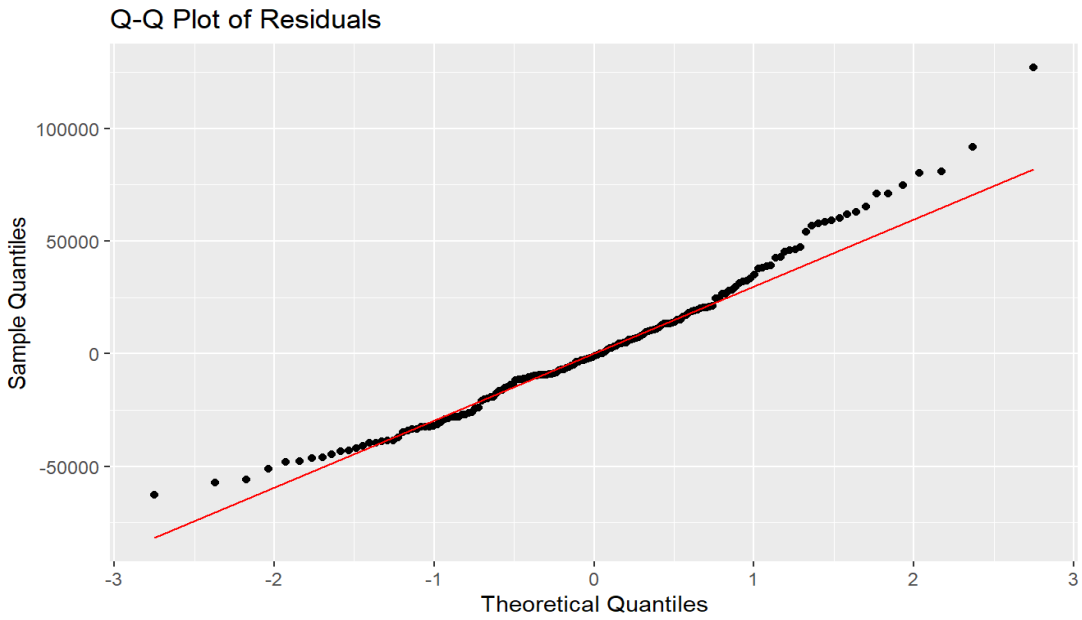


Figure 89: Standardised residuals for Urea time series

4.4.2.7 Evaluation of the Forecast Accuracy for Total Fertiliser demand

To test the normality of the forecast points of total fertiliser demand, a plot of forecast data points was done, and all the forecasted data points indicated that the forecast was normally distributed (figure 90). The plot of the fitted and the actual demand points also showed a slight difference, indicating that the fitted model correctly represented the demand for this fertiliser type. In order to test the normality of the residues, a histogram of residuals was plotted, indicating that the residuals were also normally distributed. The Shapiro-Wilk normality test also confirmed these findings, which yielded the values W = 0.97112, p-value = 0.001414. The test statistic W is close to 1, confirming the histogram results that the residues were normally distributed (Figures 92 and 93).

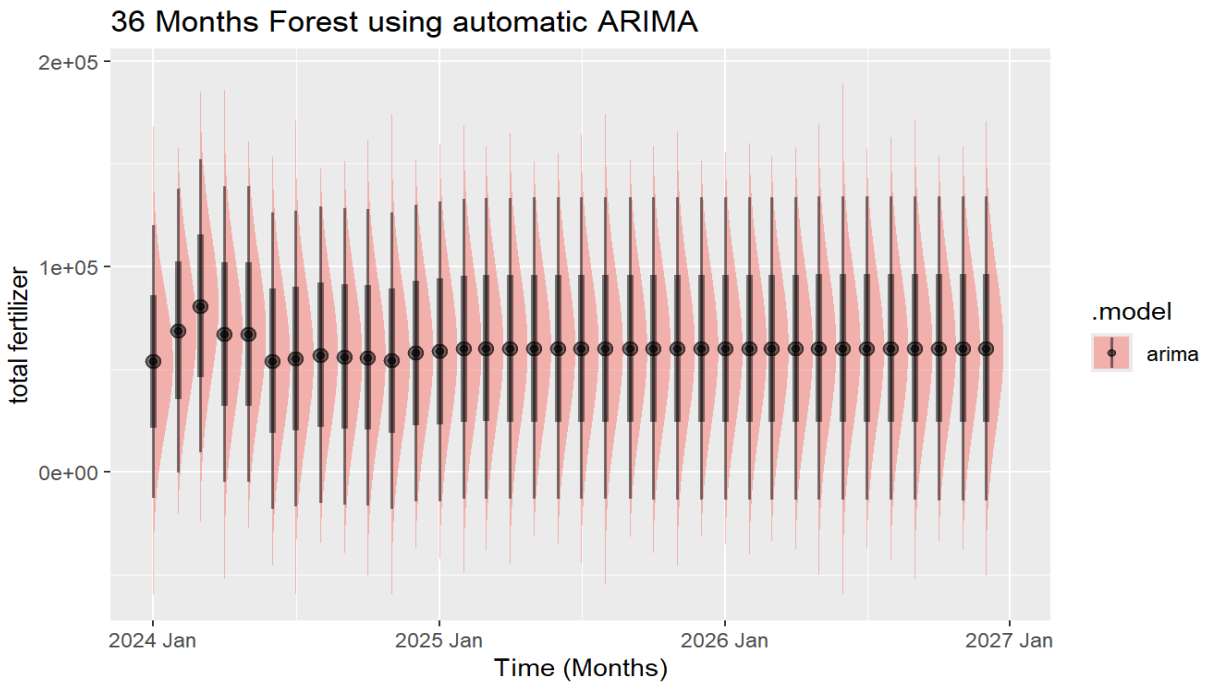


Figure 90: Distribution of the forecast total fertiliser.

The plot in Figure 91 compares the observed data (red line) and the fitted data from a forecasting model from January 2010 to December 2023. The close alignment between the actual and fitted lines suggests that the forecasting model accurately captures the overall trend and patterns in the data. While the actual data shows noticeable fluctuations and peaks indicative of periodic surges or variations in the total fertiliser, the fitted line successfully tracks these movements with relatively minor deviations. This close correspondence between the actual and fitted curves indicates that the model effectively predicts the underlying trend in the presence of short-term noise and variability (Doulah, 2018). The model's ability to follow the actual data closely, even with its volatility, confirms its reliability and accuracy in forecasting, making it a valuable tool for anticipating future values based on historical patterns.

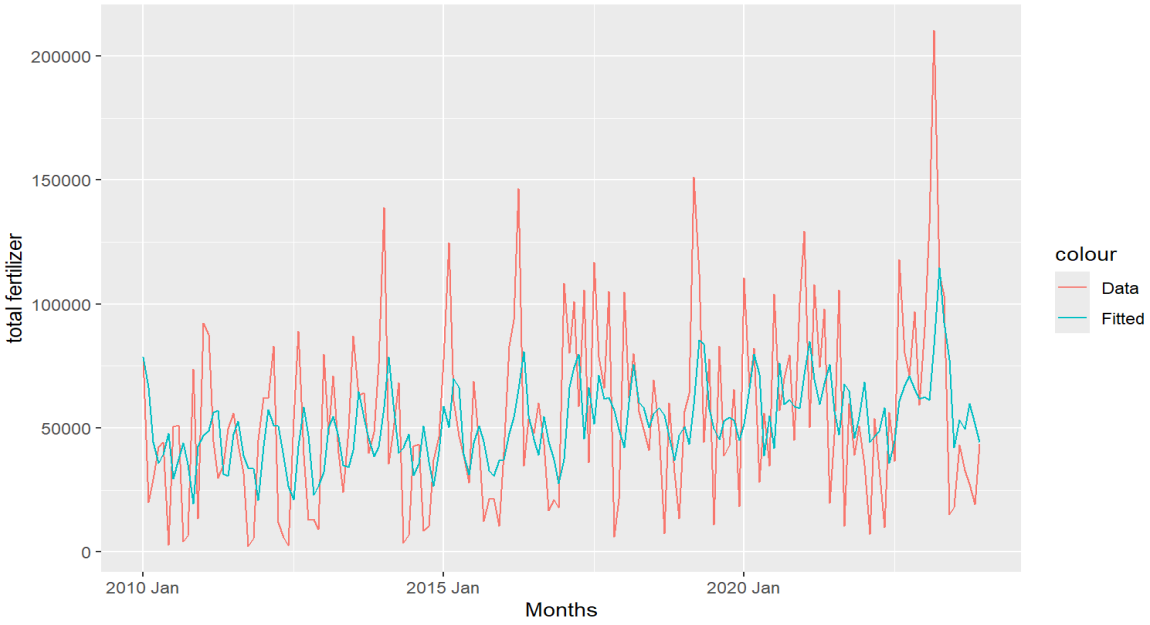


Figure 91: Plot of fitted vs. actual data for total fertiliser

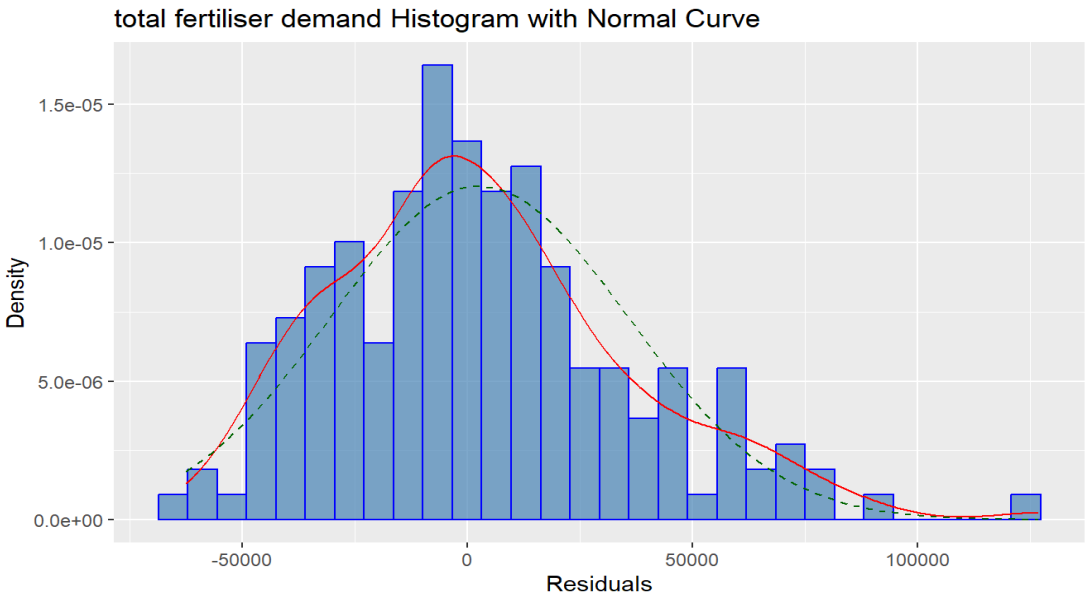


Figure 92: Histogram of residuals for total fertiliser

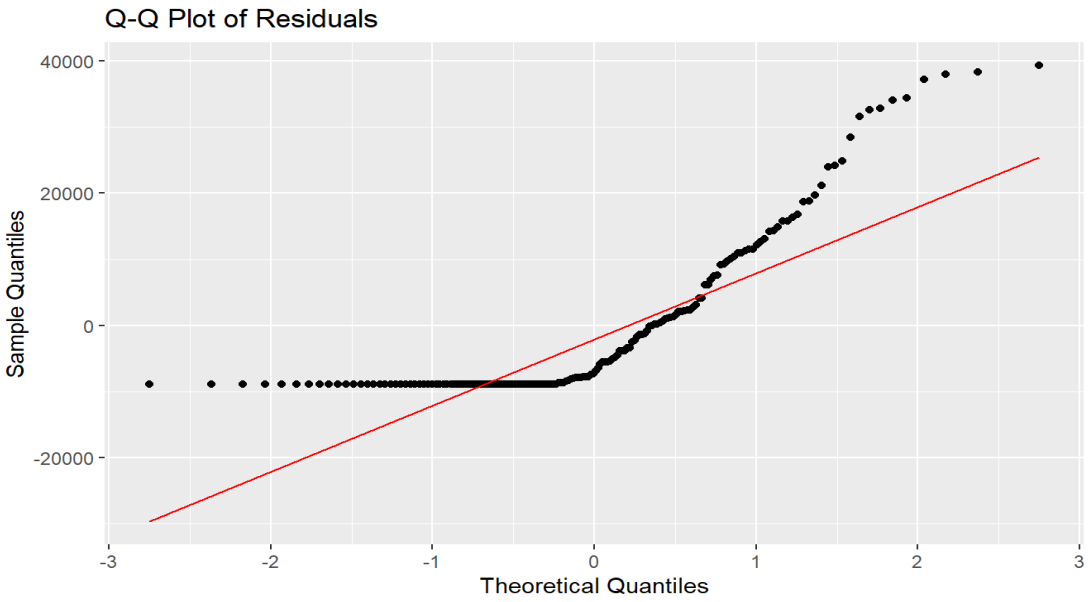


Figure 93: Standardised residuals plot for total fertiliser

All the test done to evaluate the accuracy of the model aligns with various test done. The Shapiro-Wilk normality test applied to all the fertiliser model residuals assessed the normality of residuals from our SARIMA models, following the approach of Harris (2001). Harris (2001) demonstrates the efficacy of the Shapiro-Wilk test in validating model residuals, ensuring that the assumptions of normality are met for reliable forecasts. Similarly, the application of this test confirmed that the residuals for each fertiliser type from our fertilizer demand models adhered to normal distribution assumptions, thereby reinforcing the accuracy and validity of our forecasting results. Q-Q residual plots utilized in this analysis visually assess the distribution of residuals from our models, echoing the methodology used by Harris (2001). As indicated by Harris (2001) Q-Q plots effectively reveal deviations from normality by comparing the quantiles of residuals to a theoretical normal distribution. By applying this technique, verification that the residuals from fertilizer demand forecasts approximately followed a normal distribution was ascertained, thereby supporting the robustness of our models and aligning with established practices in time series forecasting.

In this study, the distribution of forecast values is done to check if all the forecast adhered to the expected statistical properties, as done in similar research by Box *et al*. (2015). Box *et al*. (2015) study highlighted the importance of verifying forecast distributions to maintain model accuracy. By applying this approach, test on whether

the forecasted values for each type of fertilizer demand conformed to expected distributional characteristics was necessary. All the forecasts indicated normal distribution, thus validating the accuracy of our predictions and aligning with best practices in statistical forecasting.

The last test that was done was plotting a graph of standardized deviation from normality to evaluate the normality of residuals, following the procedures outlined by Muriithi *et al*. (2022). Their study demonstrated that this plot, with a reference trend line at 45 degrees, can effectively indicate the normality of residuals. Our use of this plot corroborated their findings, showing that the residuals from our forecasts (in all models) were generally aligned with the normal distribution, thereby confirming the reliability of our model results in predicting fertilizer demand.

CHAPTER FIVE
SUMMARY, CONCLUSION, AND RECOMMENDATIONS

5.1 Summary

This study addressed the urgent need to forecast fertiliser demand in Kenya, a country where agriculture plays a pivotal role in the economy and sustenance of the population. Despite the increasing population, agricultural productivity has stagnated due to factors like suboptimal fertiliser application and distribution. The knowledge gap identified was the lack of a time series model can solve the problem of inadequate fertiliser in Kenya, considering the seasonality of Kenyan agricultural patterns. The study's objectives were to analyse trends and seasonal patterns in Kenya's fertiliser demand, develop appropriate SARIMA models, and apply these models to forecast future demand. Using time series analysis, the study successfully identified patterns in fertiliser consumption and developed SARIMA models for different fertilisers using the Box Jankins approach. These models were then applied to forecast future fertiliser demand, providing a valuable tool for stakeholders in the agricultural sector to make informed decisions and improve planning processes in Kenya.

The study's key findings reveal that fertiliser demand in Kenya exhibits distinct seasonal patterns, with higher demand during specific periods due to the country’s agricultural practices. Notably, there is an increasing trend in fertiliser demand, indicating that future needs will grow as the population and agricultural activity expand. The seasonal nature of fertiliser demand is particularly pronounced during the long and short rainy seasons when agricultural activities peak and the need for fertiliser intensifies. This cyclical variation in demand underscores the importance of accounting for seasonal fluctuations in planning and supply to effectively meet the evolving demands of Kenya’s agriculture.

The analysis showed that demand for Urea and Calcium ammonium nitrate (CAN) can be explained by an ARIMA Model: ARIMA (0, 1,1) and ARIMA (0,0,0) w/mean, respectively. The two fertiliser types did not show seasonality, so they are only explained by the non-seasonal part. Other types of fertilisers had seasonal and non-seasonal parts. The SARIMA models for these types are Calcium Nitrate; SARIMA (1,1,1) (0,0,1) [12], Diammonium Phosphate; SARIMA (0,0,0) (2,0,0) [12], Muriate of

Potash; SARIMA (1,1,4) (0,0,1) [12], NPK; SARIMA (2,0,0) (2,0,0) [12]. The total demand for the fertiliser was seen to be explained by the SARIMA (1,1,4) (0,0,1) [12].

The tests curried on the forecasted demand indicated that the models accurately predicted the fertiliser demand. The Q-Q plots indicated that the residues were normally distributed. It was also confirmed by the Shapiro-Wilk normality test, which was performed on every model. The forecast covered 24 months, from August 2024 to July 2026. Extending the forecast beyond 24 months would be erroneous, as the ACF and PACF indicated short-term dependencies.

5.2 Conclusion

Forecasting is a common and essential practice for informed policy-making and effective planning. Reliable forecasts enable stakeholders to make strategic decisions, particularly in sectors sensitive to fluctuations, such as agriculture. In the context of forecasting fertiliser demand in Kenya, the Box-Jenkins methodology was used to identify the most suitable SARIMA models for various fertilisers. The analysis determined that the optimal models are as follows: for Calcium Nitrate, SARIMA (1,1,1) (0,0,1) [12]; for Diammonium Phosphate, SARIMA (0,0,0) (2,0,0) [12]; for Muriate of Potash, SARIMA (1,1,4) (0,0,1) [12]; for Nitrogen, Phosphorus, and Potassium Fertiliser, SARIMA (2,0,0) (2,0,0) [12]; for Calcium Ammonium Nitrate, ARIMA (0,0,0) with mean; for Urea, ARIMA (0,1,1); and for Total Fertiliser Demand, SARIMA (1,1,4) (0,0,1) [12].

These models offer valuable insights into the demand for various types of fertilisers. However, the forecasts extending 24 steps ahead show considerable volatility, indicating that SARIMA models are unstable for long-term predictions. This suggests that SARIMA models are more effective for short-term forecasting. Given these limitations, policymakers should explore other strategies and more robust forecasting methods for long-term planning. They should also implement suitable monetary policies to stabilize the food and beverage prices index, which can be affected by seasonal fluctuations. By addressing these issues, stakeholders can improve forecast reliability and reduce the impact of price volatility on the agricultural sector.

The ARIMA function effectively determined the most appropriate SARIMA models for each fertiliser type, efficiently capturing both seasonal and non-seasonal demand components. The models for Urea and Calcium Ammonium Nitrate (CAN) do not show seasonality, while the other types of fertilisers demonstrate significant seasonal patterns that require seasonal ARIMA components. The identified optimal SARIMA models can be used to accurately forecast fertiliser demand, thereby supporting better inventory management and supply chain planning.

5.3 Recommendation from the Study

This study is guided by the fact that fertiliser shortage has been a problem for Kenya, and therefore, the following recommendation will mitigate the problem.

i. Implement Seasonal Inventory and Distribution Planning: Given the clear seasonal patterns in fertiliser demand, particularly during the long and short rainy seasons, it is crucial for stakeholders to coordinate fertiliser supply with these periods. By adjusting stock levels to ensure adequate availability before these peak times, agricultural suppliers can avoid shortages and ensure farmers have timely access to fertilisers, thereby reducing disruptions in agricultural activities.

ii. Formulate Supply Strategies to Specific Fertiliser Types: The analysis indicates that certain fertilisers, such as Urea and Calcium Ammonium Nitrate (CAN), do not follow seasonal patterns, while others do. A targeted approach is recommended, with non-seasonal fertilisers being supplied consistently throughout the year, and fertilisers with seasonal demand (e.g., Calcium Nitrate, Diammonium Phosphate, NPK) requiring increased supply during peak agricultural periods. This strategy will optimise resource allocation and minimise both wastage and shortages.

iii. Utilise SARIMA and ARIMA Models for Forecasting: The SARIMA and ARIMA models developed in this study can be used to improve long-term forecasting of fertiliser demand for different types. By integrating these models in decision-making processes, policymakers and agricultural planners can better predict future fertiliser requirements, taking into account both population growth and increased agricultural activity. This will enable more effective

planning in terms of procurement, distribution, and pricing, ensuring that future fertiliser demand is met efficiently.

5.4 Recommendation for Further Research

i. Explore the Use of Machine Learning Models for Forecasting fertiliser demand. Future investigations on applying machine learning techniques, such as Long Short-Term Memory (LST M) networks or Gradient Boosting Machines (GBMs), for forecasting fertiliser demand need to be explored. These models can capture complex non-linear relationships and interactions that traditional time series models may not fully address.

ii. Incorporate SARIMAX and Hybrid Models for Comprehensive Forecasting. Future research should consider employing SARIMAX (Seasonal Autoregressive Integrated Moving Average with Exogenous Regressors) models alongside hybrid forecasting models. SARIMAX models can account for seasonal and non-seasonal components and external factors influencing fertiliser demand. Additionally, exploring hybrid models that combine SARIMAX with machine learning techniques, such as ensemble methods or neural networks, could offer enhanced predictive accuracy by leveraging the strengths of both statistical and data-driven approaches.

REFERENCES

Acquah, H. D. G. (2010). Comparison of Akaike information criterion (AIC) and Bayesian information criterion (BIC) in an asymmetric price relationship selection.

Akaike, H. (1974). A new look at the statistical model identification. IEEE Transactions on Automatic Control, 19(6), 716-723.

Akaike, H. (1979). Bayesian analysis of statistical models. Proceedings of the 12th Symposium on the Interface between Computer Science and Statistics, 26-31.

Aue, A., & Burman, P. (2023). Estimation of prediction error in time series. *Biometrika*. https://doi.org/10.1093/biomet/asad053

Bal, S. K., Manikandan, N., Sandeep, V. M., Kumar, P. V., Lunagaria, M. M., Rao, A. S., ... & Singh, V. K. (2022). Criteria based decisions for determining agroclimatic onset of the crop growing season. Agricultural and Forest Meteorology, 317, 108903.

Barnett, V., & Lewis, T. (1994). Outliers in statistical data. Wiley

Baum, C. F. (2000). sts15: Tests for stationarity of a time series. Stata Technical Bulletin, 57, 36-39.

Bezerra, E, Ogasawara, E.S., Oliveira, D.D., Junior, F.P., Castañeda, R., Amorim, M., Mauro, R.C., Soares, J.D., Quadros, J.R., &. (2013). A Forecasting Method for Fertilisers Consumption in Brazil. *Int. J. Agric. Environ. Inf. Syst., 4*, 23-36.

Borkar, P. (2023). Statistical Modeling for Forecasting Fertiliser Consumption in India. https://orcid.org/0000-0002-5437-7001

Box, G. E. P., & Jenkins, G. M. (1976). *Time Series Analysis: Forecasting and Control*. San Francisco: Holden-Day.

Box, G. E. P., Jenkins, G. M., & Reinsel, G. C. (2015). Time Series Analysis: Forecasting and Control (5th ed.). Wiley.

Brockwell, P. J., & Davis, R. A. (2016). Introduction to Time Series and Forecasting (3rd ed.). Springer.

Carnero, M. A., Pena, D., & Ruiz, E. (2007). Effects of outliers on the identification and estimation of GARCH models. *Journal of time series analysis*, *28*(4), 471-497.

Cavanaugh, J. E., & Neath, A. A. (2019). The Akaike information criterion: Background, derivation, properties, application, interpretation, and refinements. *Wiley Interdisciplinary Reviews: Computational Statistics*, *11*(3), e1460.

Chen, C. (2017). Outlier Detection with Applications in R. CRC Press.

Chen, J., Wei, Y., & Liu, Q. (2015). On the application of model selection criteria to time series models. Statistical Modelling, 15(4), 445-470.

Cryer, J. D., & Kellet, N. (1991). *Time series analysis*. Royal Victorian Institute for the Blind. Tertiary Resource Service.

Danielle C. M. Ristow, Elisa Henning, Andreza Kalbusch, Cesar E. Petersen (2021). Models for forecasting water demand using time series analysis: a case study in Southern Brazil. Journal of Water, Sanitation and Hygiene for Development; 11 (2): 231–240. doi: https://doi.org/10.2166/washdev.2021.208

El Gayar, A. (2021). A study on Nutrients in sustainable cropping systems. https://kosmospublishers.com/wp-content/uploads/2021/03/A-study-on-Nutrients-in-Sustainable-Cropping-Systems.pdf

Filzmoser, P., Maronna, R., & Werner, M. (2008). Outlier identification in high dimensions. Computational Statistics & Data Analysis, 52(3), 1694-1711.

Franses, P. H., & Ghiassi, M. (2009). A seasonal periodic autoregressive time series model. Journal of Time Series Analysis, 30(1), 53-68. doi:10.1111/j.1467-9892.2008.00596.x

Gneiting, T., & Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. Journal of the American Statistical Association, 102(477), 359-378.

Hannah Ritchie, Max Roser and Pablo Rosado (2022) - "Fertilisers". Published online at OurWorldInData.org. Retrieved from: 'https://ourworldindata.org/fertilisers' [Online Resource]

Harris, D. (2001). Statistical Modeling in S-Plus. Springer.

Hyndman, R.J., & Athanasopoulos, G. (2018). Forecasting: Principles and Practice. OTexts. Available at: https://otexts.com/fpp3/

Ihaka, R. (2005). Time Series Analysis: lecture notes for 475.726. Erişim adresi https://www. stat. auckland. ac. nz/~ ihaka/726/notes. pdf.

Jarque, C. M., & Bera, A. K. (1987). A test for normality of observations and regression residuals. International Statistical Review, 55(2), 163-172.

Joshua Ariga,, J., Jayne, T. S., & Nyoro, J. (2006). Factors Driving the Growth in Fertilizer Consumption in Kenya, 1990-2005: Sustaining the Momentum in Kenya and Lessons for Broader Replicability in Sub-Saharan Africa. https://pdf.usaid.gov/pdf_docs/Pnadh710.pdf

Kim, T. H., & White, H. (2004). Estimation, testing, and specification of moments structures in generalized method of moments models. Journal of Econometrics, 117(2), 379-399.

Kirchgässner, G., Wolters, J., & Hassler, U. (2012). *Introduction to modern time series analysis*. Springer Science & Business Media.

Ljung, G. M., & Box, G. E. (1978). On a measure of lack of fit in time series models. *Biometrika*, *65*(2), 297-303.

Lütkepohl, H. (2006). New Introduction to Multiple Time Series Analysis. Springer.

Manigandan P, Alam MS, Alharthi M, Khan U, Alagirisamy K, Pachiyappan D, Rehman A. (2021). Forecasting Natural Gas Production and Consumption in United States-Evidence from SARIMA and SARIMAX Models. Energies.; 14(19):6021. https://doi.org/10.3390/en14196021

Mathenge, M. K. (2009). Fertilizer Types, Availability and Consumption In Kenya. Tegemeo Institute, Egerton University. https://statistics.kilimo.go.ke/files/bookpage/Fertilizer_Use_Kenya_Data-Tegemeo.pdf

McQuarrie, A. D., & Tsai, C. L. (1998). Regression and time series model selection. World Scientific.

Ministry of Agriculture and Livestock Development. (2022, September 20). *Fertilizer subsidy 2022*. https://kilimo.go.ke/fertilizer-subsidy-2022

Mose, O. L. (2019). Factors affecting the distribution and use of fertiliser in Kenya: Preliminary assessment. Egerton University Institutional Repository. Retrieved June 17, 2024, from https://41.89.96.81:8080/xmlui/handle/123456789/2372

Mutegi, J., Adolwa, I., Kiwia, A., Njoroge, S., Gitonga, A., Muthamia, J., & Kansiime, M. (2024). Agricultural production and food security implications of Covid-19 disruption on small-scale farmer households: lessons from Kenya. World Development, 173, 106405.

NCBP. (2022). *Mitigating High Fertiliser Prices For The 2022 Long And Short Rains*. Ministry Of Agriculture, Livestock, Fisheries & Cooperatives Press Statement. https://ncpb.co.ke/wp-content/uploads/2022/05/CS-Press-Statement-Fertiliser-Support-Interventio_220401_134136.pdf

Nchanji, E. B., & Lutomia, C. K. (2021). Regional impact of COVID-19 on the production and food security of common bean smallholder farmers in Sub-Saharan Africa: Implication for SDG's. *Global Food Security*, *29*, 100524.

Njagi, T., Riungu, C., Opiyo, J., Mwadime, R. K., & Aloo, S. Y. (2024). Assessment of the impact of the Kenya Government fertiliser subsidy on the performance of domestic private sector fertiliser trade.

Okello, E. A. (2023). Application of Hybrid seasonal ARIMA-GARCH Model in modelling and forecasting fertiliser prices in Kenya [Strathmore University]. http://hdl.handle.net/11071/15390

Omega, G., & Omega, M. (2024). A Green Revolution in Kenya's Arid and Semi-Arid Lands: The Fertilizer Solution. kippra.or.ke. https://repository.kippra.or.ke/bitstream/handle/123456789/4636/PB12%202023-2024%20Fertilizer%20Use.pdf?sequence=1&isAllowed=y

Osborn, D. R. (2018). Periodic autoregressive models. In Handbook of Economic Forecasting (Vol. 2, pp. 227-285). Elsevier.

Otu, O. A., Osuji, G. A., Opara, J., Mbachu, H. I., & Iheagwara, A. I. (2014). Application of Sarima models in modeling and forecasting Nigeria's inflation rates. *American Journal of Applied Mathematics and Statistics*, *2*(1), 16-28.

Pennstate college. (2018). *11.5 - Information criteria and PRESS*. Pennstate Eberly college of science. https://online.stat.psu.edu/stat462/node/199/

Pisuttinusart, C., Jatuporn, C., Suvanvihok, V., & Seerasarn, N. (2022). Forecasting The import Demand for chemical fertiliser in Thailand. *The EUrASEANs: journal on global socio-economic dynamics*, (3 (34)), 61-70.

Rural Resilience Programme. (2018, August). Ministry of Agriculture and Livestock Development. https://kilimo.go.ke/wp-content/uploads/2021/02/Field-Guide-6-Growing-Rainfed-Crops-in-Dryland-Zones-Cereals-and-Legumes_reduce.pdf

Saxena, K. K., & Kamnge, J. S. (2020). Comparative study of exponential smoothing models and Box-Jenkins ARIMA model of partitioned data of daily stock prices of the CRDB Bank in Tanzania. https://www.indianjournals.com/ijor.aspx?target=ijor:bpasms&volume=39e&issue=1&article=001

Shapiro, S. S., & Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). Biometrika, 52(3/4), 591-611.

Sheahan, M., Ariga, J., & Jayne, T. S. (2016). Modeling the effects of input market reforms on fertiliser demand and maize production: A case study from Kenya. *Journal of Agricultural Economics*, *67*(2), 420-447.

Shumway, R. H., & Stoffer, D. S. (2017). Time Series Analysis and Its Applications: With R Examples (4th Edition). Springer.

Shumway, R. H., Stoffer, D. S., & Stoffer, D. S. (2000). *Time series analysis and its applications* (Vol. 3). New York: springer.

Shumway, R. H., Stoffer, D. S., Shumway, R. H., & Stoffer, D. S. (2017). ARIMA models. Time series analysis and its applications: with R examples, 75-163.

Shumway, R.H., & Stoffer, D.S. (2017). Time Series Analysis and Its Applications: With R Examples. Springer.

Statista. (2021, July 8). Kenya: Agricultural use of fertilisers by type. Statista. https://www.statista.com/statistics/1289850/agricultural-use-of-fertilisers-in-kenya-by-type/

Statista. (2024, January 20). *Global fertiliser consumption by nutrient 1965-2021*. https://www.statista.com/statistics/438967/fertiliser-consumption-globally-by-nutrient/

Sukprasertb, P., Tongchurec, S., Suvanvihok, V., & Thongkaew, S. (2020). Forecasting import demand of table grapes: Empirical evidence from Thailand. Asian Journal of Agriculture and Rural Development, 10(2), 578-586.

Taylor, M. D., & Locascio, S. J. (2004). Blossom-end rot: a calcium deficiency. *Journal of Plant nutrition*, *27*(1), 123-139.

Tenkorang, Frank & Lowenberg-DeBoer, James. (2008). Forecasting Long-Term Global Fertiliser Demand. Nutrient Cycling in Agroecosystems. 83. 233-247. 10.1007/s10705-008-9214-y.

The World Bank. (2018, April 11). *Poverty incidence in Kenya declined significantly, but unlikely to be eradicated by 2030*. World Bank. https://www.worldbank.org/en/country/kenya/publication/kenya-economic-update-poverty-incidence-in-kenya-declined-significantly-but-unlikely-to-be-eradicated-by-2030

Wanjuki, T. M., Wagala, A. and Muriithi, D. K. (2021). Sarima models: review and its application to Kenyan's commodity price index of food and beverage. In*: Isutsa, D. K. (Ed.). Proceedings of the 7th International Research Conference held in Chuka University from 3rd to 4th December 2020, Chuka, Kenya, p. 574-586*

World Bank Open Data. (2022). World Bank Open Data. Retrieved January 6, 2024, from https://data.worldbank.org/indicator/AG.LND.ARBL.ZS?locations=KE

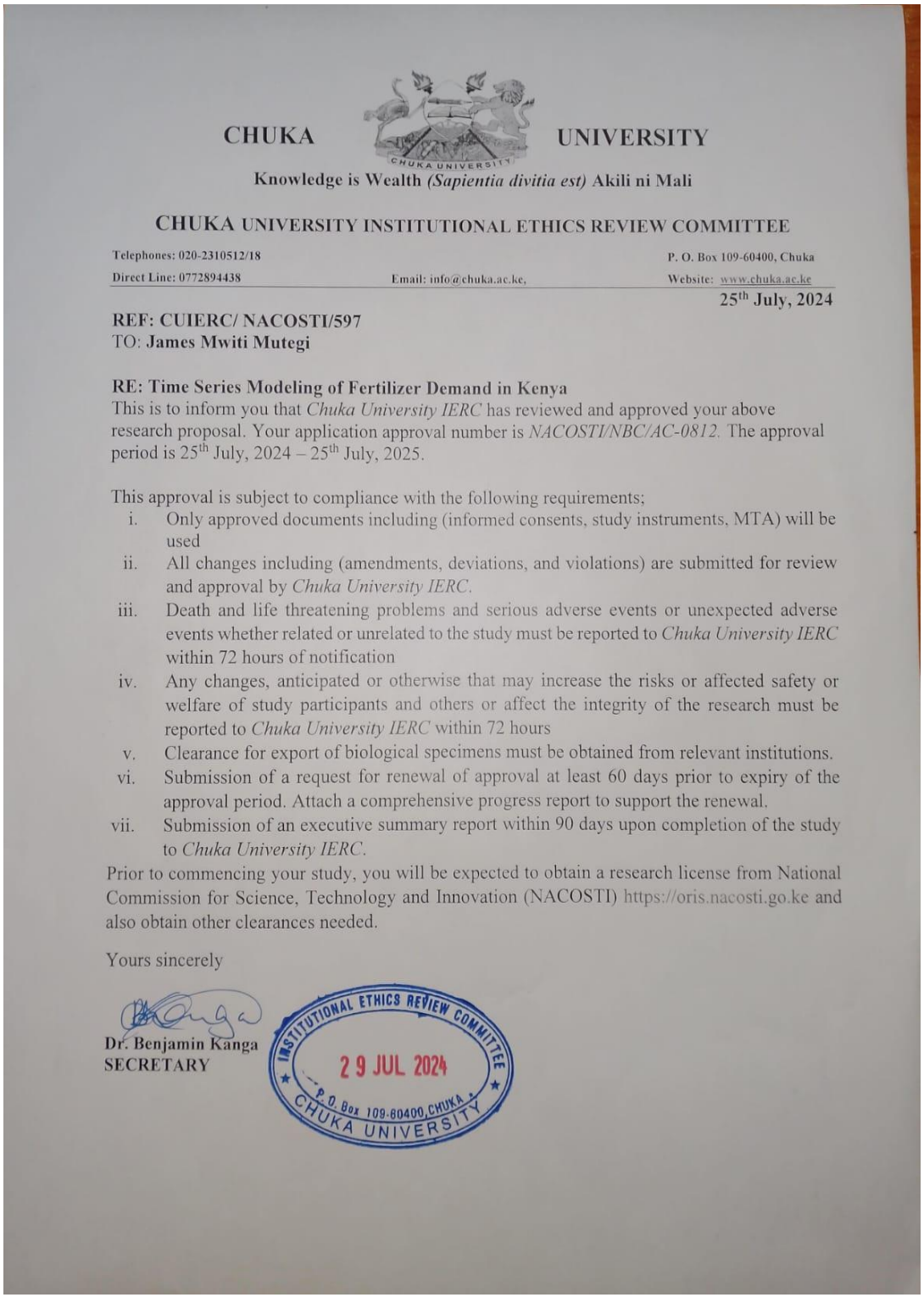
Yaseen, M., Ahmad, A., Younas, N., Naveed, M., Ali, M. A., Shah, S. S. H., ... & Mustafa, A. (2023). Value-Added Fertilisers Enhanced Growth, Yield and Nutrient Use Efficiency through Reduced Ammonia Volatilization Losses under Maize–Rice Cropping Cultivation. Sustainability, 15(3), 2021.

Zellner, M., Abbas, A. E., Budescu, D. V., & Galstyan, A. (2021). A survey of human judgement and quantitative forecasting methods. *Royal Society open science, 8(2), 201187.*

Zheng, W., Liu, Z., Zhang, M., Shi, Y., Zhu, Q., Sun, Y., ... & Geng, J. (2017). Improving crop yields, nitrogen use efficiencies, and profits by using mixtures of coated controlled-released and uncoated urea in a wheat-maize system. *Field Crops Research*, *205*, 106-115.

APPENDICES

Appendix 1: Chuka university Research Permit



CHUKA UNIVERSITY

Knowledge is Wealth *(Sapientia divitia est)* Akili ni Mali

CHUKA UNIVERSITY INSTITUTIONAL ETHICS REVIEW COMMITTEE

Telephones: 020-2310512/18 | P. O. Box 109-60400, Chuka
Direct Line: 0772894438 | Email: info@chuka.ac.ke, | Website: www.chuka.ac.ke

25th July, 2024

REF: CUIERC/ NACOSTI/597
TO: **James Mwiti Mutegi**

RE: Time Series Modeling of Fertilizer Demand in Kenya
This is to inform you that *Chuka University IERC* has reviewed and approved your above research proposal. Your application approval number is *NACOSTI/NBC/AC-0812*. The approval period is 25th July, 2024 – 25th July, 2025.

This approval is subject to compliance with the following requirements;

i. Only approved documents including (informed consents, study instruments, MTA) will be used
ii. All changes including (amendments, deviations, and violations) are submitted for review and approval by *Chuka University IERC*.
iii. Death and life threatening problems and serious adverse events or unexpected adverse events whether related or unrelated to the study must be reported to *Chuka University IERC* within 72 hours of notification
iv. Any changes, anticipated or otherwise that may increase the risks or affected safety or welfare of study participants and others or affect the integrity of the research must be reported to *Chuka University IERC* within 72 hours
v. Clearance for export of biological specimens must be obtained from relevant institutions.
vi. Submission of a request for renewal of approval at least 60 days prior to expiry of the approval period. Attach a comprehensive progress report to support the renewal.
vii. Submission of an executive summary report within 90 days upon completion of the study to *Chuka University IERC*.

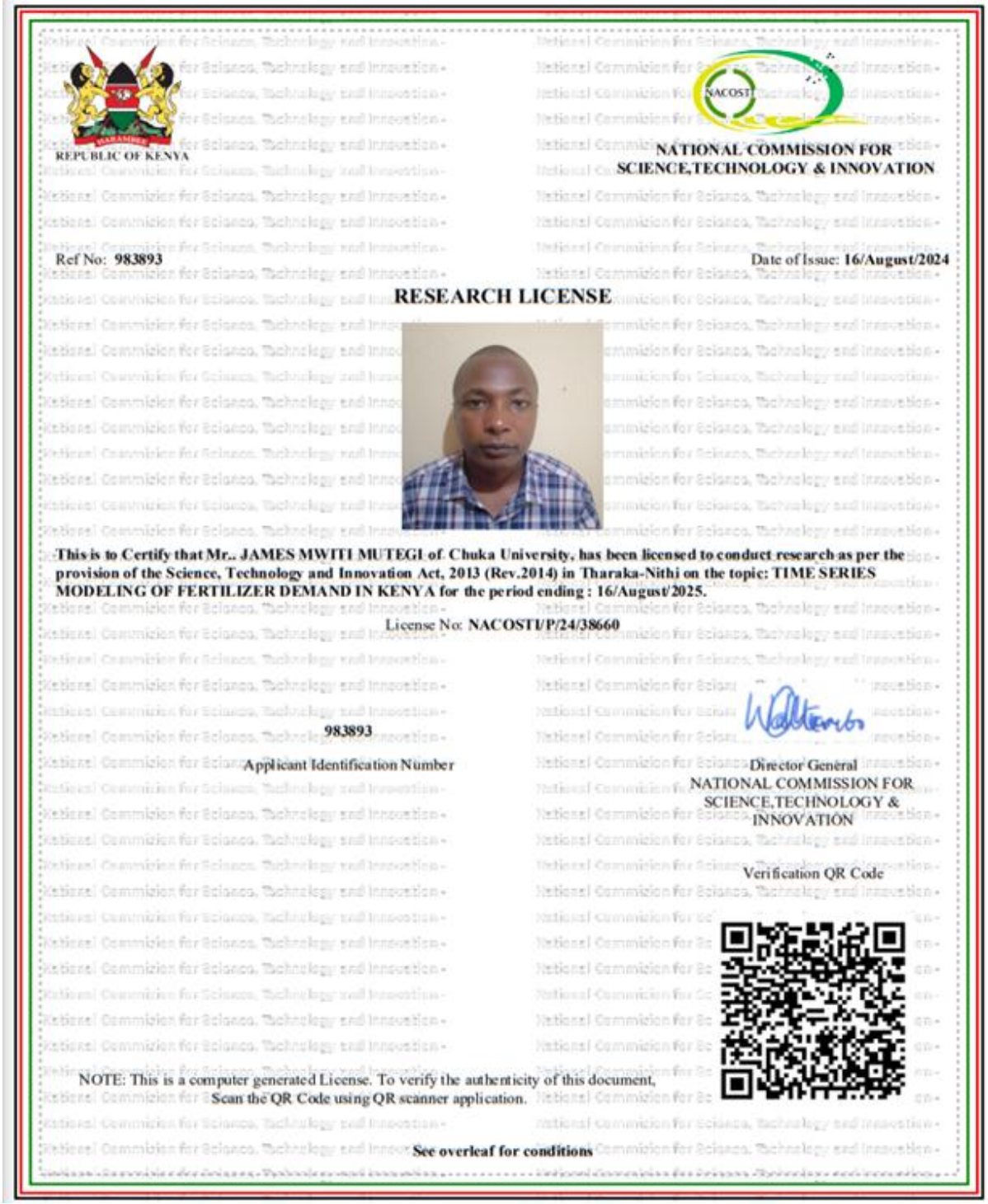
Prior to commencing your study, you will be expected to obtain a research license from National Commission for Science, Technology and Innovation (NACOSTI) https://oris.nacosti.go.ke and also obtain other clearances needed.

Yours sincerely

Dr. Benjamin Kanga
SECRETARY

INSTITUTIONAL ETHICS REVIEW COMMITTEE
29 JUL 2024
P.O. Box 109-60400, CHUKA
CHUKA UNIVERSITY

Appendix 2: National Commission for Science, Technology & Innovation Permit



REPUBLIC OF KENYA

NATIONAL COMMISSION FOR SCIENCE,TECHNOLOGY & INNOVATION

Ref No: **983893**

Date of Issue: **16/August/2024**

RESEARCH LICENSE

This is to Certify that Mr.. JAMES MWITI MUTEGI of Chuka University, has been licensed to conduct research as per the provision of the Science, Technology and Innovation Act, 2013 (Rev.2014) in Tharaka-Nithi on the topic: TIME SERIES MODELING OF FERTILIZER DEMAND IN KENYA for the period ending : 16/August/2025.

License No: **NACOSTI/P/24/38660**

983893

Applicant Identification Number

Director General
NATIONAL COMMISSION FOR SCIENCE,TECHNOLOGY & INNOVATION

Verification QR Code

NOTE: This is a computer generated License. To verify the authenticity of this document, Scan the QR Code using QR scanner application.

See overleaf for conditions

Appendix 3: R- code

Load the following libraries

```
library(fpp3)
library(tibble)
library(tsibble)
library(tidyverse)
library(tsibbledata)
library(fable)
```

Data Importation

```
Fertiliser <- read_csv("Fertiliser.csv")
Fertiliser$others <- NULL
Fertiliser
# A tibble: 168 × 8
   Months    `Calcium Ammonium Nitrate` `Calcium Nitrate` `Diammonium Phosphate`
   <chr>                          <dbl>            <dbl>                  <dbl>
 1 1/1/2010                        5000             3644                  35594
 2 2/1/2010                           0                0                   2535
 3 3/1/2010                        1404                0                  25036
 4 4/1/2010                           0                0                  20436
 5 5/1/2010                       21500                0                      0
 6 6/1/2010                           0                0                    319
 7 7/1/2010                        5000                0                      0
 8 8/1/2010                        7481                0                      0
 9 9/1/2010                         552.               0                    475.
10 10/1/2010                          0              675                    300
# □ 158 more rows
# □ 4 more variables: `Muriate of Potash` <dbl>, NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER
<dbl>, Urea <dbl>,
#   `total fertiliser` <dbl>
library(kableExtra)
library(psych)
describeBy(Fertiliser[,-1])
```

Index by year (Declaring the Data Time Series)

```
Fertiliser <- Fertiliser |>
  mutate(Months = yearmonth(Months)) |>
  as_tsibble(index = Months)
Fertiliser
# A tsibble: 168 x 8 [1M]
```

```{r}
years <- 2010:2023
colors <- c("#FF4500",  # 2010 - Orange Red
  "#FFA500",  # 2011 - Orange
  "#FFD700",  # 2012 - Gold
  "#ADFF2F",  # 2013 - Green Yellow
  "#32CD32",  # 2014 - Lime Green
  "#00FA9A",  # 2015 - Medium Spring Green
  "#00FFFF",  # 2016 - Cyan / Aqua
  "#87CEEB",  # 2017 - Sky Blue
  "#0000FF",  # 2018 - Blue
  "#8A2BE2",  # 2019 - Blue Violet
  "#FF00FF",  # 2020 - Magenta / Fuchsia
  "#FF69B4",  # 2021 - Hot Pink
  "#FF1493",  # 2022 - Deep Pink
  "#FFB6C1"   # 2023 - Light Pink
)
color_key <- data.frame(Year = years, Color = colors)
print(color_key)

```
```

Calcium Ammonium Nitrate

```
Fertiliser|>
  autoplot(`Calcium Ammonium Nitrate`)+
  xlab("Months") +
```
```

```
  ylab("Fertiliser (CAN)")+
   ggtitle("Time Series Plot of Calcium Ammnonium Nitrate Over Time [2010 - 2023")
Normality test
library(tseries)
jarque.bera.test(Fertiliser$`Calcium Ammonium Nitrate`)

    Jarque Bera Test

data:  Fertiliser$`Calcium Ammonium Nitrate`
X-squared = 82.267, df = 2, p-value < 2.2e-16

Plot a gg_season plot
Fertiliser |>
  gg_season(`Calcium Ammonium Nitrate`)+
  ggtitle("Time Series Plot (GG-Season plot) of CAN Over Time [2010 - 2023")
Seasonal Subseries Plot
Fertiliser |>
  gg_subseries(`Calcium Ammonium Nitrate`)+
  ggtitle("Time Series Plot (GG-Sub series plot) of Total Fertiliser Over Time [2010 - 2023")
dcmp <- Fertiliser |>
  model(STL(`Calcium Ammonium Nitrate`))
dcmp
# A mable: 1 x 1
  `STL(\`Calcium Ammonium Nitrate\`)`

Plot the Components
dcmp |>
  components() |>
  autoplot(`Calcium Ammonium Nitrate`)

Extract the Components using Component Function
dcmp |>
  components()
Seasonality of the season_year
dcmp |>
  components() |>
  gg_season(season  year)+
  ggtitle("Seasonal Variation in CAN Fertiliser")
Seasonal Sub Series Plot
dcmp |>
  components() |>
  gg_subseries(season_year)+
  ggtitle("Time Series Plot (GG-Sub series for season year) for CAN Fertiliser [2010 - 2023")
```

Seasonal Difference

```
Fertiliser|>
  autoplot(`Calcium Ammonium Nitrate`|>
           difference(12))+
  xlab("Year") +
  ylab("CAN")+
   ggtitle("Time Series Plot of Stationarity CAN Fertiliser")
tabilize the Variance by Taking the Normal Difference
Fertiliser |>
  autoplot(log(`Calcium Ammonium Nitrate`) |>
           #difference(12) |>
           difference(1))+
  ggtitle("Seasonal and First Difference to Stabalize the Variance")
```

Stationarity test (kpss function)

```
URTST1<- Fertiliser |>
  features(`Calcium Ammonium Nitrate`, unitroot_kpss)
URTST1
```

```
# A tibble: 1 × 2
  kpss_stat kpss_pvalue
      <dbl>       <dbl>
1     0.182         0.1
```

Null and Alternative hypothesis

Null Hypotheses

The time series is stationary

Alternative HYpothesis

The time series is not stationary

From the results above, the times series is not stationary since the p-value is less than 0.05.

Test How many Differencing we need to make the series stationary

```
Fertiliser |>
  features(`Calcium Ammonium Nitrate`, unitroot_ndiffs)
# A tibble: 1 × 1
  ndiffs
   <int>
1      0
Fertiliser |>
  features(`Calcium Ammonium Nitrate`, unitroot_nsdiffs)
# A tibble: 1 × 1
  nsdiffs
    <int>
1       0
```

Autocorrelation Function (ACF)

```
Fertiliser |>
  ACF(`Calcium Ammonium Nitrate`) |>
  autoplot()+
  labs(title = "ACF Plot for CAN", x = "Time (Months)", y = "ACF")
```

Parctial Autocorrelation Function

```
Fertiliser |>
  PACF(`Calcium Ammonium Nitrate`) |>
  autoplot()+
  labs(title = "PACF Plot for CAN", x = "Time (Months)", y = "PACF")
```

CALCIUM NITRATE

```
Fertiliser|>
  autoplot(`Calcium Nitrate`)+
  xlab("Months") +
  ylab("Fertiliser (CN)")+
   ggtitle("Time Series Plot of Calcium Nitrate Over Time [2010 - 2023")
```

Normality test

```
library(tseries)
jarque.bera.test(Fertiliser$`Calcium Nitrate`)


    Jarque Bera Test


data:  Fertiliser$`Calcium Nitrate`
X-squared = 38584, df = 2, p-value < 2.2e-16
```

Plot a gg_season plot

```
Fertiliser |>
  gg_season(`Calcium Nitrate`)+
  ggtitle("Time Series Plot (GG-Season plot) of Calcium Nitrate Fertiliser Over Time [2010 - 2023")
```

Seasonal Subseries Plot

```
Fertiliser |>
  gg_subseries(`Calcium Nitrate`)+
  ggtitle("Time Series Plot (GG-Sub series plot) of Calcium Nitrate Fertiliser Over Time [2010 - 2023")
```

```
dcmp1 <- Fertiliser |>
  model(STL(`Calcium Nitrate`))
dcmp1
# A mable: 1 x 1
  `STL(\`Calcium Nitrate\`)`
                     <model>
1                      <STL>
```

Plot the Components

```
dcmp1 |>
  components() |>
  autoplot(`Calcium Nitrate`)
```

Extract the Components using Component Function

```
dcmp1 |>
  components()
```

Seasonality of the season_year

```
dcmp1 |>
  components() |>
  gg_season(season_year)+
  ggtitle("Seasonal Variation in Calcium Nitrate Fertiliser")
```

Seasonal Sub Series Plot

```
dcmp1 |>
  components() |>
  gg_subseries(season  year)+
  ggtitle("Time Series Plot (GG-Sub series for season year) for Calcium Nitrate Fertiliser [2010 - 2023")
```

Difference the Series

Here we use the function, differencing as shown below

Seasonal Difference

```
Fertiliser|>
  autoplot(`Calcium Nitrate`|>
             difference(12))+
  xlab("Year") +
  ylab("Calcium Nitrate")+
   ggtitle("Time Series Plot of Stationary Calcium Nitrate Fertiliser")
```

Stabilize the Variance by Taking the Normal Difference

Note!!

It is recommended that you take the seasonal difference (quarter, m = 4, monthly, m = 12 and so on) and then take the first difference.

```
Fertiliser |>
  autoplot(log(`Calcium Nitrate`)|>
             difference(1))+
  ggtitle("Seasonal and First Difference to Stabalize the Variance")
```

Stationarity test (kpss function)

```
URTST1<- Fertiliser |>
  features(`Calcium Nitrate`, unitroot_kpss)
URTST1
# A tibble: 1 × 2
  kpss_stat kpss_pvalue
      <dbl>       <dbl>
1     0.792        0.01
```

Null and Alternative hypothesis

Null Hypotheses

The time series is stationary

Alternative HYpothesis

The time series is not stationary

Test How many Differencing we need to make the series stationary

```
Fertiliser |>
  features(`Calcium Nitrate`, unitroot_ndiffs)
# A tibble: 1 × 1
  ndiffs
```

```
  <int>
1      1
Fertiliser |>
  features(`Calcium Nitrate`, unitroot_nsdiffs)
# A tibble: 1 × 1
  nsdiffs
    <int>
1       0
```

Autocorrelation Function (ACF)

```
Fertiliser |>
  ACF(`Calcium Nitrate`) |>
  autoplot()+
  labs(title = "ACF Plot for CN", x = "Time (Months)", y = "ACF")
```

Parctial Autocorrelation Function

```
Fertiliser |>
  PACF(`Calcium Nitrate`) |>
  autoplot()+
  labs(title = "PACF Plot for CN", x = "Time (Months)", y = "PACF")
```

Test with a Differenced Series

```
#Fertiliser |>
 # features(`Calcium Nitrate` |>
  #          difference(12) |>
   #         difference(1),unitroot_kpss)
```

Let us now test stationarity after taking one normal differencing.

```
URTST<- Fertiliser |>
  features(`Calcium Nitrate`|>
             difference(1),unitroot_kpss)
URTST
# A tibble: 1 × 2
  kpss_stat kpss_pvalue
      <dbl>       <dbl>
1    0.0274         0.1
```

DIAMMONIUM PHOSPHATE

```
Fertiliser|>
  autoplot(`Diammonium Phosphate`)+
  xlab("Months") +
  ylab("Fertiliser (DAP)")+
   ggtitle("Time Series Plot of DAP Over Time [2010 - 2023")
```

Normality test

```
library(tseries)
jarque.bera.test(Fertiliser$`Diammonium Phosphate`)


    Jarque Bera Test

data:  Fertiliser$`Diammonium Phosphate`
X-squared = 78.566, df = 2, p-value < 2.2e-16
```

Plot a gg_season plot

```
Fertiliser |>
  gg_season(`Diammonium Phosphate`)+
  ggtitle("Time Series Plot (GG-Season plot) of DAP Over Time [2010 - 2023")
```

Seasonal Subseries Plot

```
Fertiliser |>
  gg_subseries(`Diammonium Phosphate`)+
  ggtitle("Time Series Plot (GG-Sub series plot) of DAP Over Time [2010 - 2023")
```

```
dcmp2 <- Fertiliser |>
  model(STL(`Diammonium Phosphate`))
dcmp2
```

```
# A mable: 1 x 1
  `STL(\`Diammonium Phosphate\`)`
                      <model>
1                       <STL>
```

Plot the Components

```
dcmp2 |>
  components() |>
  autoplot(`Diammonium Phosphate`)
```

Extract the Components using Component Function

```
dcmp2 |>
  components()
# A dable: 168 x 7 [1M]
# Key:     .model [1]
# :        Diammonium Phosphate = trend + season_year + remainder
   .model            Months `Diammonium Phosphate`  trend season  year remainder
   <chr>             <mth>                  <dbl>  <dbl>      <dbl>     <dbl>
 1 STL(`Diammonium… 2010 Jan                35594   6753.     29267.     -426.
 2 STL(`Diammonium… 2010 Feb                 2535   7793.      1219.    -6477.
 3 STL(`Diammonium… 2010 Mar                25036   8833.      8643.     7560.
 4 STL(`Diammonium… 2010 Apr                20436   9873.      3126.     7437.
 5 STL(`Diammonium… 2010 May                    0  10873.     -5281.    -5591.
 6 STL(`Diammonium… 2010 Jun                  319  11872.     -8578.    -2975.
 7 STL(`Diammonium… 2010 Jul                    0  12872.    -12176.     -697.
 8 STL(`Diammonium… 2010 Aug                    0  13739.     -9461.    -4278.
 9 STL(`Diammonium… 2010 Sep                  475. 14606.     -2402.   -11730.
10 STL(`Diammonium… 2010 Oct                  300  15473.     -9909.    -5264.
# □ 158 more rows
# □ 1 more variable: season_adjust <dbl>
```

Seasonality of the season_year

I want to better understanding of what is happening in the seasonal component.

```
dcmp2 |>
  components() |>
  gg_season(season  year)+
  ggtitle("Seasonal Variation in DAP Fertiliser")
```

Seasonal Sub Series Plot

```
dcmp2 |>
  components() |>
  gg_subseries(season  year)+
  ggtitle("Time Series Plot (GG-Sub series for season year) for DAP [2010 - 2023")
```

Difference the Series

Here we use the function, Differencing as shown below

Seasonal Difference

```
Fertiliser|>
  autoplot(`Diammonium Phosphate`|>
           difference(12))+
  xlab("Year") +
  ylab("DAP")+
   ggtitle("Time Series Plot of Stationary DAP")
```

Stabilize the Variance by Taking the Normal Difference

It is recommended that you take the seasonal difference (quarter, m = 4, monthly, m = 12 and so on) and then take the first difference.

```
Fertiliser |>
  autoplot(`Diammonium Phosphate`|>
           difference(1))+
  ggtitle("Seasonal and First Difference to Stabalize the Variance")
```

Stationarity test (kpss function)

```
URTST1<- Fertiliser |>
  features(`Diammonium Phosphate`, unitroot_kpss)
URTST1
```

```
# A tibble: 1 × 2
  kpss_stat kpss_pvalue
      <dbl>       <dbl>
1     0.199         0.1
```

Null and Alternative hypothesis

Null Hypotheses

The time series is stationary

Alternative HYpothesis

The time series is not stationary

From the results above, the times series is not stationary since the p-value is less than 0.05.

Test How many Differencing we need to make the series stationary

```
Fertiliser |>
  features(`Diammonium Phosphate`, unitroot_ndiffs)
# A tibble: 1 × 1
  ndiffs
   <int>
1      0
Fertiliser |>
  features(`Diammonium Phosphate`, unitroot_nsdiffs)
# A tibble: 1 × 1
  nsdiffs
    <int>
1       0
```

Autocorrelation Function (ACF)

```
Fertiliser |>
  ACF(`Diammonium Phosphate`) |>
  autoplot()+
  labs(title = "ACF Plot for DAP", x = "Time (Months)", y = "ACF")
```

Parctial Autocorrelation Function

```
Fertiliser |>
  PACF(`Diammonium Phosphate`) |>
  autoplot()+
  labs(title = "PACF Plot for DAP", x = "Time (Months)", y = "PACF")
```

Test with a Differenced Series

```
#Fertiliser |>
 # features(`Calcium Nitrate` |>
  #         difference(12) |>
   #         difference(1),unitroot_kpss)
```

Let us now test stationarity after taking one normal differencing.

```
#URTST<- Fertiliser |>
 # features(`Calcium Nitrate`|>
  #         difference(1),unitroot_kpss)
#URTST
```

MURIATE OF POTASH

```
Fertiliser|>
  autoplot(`Muriate of Potash`)+
  xlab("Months") +
  ylab("Fertiliser (MP)")+
   ggtitle("Time Series Plot of Muriate of Potash Over Time [2010 - 2023")
```

Normality test

```
library(tseries)
jarque.bera.test(Fertiliser$`Muriate of Potash`)


    Jarque Bera Test


data:  Fertiliser$`Muriate of Potash`
X-squared = 10690, df = 2, p-value < 2.2e-16
```

Plot a gg_season plot

```
Fertiliser |>
  gg_season(`Muriate of Potash`)+
  ggtitle("Time Series Plot (GG-Season plot) of MP Over Time [2010 - 2023")
```

Seasonal Subseries Plot

```
Fertiliser |>
  gg_subseries(`Muriate of Potash`)+
  ggtitle("Time Series Plot (GG-Sub series plot) of MP Over Time [2010 - 2023")
```

```
dcmp3 <- Fertiliser |>
  model(STL(`Muriate of Potash`))
dcmp3
# A mable: 1 x 1
  `STL(\`Muriate of Potash\`)`
                        <model>
1                         <STL>
```

Plot the Components

```
dcmp3 |>
  components() |>
  autoplot(`Muriate of Potash`)
```

Extract the Components using Component Function

```
dcmp3 |>
  components()
# A dable: 168 x 7 [1M]
# Key:     .model [1]
# :        Muriate of Potash = trend + season_year + remainder
   .model   Months `Muriate of Potash` trend season_year remainder season_adjust
   <chr>     <mth>              <dbl> <dbl>       <dbl>     <dbl>         <dbl>
 1 STL(`… 2010 Jan              3644  1176.      264.      2203.        3380.
 2 STL(`… 2010 Feb                 0  1069.     -315.      -754.         315.
 3 STL(`… 2010 Mar               513   962.       65.4     -514.         448.
 4 STL(`… 2010 Apr               418.  854.      517.      -953.         -98.9
 5 STL(`… 2010 May               220   750.     -192.      -338.         412.
 6 STL(`… 2010 Jun               550   646.     -257.       161.         807.
 7 STL(`… 2010 Jul                 0   542.     -368.      -175.         368.
 8 STL(`… 2010 Aug               620   449.       -3.34     174.         623.
 9 STL(`… 2010 Sep                 0   357.      184.      -540.        -184.
10 STL(`… 2010 Oct                 0   264.       14.5     -278.         -14.5
# □ 158 more rows
```

Seasonality of the season_year

I want to better understanding of what is happening in the seasonal component.

```
dcmp3 |>
  components() |>
  gg_season(season_year)+
  ggtitle("Seasonal Variation in MP")
```

Seasonal Sub Series Plot

```
dcmp3 |>
  components() |>
  gg_subseries(season_year)+
  ggtitle("Time Series Plot (GG-Sub series for season year) for MP [2010 - 2023")
```

From the June seasonality has been decreasing. The same is seen in December

Difference the Series

Here we use the function, differencing as shown below

Seasonal Difference

```
Fertiliser|>
  autoplot(`Muriate of Potash`|>
           difference(12))+
  xlab("Year") +
```

```
 ylab("MP")+
  ggtitle("Time Series Plot of Stationary for MP")
```

Stabilize the Variance by Taking the Normal Difference

It is recommended that you take the seasonal difference (quarter, m = 4, monthly, m = 12 and so on) and then take the first difference.

```
Fertiliser |>
  autoplot(`Muriate of Potash`|>
           difference(1))+
  ggtitle("Seasonal and First Difference to Stabalize the Variance")
```

Stationarity test (kpss function)

```
URTST1<- Fertiliser |>
  features(`Muriate of Potash`, unitroot_kpss)
URTST1
# A tibble: 1 × 2
  kpss_stat kpss_pvalue
      <dbl>       <dbl>
1     0.602      0.0225
```

Null and Alternative hypothesis

Null Hypotheses

The time series is stationary

Alternative HYpothesis

The time series is not stationary

From the results above, the times series is not stationary since the p-value is less than 0.05.

Test How many Differencing we need to make the series stationary

```
Fertiliser |>
  features(`Muriate of Potash`, unitroot_ndiffs)
# A tibble: 1 × 1
  ndiffs
   <int>
1      1
Fertiliser |>
  features(`Muriate of Potash`, unitroot_nsdiffs)
# A tibble: 1 × 1
  nsdiffs
    <int>
1       0
```

Autocorrelation Function (ACF)

```
Fertiliser |>
  ACF(`Muriate of Potash`) |>
  autoplot()+
  labs(title = "ACF Plot for MP", x = "Time (Months)", y = "ACF")
```

Parctial Autocorrelation Function

```
Fertiliser |>
  PACF(`Muriate of Potash`) |>
  autoplot()+
  labs(title = "PACF Plot for MP", x = "Time (Months)", y = "PACF")
```

Test with a Differenced Series

```
#Fertiliser |>
 # features(`Calcium Nitrate` |>
  #          difference(12) |>
   #         difference(1),unitroot_kpss)
```

Let us now test stationarity after taking one normal differencing.

```
URTST<- Fertiliser |>
 features(`Calcium Nitrate`|>
           difference(1),unitroot_kpss)
URTST
# A tibble: 1 × 2
  kpss_stat kpss_pvalue
```

```
    <dbl>      <dbl>
1   0.0274       0.1
```

NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER

```
Fertiliser|>
  autoplot(`NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER`)+
  xlab("Months") +
  ylab("Fertiliser (NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER)")+
   ggtitle("Time Series Plot of NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER Over Time [2010 -
2023")
```

Normality test

```
library(tseries)
jarque.bera.test(Fertiliser$`NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER`)
```

```
    Jarque Bera Test

data:  Fertiliser$NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER
X-squared = 144.93, df = 2, p-value < 2.2e-16
```

Plot a gg_season plot

```
Fertiliser |>
  gg_season(`NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER`)+
  ggtitle("Time Series Plot (GG-Season plot) of NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER
Over Time [2010 - 2023")
```

Seasonal Subseries Plot

```
Fertiliser |>
  gg_subseries(`NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER`)+
  ggtitle("Time Series Plot (GG-Sub series plot) of NITROGEN, PHOSPHORUS AND POTASIUM
FERTILISER Over Time [2010 - 2023")
```

```
dcmp4 <- Fertiliser |>
  model(STL(`NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER`))
dcmp4
# A mable: 1 x 1
  `STL(NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER)`
     <model>
1      <STL>
```

Plot the Components

```
dcmp4 |>
  components() |>
  autoplot(`NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER`)
```

Extract the Components using Component Function

```
dcmp4 |>
  components()
# A dable: 168 x 7 [1M]
# Key:     .model [1]
# :        NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER = trend + season_year + remainder
   .model     Months    NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER trend season_year
remainder season_adjust
   <chr>       <mth>  <dbl> <dbl>       <dbl>     <dbl>         <dbl>
 1 STL(NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER) 2010 Jan 23061  7648.       5713.     9700.
17348.
 2 STL(NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER) 2010 Feb  5619  7884.       5332.    -7596.
287.
 3 STL(NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER) 2010 Mar  1742  8120.      -3873.    -
2505.         5615.
 4 STL(NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER) 2010 Apr   362. 8356.      -6912.    -1082.
7273.
 5 STL(NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER) 2010 May  1456  8585.      -8393.
1264.         9849.
 6 STL(NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER) 2010 Jun   222. 8814.      -8476.     -115.
8699.
```

```
 7 STL(NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER) 2010 Jul 39907  9044.     28379.    2485.
11528.
 8 STL(NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER) 2010 Aug 28505  9287.     26045.    -
6827.        2460.
 9 STL(NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER) 2010 Sep  1191  9531.     -8940.     600.
10131.
10 STL(NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER) 2010 Oct  4142  9775.     -8217.
2584.        12359.
# □ 158 more rows
```

Seasonality of the season_year

I want to better understanding of what is happening in the seasonal component.

```
dcmp4 |>
  components() |>
  gg_season(season_year)+
  ggtitle("Seasonal Variation in NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER")
```

Seasonal Sub Series Plot

```
dcmp4 |>
  components() |>
  gg_subseries(season_year)+
  ggtitle("Time Series Plot (GG-Sub series for season year) for NITROGEN, PHOSPHORUS AND POTASIUM
FERTILISER [2010 - 2023")
```

From the June seasonality has been decreasing. The same is seen in December

Difference the Series

Here we use the function, differencing as shown below

Seasonal Difference

```
Fertiliser|>
  autoplot(`NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER`|>
           difference(12))+
  xlab("Year") +
  ylab("NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER")+
   ggtitle("Time Series Plot of Stationary for NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER")
```

Stabilize the Variance by Taking the Normal Difference

```
Fertiliser |>
  autoplot(`NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER`|>
           difference(1))+
  ggtitle("Seasonal and First Difference to Stabalize the Variance")
```

Stationarity test (kpss function)

```
URTST1<- Fertiliser |>
  features(NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER, unitroot_kpss)
URTST1
# A tibble: 1 × 2
  kpss_stat kpss_pvalue
      <dbl>       <dbl>
1    0.0562         0.1
```

Null and Alternative hypothesis

Null Hypotheses

The time series is stationary

Alternative HYpothesis

The time series is not stationary

From the results above, the times series is not stationary since the p-value is less than 0.05.

Test How many Differencing we need to make the series stationary

```
Fertiliser |>
  features(`NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER`, unitroot_ndiffs)
# A tibble: 1 × 1
  ndiffs
```

```
  <int>
1     0
```

```
Fertiliser |>
  features(`NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER`, unitroot_nsdiffs)
# A tibble: 1 × 1
  nsdiffs
    <int>
1       0
```

Autocorrelation Function (ACF)

```
Fertiliser |>
  ACF(`NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER`) |>
  autoplot()+
  labs(title = "ACF Plot for NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER", x = "Time
(Months)", y = "ACF")
```

Parctial Autocorrelation Function

```
Fertiliser |>
  PACF(`NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER`) |>
  autoplot()+
  labs(title = "PACF Plot for NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER", x = "Time
(Months)", y = "PACF")
```

Test with a Differenced Series

```
#Fertiliser |>
 # features(`Calcium Nitrate` |>
  #          difference(12) |>
   #         difference(1),unitroot_kpss)
```

Let us now test stationarity after taking one normal differencing.

```
#URTST<- Fertiliser |>
 #features(`Calcium Nitrate`|>
 #          difference(1),unitroot_kpss)
#URTST
```

UREA

```
Fertiliser|>
  autoplot(`Urea`)+
  xlab("Months") +
  ylab("Fertiliser (Urea)")+
   ggtitle("Time Series Plot of Urea Over Time [2010 - 2023")
```

Normality test

```
library(tseries)
jarque.bera.test(Fertiliser$Urea)
```

```
    Jarque Bera Test

data:  Fertiliser$Urea
X-squared = 152.66, df = 2, p-value < 2.2e-16
```

Plot a gg_season plot

```
Fertiliser |>
  gg_season(Urea)+
  ggtitle("Time Series Plot (GG-Season plot) of Urea Over Time [2010 - 2023")
```

Seasonal Subseries Plot

```
Fertiliser |>
  gg_subseries(`Urea`)+
  ggtitle("Time Series Plot (GG-Sub series plot) of Urea Over Time [2010 - 2023")
```

```
dcmp5 <- Fertiliser |>
  model(STL(`Urea`))
dcmp5
# A mable: 1 x 1
  `STL(Urea)`
```

```
    <model>
1     <STL>
```

Plot the Components

```
dcmp5 |>
  components() |>
  autoplot(`Urea`)
```

Extract the Components using Component Function

```
dcmp5 |>
  components()
# A dable: 168 x 7 [1M]
# Key:     .model [1]
# :        Urea = trend + season_year + remainder
   .model      Months  Urea trend season_year remainder season_adjust
   <chr>        <mth> <dbl> <dbl>       <dbl>     <dbl>         <dbl>
 1 STL(Urea) 2010 Jan     0 4287.    -1815.      -2472.       1815.
 2 STL(Urea) 2010 Feb     0 4455.     3422.      -7877.      -3422.
 3 STL(Urea) 2010 Mar     0 4624.     4602.      -9226.      -4602.
 4 STL(Urea) 2010 Apr 19149 4792.     2695.      11662.      16454.
 5 STL(Urea) 2010 May 16300 4905.     -910.      12305.      17210.
 6 STL(Urea) 2010 Jun     0 5017.      -6.17    -5011.         6.17
 7 STL(Urea) 2010 Jul  5618 5129.    -2735.      3224.       8353.
 8 STL(Urea) 2010 Aug 12020 5220.     -180.      6980.      12200.
 9 STL(Urea) 2010 Sep   500 5310.    -1974.     -2836.       2474.
10 STL(Urea) 2010 Oct     0 5401.    -1918.     -3482.       1918.
# □ 158 more rows
```

Seasonality of the season_year

I want to better understanding of what is happening in the seasonal component.

```
dcmp5 |>
  components() |>
  gg_season(season_year)+
  ggtitle("Seasonal Variation in Urea")
```

Seasonal Sub Series Plot

```
dcmp5 |>
  components() |>
  gg_subseries(season_year)+
  ggtitle("Time Series Plot (GG-Sub series for season year) for Urea [2010 - 2023")
```

From the June seasonality has been decreasing. The same is seen in December

Difference the Series

Here we use the function, differencing as shown below

Seasonal Difference

```
Fertiliser|>
  autoplot(`Urea`|>
           difference(12))+
  xlab("Year") +
  ylab("Urea")+
   ggtitle("Time Series Plot of Stationary for Urea")
```

Stabilize the Variance by Taking the Normal Difference

Note!!

It is recommended that you take the seasonal difference (quarter, m = 4, monthly, m = 12 and so on) and then take the first difference.

```
Fertiliser |>
  autoplot(`Urea`|>
           difference(1))+
  ggtitle("Seasonal and First Difference to Stabalize the Variance")
```

Stationarity test (kpss function)

```
URTST1<- Fertiliser |>
  features(Urea, unitroot_kpss)
URTST1
```

```
# A tibble: 1 × 2
  kpss_stat kpss_pvalue
      <dbl>       <dbl>
1     0.469      0.0487
```

Null and Alternative hypothesis

Null Hypotheses

The time series is stationary

Alternative HYpothesis

The time series is not stationary

Test How many Differencing we need to make the series stationary

```
Fertiliser |>
  features(`Urea`, unitroot_ndiffs)
# A tibble: 1 × 1
  ndiffs
   <int>
1      1
Fertiliser |>
  features(`Urea`, unitroot_nsdiffs)
# A tibble: 1 × 1
  nsdiffs
    <int>
1       0
```

Autocorrelation Function (ACF)

```
Fertiliser |>
  ACF(`Urea`) |>
  autoplot()+
  labs(title = "ACF Plot for Urea", x = "Time (Months)", y = "ACF")
```

Parctial Autocorrelation Function

```
Fertiliser |>
  PACF(`Urea`) |>
  autoplot()+
  labs(title = "PACF Plot for Urea", x = "Time (Months)", y = "PACF")
```

Test with a Differenced Series

```
#Fertiliser |>
 # features(`Calcium Nitrate` |>
  #          difference(12) |>
   #         difference(1),unitroot_kpss)
```

Let us now test stationarity after taking one normal differencing.

```
URTST<- Fertiliser |>
 features(`Calcium Nitrate`|>
           difference(1),unitroot_kpss)
URTST
# A tibble: 1 × 2
  kpss_stat kpss_pvalue
      <dbl>       <dbl>
1    0.0274         0.1
```

Total Fertiliser Demand

```
Fertiliser|>
  autoplot(`total fertiliser`)+
  xlab("Months") +
  ylab("Fertiliser (Total)")+
   ggtitle("Time Series Plot of Total Fertiliser Over Time [2010 - 2023")
```

Normality test

```
library(tseries)
jarque.bera.test(Fertiliser$`total fertiliser`)


    Jarque Bera Test
```

```
data:  Fertiliser$`total fertiliser`
X-squared = 35.55, df = 2, p-value = 1.908e-08
```

Plot a gg_season plot

```
Fertiliser |>
  gg_season(`total fertiliser`)+
  ggtitle("Time Series Plot (GG-Season plot) of Total Fertiliser Over Time [2010 - 2023")
```

Seasonal Subseries Plot

```
Fertiliser |>
  gg_subseries(`total fertiliser`)+
  ggtitle("Time Series Plot (GG-Sub series plot) of Total Fertiliser Over Time [2010 - 2023")
```

Seasonal Component:.

```
dcmp6 <- Fertiliser |>
  model(STL(`total fertiliser`))
dcmp6
# A mable: 1 x 1
  `STL(\`total fertiliser\`)`
                      <model>
1                       <STL>
```

Plot the Components

```
dcmp6 |>
  components() |>
  autoplot(`total fertiliser`)
```

Extract the Components using Component Function

```
dcmp6 |>
  components()
# A dable: 168 x 7 [1M]
# Key:     .model [1]
# :        total fertiliser = trend + season_year + remainder
   .model   Months `total fertiliser`  trend season_year remainder season_adjust
   <chr>     <mth>             <dbl>  <dbl>       <dbl>     <dbl>         <dbl>
 1 STL(`… 2010 Jan            78554  22734.      41671.    14148.        36883.
 2 STL(`… 2010 Feb            20030  24861.      19232.   -24063.          798.
 3 STL(`… 2010 Mar            29276. 26988.      17417.   -15129.        11859.
 4 STL(`… 2010 Apr            42409. 29114.       3651.     9643.        38758.
 5 STL(`… 2010 May            44149  31129.     -14984.    28004.        59133.
 6 STL(`… 2010 Jun            2914. 33144.     -18890.   -11340.        21804.
 7 STL(`… 2010 Jul            50525  35159.      15244.      123.        35281.
 8 STL(`… 2010 Aug            51150  37087.      11398.     2665.        39752.
 9 STL(`… 2010 Sep            4276. 39016.     -16484.   -18255.        20760.
10 STL(`… 2010 Oct            6514  40944.     -26936.    -7494.        33450.
# □ 158 more rows
```

Seasonality of the season_year

I want to better understanding of what is happening in the seasonal component.

```
dcmp6 |>
  components() |>
  gg_season(season_year)+
  ggtitle("Seasonal Variation in Total Fertiliser")
```

Seasonal Sub Series Plot

```
dcmp6 |>
  components() |>
  gg_subseries(season_year)+
  ggtitle("Time Series Plot (GG-Sub series for season year) for Total Fertiliser [2010 - 2023")
```

Difference the Series

Here we use the function, differencing as shown below

Seasonal Difference

```
Fertiliser|>
  autoplot(`total fertiliser`|>
           difference(12))+
```

```
  xlab("Year") +
  ylab("Total Fertiliser")+
   ggtitle("Time Series Plot of Stationary for Total Fertiliser")
```

Stabilize the Variance by Taking the Normal Difference

```
Fertiliser |>
  autoplot(`total fertiliser`|>
             difference(1))+
  ggtitle("Seasonal and First Difference to Stabalize the Variance")
```

Stationarity test (kpss function)

```
URTST1<- Fertiliser |>
  features(`total fertiliser`, unitroot_kpss)
URTST1
# A tibble: 1 × 2
  kpss_stat kpss_pvalue
      <dbl>       <dbl>
1     0.623      0.0205
```

Null and Alternative hypothesis

Null Hypotheses

The time series is stationary

Alternative HYpothesis

The time series is not stationary

From the results above, the times series is not stationary since the p-value is less than 0.05.

Test How many Differencing we need to make the series stationary

```
Fertiliser |>
  features(`total fertiliser`, unitroot_ndiffs)
# A tibble: 1 × 1
  ndiffs
    <int>
1      1
Fertiliser |>
  features(`total fertiliser`, unitroot_nsdiffs)
# A tibble: 1 × 1
  nsdiffs
    <int>
1       0
```

Autocorrelation Function (ACF)

```
Fertiliser |>
  ACF(`total fertiliser`) |>
  autoplot()+
  labs(title = "ACF Plot for Total Fertiliser", x = "Time (Months)", y = "ACF")
```

Parctial Autocorrelation Function

```
Fertiliser |>
  PACF(`total fertiliser`) |>
  autoplot()+
  labs(title = "PACF Plot for Total Fertiliser", x = "Time (Months)", y = "PACF")
```

Test with a Differenced Series

```
#Fertiliser |>
 # features(`Calcium Nitrate` |>
  #          difference(12) |>
   #         difference(1),unitroot_kpss)
```

Let us now test stationarity after taking one normal differencing.

```
URTST<- Fertiliser |>
 features(`Calcium Nitrate`|>
            difference(1),unitroot_kpss)
URTST
```

```
# A tibble: 1 × 2
  kpss_stat kpss_pvalue
      <dbl>       <dbl>
1    0.0274         0.1
```

SEASONAL AUTOREGRESSIVE INTEGRATED MOVING AVERAGE (SARIMA)

Fit the ARIMA Models

```
fit  CN <- Fertiliser |>
  model(arima = ARIMA(`Calcium Nitrate`))
fit_CN
# A mable: 1 x 1
                  arima
                 <model>
1 <ARIMA(1,1,1)(0,0,1)[12]>
fit  CAN <- Fertiliser |>
  model(arima = ARIMA(`Calcium Ammonium Nitrate`))
fit_CAN
# A mable: 1 x 1
                arima
               <model>
1 <ARIMA(0,0,0) w/ mean>
fit_DAP <- Fertiliser |>
  model(arima = ARIMA(`Diammonium Phosphate`))
fit  DAP
# A mable: 1 x 1
                        arima
                       <model>
1 <ARIMA(0,0,0)(2,0,0)[12] w/ mean>
fit  MP <- Fertiliser |>
  model(arima = ARIMA(`Muriate of Potash`))
fit_MP
# A mable: 1 x 1
                  arima
                 <model>
1 <ARIMA(1,1,4)(0,0,1)[12]>
fit  NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER <- Fertiliser |>
  model(arima = ARIMA(NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER))
fit_NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER
# A mable: 1 x 1
                        arima
                       <model>
1 <ARIMA(2,0,0)(2,0,0)[12] w/ mean>
fit_Urea <- Fertiliser |>
  model(arima = ARIMA(Urea))
fit_Urea
# A mable: 1 x 1
         arima
        <model>
1 <ARIMA(0,1,1)>
fit_Total <- Fertiliser |>
  model(arima = ARIMA(`total fertiliser`))
fit_Total
# A mable: 1 x 1
                  arima
                 <model>
1 <ARIMA(1,1,4)(0,0,1)[12]>
```

View the Model Output Using report() Function

```
report(fit_CAN)
Series: Calcium Ammonium Nitrate
Model: ARIMA(0,0,0) w/ mean


Coefficients:
      constant
     8853.0020
s.e.   956.3476
```

```
sigma^2 estimated as 154571384:  log likelihood=-1821.8
AIC=3647.6   AICc=3647.67   BIC=3653.84
report(fit_CN)
Series: Calcium Nitrate
Model: ARIMA(1,1,1)(0,0,1)[12]

Coefficients:
         ar1      ma1    sma1
      0.1029  -0.9615  0.1373
s.e.  0.0816   0.0244  0.1134

sigma^2 estimated as 6118068:  log likelihood=-1541.51
AIC=3091.02   AICc=3091.26   BIC=3103.49
report(fit_DAP)
Series: Diammonium Phosphate
Model: ARIMA(0,0,0)(2,0,0)[12] w/ mean

Coefficients:
        sar1    sar2  constant
      0.3092  0.2059  8432.586
s.e.  0.0774  0.0788  1383.260

sigma^2 estimated as 395584183:  log likelihood=-1901.23
AIC=3810.46   AICc=3810.7   BIC=3822.95
report(fit_MP)
Series: Muriate of Potash
Model: ARIMA(1,1,4)(0,0,1)[12]

Coefficients:
         ar1      ma1     ma2     ma3     ma4     sma1
      -0.0901  -0.6831  0.0808  0.0496  -0.3994  -0.0428
s.e.   0.2334   0.2185  0.2060  0.1078   0.0973   0.1381

sigma^2 estimated as 4300024:  log likelihood=-1510.62
AIC=3035.23   AICc=3035.94   BIC=3057.06
report(fit_NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER)
Series: NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER
Model: ARIMA(2,0,0)(2,0,0)[12] w/ mean

Coefficients:
         ar1     ar2    sar1    sar2   constant
      0.0454  0.1770  0.2591  0.3985  3219.8165
s.e.  0.0780  0.0808  0.0751  0.0839   865.3579

sigma^2 estimated as 187363368:  log likelihood=-1839.26
AIC=3690.52   AICc=3691.05   BIC=3709.27
report(fit_Urea)
Series: Urea
Model: ARIMA(0,1,1)

Coefficients:
          ma1
      -0.9747
s.e.   0.0177

sigma^2 estimated as 83726476:  log likelihood=-1761.25
AIC=3526.51   AICc=3526.58   BIC=3532.74
report(fit_Total)
Series: total fertiliser
Model: ARIMA(1,1,4)(0,0,1)[12]

Coefficients:
```

```
        ar1      ma1      ma2      ma3      ma4    sma1
     -0.5751  -0.1384  -0.4289  -0.1520  -0.2326  0.1308
s.e.  0.4092   0.3967   0.3128   0.0786   0.0946  0.0755

sigma^2 estimated as 1.149e+09:  log likelihood=-1977.17
AIC=3968.33   AICc=3969.04   BIC=3990.16
fit  CAN|>
  tidy()
# A tibble: 1 × 6
  .model term     estimate std.error statistic  p.value
  <chr>  <chr>       <dbl>     <dbl>     <dbl>    <dbl>
1 arima  constant    8853.      956.      9.26 9.62e-17
fit_CN|>
  tidy()
# A tibble: 3 × 6
  .model term  estimate std.error statistic  p.value
  <chr>  <chr>    <dbl>     <dbl>     <dbl>    <dbl>
1 arima  ar1      0.103    0.0816      1.26 2.09e- 1
2 arima  ma1     -0.962    0.0244    -39.4  1.73e-86
3 arima  sma1     0.137    0.113       1.21 2.28e- 1
fit_DAP|>
  tidy()
# A tibble: 3 × 6
  .model term     estimate std.error statistic       p.value
  <chr>  <chr>       <dbl>     <dbl>     <dbl>         <dbl>
1 arima  sar1        0.309    0.0774      3.99 0.0000974
2 arima  sar2        0.206    0.0788      2.61 0.00975
3 arima  constant 8433.    1383.          6.10 0.00000000722
fit_MP|>
  tidy()
# A tibble: 6 × 6
  .model term  estimate std.error statistic   p.value
  <chr>  <chr>    <dbl>     <dbl>     <dbl>     <dbl>
1 arima  ar1    -0.0901    0.233     -0.386 0.700
2 arima  ma1    -0.683     0.219     -3.13  0.00209
3 arima  ma2     0.0808    0.206      0.392 0.695
4 arima  ma3     0.0496    0.108      0.460 0.646
5 arima  ma4    -0.399     0.0973    -4.10  0.0000634
6 arima  sma1   -0.0428    0.138     -0.310 0.757
fit  NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER|>
  tidy()
# A tibble: 5 × 6
  .model term      estimate std.error statistic    p.value
  <chr>  <chr>        <dbl>     <dbl>     <dbl>      <dbl>
1 arima  ar1         0.0454    0.0780     0.581 0.562
2 arima  ar2         0.177     0.0808     2.19  0.0299
3 arima  sar1        0.259     0.0751     3.45  0.000712
4 arima  sar2        0.398     0.0839     4.75  0.00000436
5 arima  constant 3220.      865.         3.72  0.000271
fit  Urea|>
  tidy()
# A tibble: 1 × 6
  .model term  estimate std.error statistic   p.value
  <chr>  <chr>    <dbl>     <dbl>     <dbl>     <dbl>
1 arima  ma1     -0.975    0.0177    -54.9 7.53e-109
fit  Total|>
  tidy()
# A tibble: 6 × 6
  .model term  estimate std.error statistic p.value
  <chr>  <chr>    <dbl>     <dbl>     <dbl>   <dbl>
1 arima  ar1     -0.575    0.409     -1.41   0.162
2 arima  ma1     -0.138    0.397     -0.349  0.728
3 arima  ma2     -0.429    0.313     -1.37   0.172
4 arima  ma3     -0.152    0.0786    -1.93   0.0548
5 arima  ma4     -0.233    0.0946    -2.46   0.0149
6 arima  sma1     0.131    0.0755     1.73   0.0850
```

Plot the Forecast for the Next 24 Periods

```
fit  CAN |>
  forecast(h=36)
# A fable: 36 x 4 [1M]
# Key:     .model [1]
   .model   Months `Calcium Ammonium Nitrate` .mean
   <chr>     <mth>                   <dist> <dbl>
 1 arima  2024 Jan          N(8853, 1.5e+08) 8853.
 2 arima  2024 Feb          N(8853, 1.5e+08) 8853.
 3 arima  2024 Mar          N(8853, 1.5e+08) 8853.
 4 arima  2024 Apr          N(8853, 1.5e+08) 8853.
 5 arima  2024 May          N(8853, 1.5e+08) 8853.
 6 arima  2024 Jun          N(8853, 1.5e+08) 8853.
 7 arima  2024 Jul          N(8853, 1.5e+08) 8853.
 8 arima  2024 Aug          N(8853, 1.5e+08) 8853.
 9 arima  2024 Sep          N(8853, 1.5e+08) 8853.
10 arima  2024 Oct          N(8853, 1.5e+08) 8853.
# □ 26 more rows
fit  CAN |>
  forecast(h=36) |>
  autoplot()
```

```
fit  CN |>
  forecast(h=36)
# A fable: 36 x 4 [1M]
# Key:     .model [1]
   .model   Months `Calcium Nitrate` .mean
   <chr>     <mth>           <dist> <dbl>
 1 arima  2024 Jan  N(3230, 6118069) 3230.
 2 arima  2024 Feb  N(2933, 6240423) 2933.
 3 arima  2024 Mar  N(2986, 6257628) 2986.
 4 arima  2024 Apr  N(6337, 6269435) 6337.
 5 arima  2024 May  N(2979, 6280743) 2979.
 6 arima  2024 Jun  N(2498, 6292002) 2498.
 7 arima  2024 Jul  N(2504, 6303254) 2504.
 8 arima  2024 Aug  N(2605, 6314507) 2605.
 9 arima  2024 Sep  N(2521, 6325759) 2521.
10 arima  2024 Oct  N(2985, 6337011) 2985.
# □ 26 more rows
fit  CN |>
  forecast(h=36) |>
  autoplot()
```

```
fit  DAP |>
  forecast(h=36)
# A fable: 36 x 4 [1M]
# Key:     .model [1]
   .model   Months `Diammonium Phosphate`  .mean
   <chr>     <mth>                <dist>  <dbl>
 1 arima  2024 Jan       N(33102, 4e+08) 33102.
 2 arima  2024 Feb        N(8433, 4e+08)  8433.
 3 arima  2024 Mar       N(38551, 4e+08) 38551.
 4 arima  2024 Apr        N(8433, 4e+08)  8433.
 5 arima  2024 May        N(8524, 4e+08)  8524.
 6 arima  2024 Jun        N(8433, 4e+08)  8433.
 7 arima  2024 Jul        N(8433, 4e+08)  8433.
 8 arima  2024 Aug       N(20405, 4e+08) 20405.
 9 arima  2024 Sep       N(22718, 4e+08) 22718.
10 arima  2024 Oct       N(11500, 4e+08) 11500.
# □ 26 more rows
fit_DAP |>
  forecast(h=36) |>
  autoplot()
```

```
fit  MP |>
  forecast(h=36)
# A fable: 36 x 4 [1M]
# Key:     .model [1]
   .model   Months `Muriate of Potash`  .mean
   <chr>     <mth>              <dist>  <dbl>
 1 arima  2024 Jan      N(93, 4300096)   93.0
 2 arima  2024 Feb    N(-545, 4521235) -545.
 3 arima  2024 Mar    N(1002, 5133300) 1002.
 4 arima  2024 Apr     N(887, 5868042)  887.
 5 arima  2024 May    N(1076, 5868525) 1076.
 6 arima  2024 Jun    N(1662, 5878022) 1662.
 7 arima  2024 Jul    N(1563, 5886242) 1563.
 8 arima  2024 Aug    N(1660, 5894573) 1660.
 9 arima  2024 Sep    N(1672, 5902894) 1672.
10 arima  2024 Oct    N(1673, 5911216) 1673.
# □ 26 more rows
fit_MP |>
  forecast(h=36) |>
  autoplot()
```

```
fit  NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER |>
  forecast(h=36)
# A fable: 36 x 4 [1M]
# Key:     .model [1]
   .model   Months              NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER  .mean
   <chr>     <mth>           <dist>  <dbl>
 1 arima  2024 Jan  N(2935, 1.9e+08)  2935.
 2 arima  2024 Feb N(11623, 1.9e+08) 11623.
 3 arima  2024 Mar N(13584, 1.9e+08) 13584.
 4 arima  2024 Apr N(10946, 1.9e+08) 10946.
 5 arima  2024 May N(11745, 1.9e+08) 11745.
 6 arima  2024 Jun  N(5334, 1.9e+08)  5334.
 7 arima  2024 Jul  N(4942, 1.9e+08)  4942.
 8 arima  2024 Aug N(24918, 1.9e+08) 24918.
 9 arima  2024 Sep  N(8679, 1.9e+08)  8679.
10 arima  2024 Oct N(22210, 1.9e+08) 22210.
# □ 26 more rows
fit_NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER |>
  forecast(h=36) |>
  autoplot()
```

```
fit  Urea |>
  forecast(h=36)
# A fable: 36 x 4 [1M]
# Key:     .model [1]
   .model   Months            Urea .mean
   <chr>     <mth>          <dist> <dbl>
 1 arima  2024 Jan N(7994, 8.4e+07) 7994.
 2 arima  2024 Feb N(7994, 8.4e+07) 7994.
 3 arima  2024 Mar N(7994, 8.4e+07) 7994.
 4 arima  2024 Apr N(7994, 8.4e+07) 7994.
 5 arima  2024 May N(7994, 8.4e+07) 7994.
 6 arima  2024 Jun N(7994, 8.4e+07) 7994.
 7 arima  2024 Jul N(7994, 8.4e+07) 7994.
 8 arima  2024 Aug N(7994, 8.4e+07) 7994.
 9 arima  2024 Sep N(7994, 8.4e+07) 7994.
10 arima  2024 Oct N(7994, 8.4e+07) 7994.
# □ 26 more rows
fit_Urea |>
  forecast(h=36) |>
  autoplot()
```

```
fit_Total |>
  forecast(h=36)
```

```
# A fable: 36 x 4 [1M]
# Key:     .model [1]
  .model   Months `total fertiliser`  .mean
  <chr>     <mth>            <dist>  <dbl>
1 arima  2024 Jan  N(53813, 1.1e+09) 53813.
2 arima  2024 Feb  N(68939, 1.2e+09) 68939.
3 arima  2024 Mar  N(80802, 1.3e+09) 80802.
4 arima  2024 Apr  N(67116, 1.3e+09) 67116.
5 arima  2024 May  N(67080, 1.3e+09) 67080.
6 arima  2024 Jun  N(54170, 1.3e+09) 54170.
7 arima  2024 Jul  N(55182, 1.3e+09) 55182.
8 arima  2024 Aug  N(57070, 1.4e+09) 57070.
9 arima  2024 Sep  N(56238, 1.4e+09) 56238.
10 arima  2024 Oct  N(55731, 1.4e+09) 55731.
# □ 26 more rows
fit_Total |>
  forecast(h=36) |>
  autoplot()
```

Plot together with the data

```
fit_CAN |>
  forecast(h=36) |>
  autoplot(Fertiliser)+
  ggtitle("Forecast Plot with data")
```

```
fit  CN |>
  forecast(h=36) |>
  autoplot(Fertiliser)+
  ggtitle("Forecast Plot with data")
```

```
fit  DAP |>
  forecast(h=36) |>
  autoplot(Fertiliser)+
  ggtitle("Forecast Plot with data")
```

```
fit_MP |>
  forecast(h=36) |>
  autoplot(Fertiliser)+
  ggtitle("Forecast Plot with data")
```

```
fit_NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER |>
  forecast(h=36) |>
  autoplot(Fertiliser)+
  ggtitle("Forecast Plot with data")
```

```
fit  Urea |>
  forecast(h=36) |>
  autoplot(Fertiliser)+
  ggtitle("Forecast Plot with data")
```

```
fit  Total |>
  forecast(h=36) |>
  autoplot(Fertiliser)+
  ggtitle("Forecast Plot with data")
```

Evaluating point forecast accuracy

Use the Model to Forecast the Next 36 Months

```
fcst1 <-
  fit_CAN |>
  forecast(h = 36)
fcst1
# A fable: 36 x 4 [1M]
# Key:     .model [1]
```

```
  .model   Months `Calcium Ammonium Nitrate` .mean
  <chr>     <mth>                  <dist> <dbl>
1 arima  2024 Jan          N(8853, 1.5e+08) 8853.
2 arima  2024 Feb          N(8853, 1.5e+08) 8853.
3 arima  2024 Mar          N(8853, 1.5e+08) 8853.
4 arima  2024 Apr          N(8853, 1.5e+08) 8853.
5 arima  2024 May          N(8853, 1.5e+08) 8853.
6 arima  2024 Jun          N(8853, 1.5e+08) 8853.
7 arima  2024 Jul          N(8853, 1.5e+08) 8853.
8 arima  2024 Aug          N(8853, 1.5e+08) 8853.
9 arima  2024 Sep          N(8853, 1.5e+08) 8853.
10 arima  2024 Oct          N(8853, 1.5e+08) 8853.
# □ 26 more rows
```

Plot the Forecast Distribution

```
library(ggdist)
ggplot(data = fcst1, mapping = aes(x = Months, ydist = `Calcium Ammonium Nitrate`, fill = .model))+
  stat_halfeye(alpha = .5)+
  xlab("Time (Months)")+
  ylab("CAN")+
  ggtitle("36 Months Forest using automatic ARIMA")
```

```
ggplot(data = fcst1, mapping = aes(x = Months, ydist = `Calcium Ammonium Nitrate`))+
  stat_halfeye(alpha = .5)+
  autolayer(filter_index(Fertiliser, "2010 Jan"~.))+
  xlab("Time (Months)")+
  ylab("Calcium Ammonium Nitrate")+
  ggtitle("36 Months Forest using Automatic ARIMA")
```

Plot the Fitted and the Data

```
fit_CAN |>
  augment()|>
  ggplot(aes(x = Months))+
  geom_line(aes(y = `Calcium Ammonium Nitrate`, color = "Data"))+
  geom_line(aes(y = .fitted, color = "Fitted"))
```

```
fcst7 <-
  fit  CN |>
  forecast(h = 36)
fcst7
# A fable: 36 x 4 [1M]
# Key:     .model [1]
  .model   Months `Calcium Nitrate` .mean
  <chr>     <mth>           <dist> <dbl>
1 arima  2024 Jan  N(3230, 6118069) 3230.
2 arima  2024 Feb  N(2933, 6240423) 2933.
3 arima  2024 Mar  N(2986, 6257628) 2986.
4 arima  2024 Apr  N(6337, 6269435) 6337.
5 arima  2024 May  N(2979, 6280743) 2979.
6 arima  2024 Jun  N(2498, 6292002) 2498.
7 arima  2024 Jul  N(2504, 6303254) 2504.
8 arima  2024 Aug  N(2605, 6314507) 2605.
9 arima  2024 Sep  N(2521, 6325759) 2521.
10 arima  2024 Oct  N(2985, 6337011) 2985.
# □ 26 more rows
```

Plot the Forecast Distribution

```
ggplot(data = fcst7, mapping = aes(x = Months, ydist = `Calcium Nitrate`, fill = .model))+
  stat_halfeye(alpha = .5)+
  xlab("Time (Months)")+
  ylab("CN")+
  ggtitle("36 Months Forest using automatic ARIMA")
```

Plot the Fitted and the Data

```
fit_CN |>
  augment()|>
  ggplot(aes(x = Months))+
```

```
  geom_line(aes(y = `Calcium Nitrate`, color = "Data"))+
  geom_line(aes(y = .fitted, color = "Fitted"))
```

```
fit  NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER |>
  augment()|>
  ggplot(aes(x = Months))+
  geom_line(aes(y = `NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER`, color = "Data"))+
  geom_line(aes(y = .fitted, color = "Fitted"))
```

```
fcst9 <-
  fit_NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER |>
  forecast(h = 36)
fcst9
# A fable: 36 x 4 [1M]
# Key:     .model [1]
   .model   Months              NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER  .mean
   <chr>     <mth>                          <dist>  <dbl>
 1 arima  2024 Jan  N(2935, 1.9e+08)  2935.
 2 arima  2024 Feb N(11623, 1.9e+08) 11623.
 3 arima  2024 Mar N(13584, 1.9e+08) 13584.
 4 arima  2024 Apr N(10946, 1.9e+08) 10946.
 5 arima  2024 May N(11745, 1.9e+08) 11745.
 6 arima  2024 Jun  N(5334, 1.9e+08)  5334.
 7 arima  2024 Jul  N(4942, 1.9e+08)  4942.
 8 arima  2024 Aug N(24918, 1.9e+08) 24918.
 9 arima  2024 Sep  N(8679, 1.9e+08)  8679.
10 arima  2024 Oct N(22210, 1.9e+08) 22210.
# □ 26 more rows
```

Plot the Forecast Distribution

```
ggplot(data = fcst9, mapping = aes(x = Months, ydist = `NITROGEN, PHOSPHORUS AND POTASIUM
FERTILISER`, fill = .model))+
  stat_halfeye(alpha = .5)+
  xlab("Time (Months)")+
  ylab("NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER")+
  ggtitle("36 Months Forest using automatic ARIMA")
```

##################DAP##################

```
fit  DAP |>
  augment()|>
  ggplot(aes(x = Months))+
  geom_line(aes(y = `Diammonium Phosphate`, color = "Data"))+
  geom_line(aes(y = .fitted, color = "Fitted"))
```

```
fcst8 <-
  fit  DAP |>
  forecast(h = 36)
fcst8
# A fable: 36 x 4 [1M]
# Key:     .model [1]
   .model   Months `Diammonium Phosphate`  .mean
   <chr>     <mth>                 <dist>  <dbl>
 1 arima  2024 Jan       N(33102, 4e+08) 33102.
 2 arima  2024 Feb        N(8433, 4e+08)  8433.
 3 arima  2024 Mar       N(38551, 4e+08) 38551.
 4 arima  2024 Apr        N(8433, 4e+08)  8433.
 5 arima  2024 May        N(8524, 4e+08)  8524.
 6 arima  2024 Jun        N(8433, 4e+08)  8433.
 7 arima  2024 Jul        N(8433, 4e+08)  8433.
 8 arima  2024 Aug       N(20405, 4e+08) 20405.
 9 arima  2024 Sep       N(22718, 4e+08) 22718.
10 arima  2024 Oct       N(11500, 4e+08) 11500.
# □ 26 more rows
```

Plot the Forecast Distribution

```
ggplot(data = fcst8, mapping = aes(x = Months, ydist = `Diammonium Phosphate`, fill = .model))+
  stat_halfeye(alpha = .5)+
  xlab("Time (Months)")+
```

```
  ylab("DAP")+
  ggtitle("36 Months Forest using automatic ARIMA")
```

```
fcst2 <-
  fit  MP |>
  forecast(h = 36)
fcst2
# A fable: 36 x 4 [1M]
# Key:     .model [1]
   .model   Months `Muriate of Potash`  .mean
   <chr>     <mth>             <dist>  <dbl>
 1 arima  2024 Jan      N(93, 4300096)   93.0
 2 arima  2024 Feb    N(-545, 4521235) -545.
 3 arima  2024 Mar    N(1002, 5133300) 1002.
 4 arima  2024 Apr     N(887, 5868042)  887.
 5 arima  2024 May    N(1076, 5868525) 1076.
 6 arima  2024 Jun    N(1662, 5878022) 1662.
 7 arima  2024 Jul    N(1563, 5886242) 1563.
 8 arima  2024 Aug    N(1660, 5894573) 1660.
 9 arima  2024 Sep    N(1672, 5902894) 1672.
10 arima  2024 Oct    N(1673, 5911216) 1673.
# □ 26 more rows
```

Plot the Forecast Distribution

```
ggplot(data = fcst2, mapping = aes(x = Months, ydist = `Muriate of Potash`, fill = .model))+
  stat_halfeye(alpha = .5)+
  xlab("Time (Months)")+
  ylab("MP")+
  ggtitle("36 Months Forest using automatic ARIMA")
```

```
ggplot(data = fcst2, mapping = aes(x = Months, ydist = `Muriate of Potash`))+
  stat_halfeye(alpha = .5)+
  autolayer(filter_index(Fertiliser, "2010 Jan"~.))+
  xlab("Time (Months)")+
  ylab("Muriate of Potash")+
  ggtitle("24 Months Forest using Automatic ARIMA")
```

Plot the Fitted and the Data

```
fit_MP |>
  augment()|>
  ggplot(aes(x = Months))+
  geom_line(aes(y = `Muriate of Potash`, color = "Data"))+
  geom_line(aes(y = .fitted, color = "Fitted"))
```

```
fcst10 <-
  fit  Urea |>
  forecast(h = 36)
fcst10
# A fable: 36 x 4 [1M]
# Key:     .model [1]
   .model   Months            Urea .mean
   <chr>     <mth>          <dist> <dbl>
 1 arima  2024 Jan N(7994, 8.4e+07) 7994.
 2 arima  2024 Feb N(7994, 8.4e+07) 7994.
 3 arima  2024 Mar N(7994, 8.4e+07) 7994.
 4 arima  2024 Apr N(7994, 8.4e+07) 7994.
 5 arima  2024 May N(7994, 8.4e+07) 7994.
 6 arima  2024 Jun N(7994, 8.4e+07) 7994.
 7 arima  2024 Jul N(7994, 8.4e+07) 7994.
 8 arima  2024 Aug N(7994, 8.4e+07) 7994.
 9 arima  2024 Sep N(7994, 8.4e+07) 7994.
10 arima  2024 Oct N(7994, 8.4e+07) 7994.
# □ 26 more rows
```

Plot the Forecast Distribution

```
ggplot(data = fcst10, mapping = aes(x = Months, ydist = `Urea`, fill = .model))+
  stat_halfeye(alpha = .5)+
  xlab("Time (Months)")+
  ylab("Urea")+
  ggtitle("36 Months Forest using automatic ARIMA")
```

Plot the Fitted and the Data

```
fit_Urea |>
  augment()|>
  ggplot(aes(x = Months))+
  geom_line(aes(y = `Urea`, color = "Data"))+
  geom_line(aes(y = .fitted, color = "Fitted"))
```

```
fcst12 <-
  fit  Total |>
  forecast(h = 36)
fcst12
# A fable: 36 x 4 [1M]
# Key:     .model [1]
   .model   Months `total fertiliser`  .mean
   <chr>     <mth>           <dist>  <dbl>
 1 arima  2024 Jan  N(53813, 1.1e+09) 53813.
 2 arima  2024 Feb  N(68939, 1.2e+09) 68939.
 3 arima  2024 Mar  N(80802, 1.3e+09) 80802.
 4 arima  2024 Apr  N(67116, 1.3e+09) 67116.
 5 arima  2024 May  N(67080, 1.3e+09) 67080.
 6 arima  2024 Jun  N(54170, 1.3e+09) 54170.
 7 arima  2024 Jul  N(55182, 1.3e+09) 55182.
 8 arima  2024 Aug  N(57070, 1.4e+09) 57070.
 9 arima  2024 Sep  N(56238, 1.4e+09) 56238.
10 arima  2024 Oct  N(55731, 1.4e+09) 55731.
# □ 26 more rows
ggplot(data = fcst12, mapping = aes(x = Months, ydist = `total fertiliser`, fill = .model))+
  stat_halfeye(alpha = .5)+
  xlab("Time (Months)")+
  ylab("total fertiliser")+
  ggtitle("36 Months Forest using automatic ARIMA")
```

Plot the Fitted and the Data

```
fit  Total |>
  augment()|>
  ggplot(aes(x = Months))+
  geom_line(aes(y = `total fertiliser`, color = "Data"))+
  geom_line(aes(y = .fitted, color = "Fitted"))
```

Diagnostic Tests

Normality of the Residuals

```
residuals <- residuals(fit_MP)
# Plot histogram of residuals with a normal density curve
ggplot(residuals, aes(x = .resid)) +
  geom_histogram(aes(y = ..density..), bins = 30, fill = "steelblue", color = "blue", alpha = 0.7) +
  geom_density(color = "red") +
  stat_function(fun = dnorm, args = list(mean = mean(residuals$.resid),
                                         sd = sd(residuals$.resid)),
                color = "darkgreen", linetype = "dashed") +
  labs(title = "Muriate of Potash-Residuals Histogram with Normal Curve", x = "Residuals", y = "Density")
```

```
residuals <- residuals(fit_NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER)
# Plot histogram of residuals with a normal density curve
ggplot(residuals, aes(x = .resid)) +
  geom_histogram(aes(y = ..density..), bins = 30, fill = "steelblue", color = "blue", alpha = 0.7) +
  geom_density(color = "red") +
  stat_function(fun = dnorm, args = list(mean = mean(residuals$.resid),
                                         sd = sd(residuals$.resid)),
                color = "darkgreen", linetype = "dashed") +
```

```
  labs(title = "NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER-Residuals Histogram with Normal
Curve", x = "Residuals", y = "Density")
```

```
residuals <- residuals(fit_Urea)
# Plot histogram of residuals with a normal density curve
ggplot(residuals, aes(x = .resid)) +
  geom_histogram(aes(y = ..density..), bins = 30, fill = "steelblue", color = "blue", alpha = 0.7) +
  geom_density(color = "red") +
  stat_function(fun = dnorm, args = list(mean = mean(residuals$.resid),
                                         sd = sd(residuals$.resid)),
                color = "darkgreen", linetype = "dashed") +
  labs(title = "Urea-Residuals Histogram with Normal Curve", x = "Residuals", y = "Density")
```

```
residuals <- residuals(fit_DAP)
# Plot histogram of residuals with a normal density curve
ggplot(residuals, aes(x = .resid)) +
  geom_histogram(aes(y = ..density..), bins = 30, fill = "steelblue", color = "blue", alpha = 0.7) +
  geom_density(color = "red") +
  stat_function(fun = dnorm, args = list(mean = mean(residuals$.resid),
                                         sd = sd(residuals$.resid)),
                color = "darkgreen", linetype = "dashed") +
  labs(title = "DAP-Residuals Histogram with Normal Curve", x = "Residuals", y = "Density")
```

```
residuals <- residuals(fit  CN)
# Plot histogram of residuals with a normal density curve
ggplot(residuals, aes(x = .resid)) +
  geom_histogram(aes(y = ..density..), bins = 30, fill = "steelblue", color = "blue", alpha = 0.7) +
  geom_density(color = "red") +
  stat_function(fun = dnorm, args = list(mean = mean(residuals$.resid),
                                         sd = sd(residuals$.resid)),
                color = "darkgreen", linetype = "dashed") +
  labs(title = "Calcium Nitrate-Residuals Histogram with Normal Curve", x = "Residuals", y = "Density")
```

```
residuals <- residuals(fit_CAN)
# Plot histogram of residuals with a normal density curve
ggplot(residuals, aes(x = .resid)) +
  geom_histogram(aes(y = ..density..), bins = 30, fill = "steelblue", color = "blue", alpha = 0.7) +
  geom_density(color = "red") +
  stat_function(fun = dnorm, args = list(mean = mean(residuals$.resid),
                                         sd = sd(residuals$.resid)),
                color = "darkgreen", linetype = "dashed") +
  labs(title = "CAN-Residuals Histogram with Normal Curve", x = "Residuals", y = "Density")
```

```
residuals <- residuals(fit_Total)
# Plot histogram of residuals with a normal density curve
ggplot(residuals, aes(x = .resid)) +
  geom_histogram(aes(y = ..density..), bins = 30, fill = "steelblue", color = "blue", alpha = 0.7) +
  geom_density(color = "red") +
  stat_function(fun = dnorm, args = list(mean = mean(residuals$.resid),
                                         sd = sd(residuals$.resid)),
                color = "darkgreen", linetype = "dashed") +
  labs(title = "total fertiliser demand Histogram with Normal Curve", x = "Residuals", y = "Density")
```

QQ Plot

```
# Q-Q plot for residuals
ggplot(residuals, aes(sample = .resid)) +
  geom_qq() +
  geom_qq_line(color = "red") +
  labs(title = "Q-Q Plot of Residuals", x = "Theoretical Quantiles", y = "Sample Quantiles")
```

```
residuals6 <- residuals(fit_NITROGEN, PHOSPHORUS AND POTASIUM FERTILISER)
shapiro_test <- shapiro.test(residuals6$.resid)
shapiro_test
```

```
    Shapiro-Wilk normality test

data:  residuals6$.resid
W = 0.83217, p-value = 1.286e-12
```

QQ Plot

```
# Q-Q plot for residuals
ggplot(residuals, aes(sample = .resid)) +
  geom_qq() +
  geom_qq_line(color = "red") +
  labs(title = "Q-Q Plot of Residuals", x = "Theoretical Quantiles", y = "Sample Quantiles")
```

```
residuals5 <- residuals(fit  MP)
shapiro_test <- shapiro.test(residuals5$.resid)
shapiro_test

    Shapiro-Wilk normality test

data:  residuals5$.resid
W = 0.6356, p-value < 2.2e-16
# Q-Q plot for residuals
ggplot(residuals, aes(sample = .resid)) +
  geom_qq() +
  geom_qq_line(color = "red") +
  labs(title = "Q-Q Plot of Residuals", x = "Theoretical Quantiles", y = "Sample Quantiles")
```

```
residuals4 <- residuals(fit  Urea)
shapiro_test <- shapiro.test(residuals4$.resid)
shapiro_test

    Shapiro-Wilk normality test

data:  residuals4$.resid
W = 0.82573, p-value = 6.943e-13
# Q-Q plot for residuals
ggplot(residuals, aes(sample = .resid)) +
  geom_qq() +
  geom_qq_line(color = "red") +
  labs(title = "Q-Q Plot of Residuals", x = "Theoretical Quantiles", y = "Sample Quantiles")
```

```
residuals3 <- residuals(fit  DAP)
shapiro  test <- shapiro.test(residuals3$.resid)
shapiro_test

    Shapiro-Wilk normality test

data:  residuals3$.resid
W = 0.90764, p-value = 8.636e-09
# Q-Q plot for residuals
ggplot(residuals, aes(sample = .resid)) +
  geom_qq() +
  geom_qq_line(color = "red") +
  labs(title = "Q-Q Plot of Residuals", x = "Theoretical Quantiles", y = "Sample Quantiles")
```

```
residuals2 <- residuals(fit  CN)
shapiro_test <- shapiro.test(residuals2$.resid)
shapiro_test

    Shapiro-Wilk normality test

data:  residuals2$.resid
W = 0.50569, p-value < 2.2e-16
```

```
# Q-Q plot for residuals
ggplot(residuals, aes(sample = .resid)) +
  geom_qq() +
  geom_qq_line(color = "red") +
  labs(title = "Q-Q Plot of Residuals", x = "Theoretical Quantiles", y = "Sample Quantiles")
```

```
residuals1 <- residuals(fit  CAN)
shapiro_test <- shapiro.test(residuals1$.resid)
shapiro_test

    Shapiro-Wilk normality test

data:  residuals1$.resid
W = 0.74617, p-value = 9.886e-16
# Q-Q plot for residuals
ggplot(residuals, aes(sample = .resid)) +
  geom_qq() +
  geom_qq_line(color = "red") +
  labs(title = "Q-Q Plot of Residuals", x = "Theoretical Quantiles", y = "Sample Quantiles")
```

```
residuals <- residuals(fit  Total)
shapiro_test <- shapiro.test(residuals$.resid)
shapiro_test

    Shapiro-Wilk normality test

data:  residuals$.resid
W = 0.97112, p-value = 0.001414
```

Basic accuracy- split to Test and Train

Testing dataset

```
forecast_horizon <- 12
test <- Fertiliser |>
  filter_index(as.character(max(Fertiliser$Months)-forecast_horizon +1) ~ .)
```

Training data set

```
train <- Fertiliser |>
  filter_index(. ~ as.character(max(Fertiliser$Months)-forecast_horizon))
fit2 <- train |>
  model(automatic_arima = ARIMA(`total fertiliser`)
        )
```

Create the Forecast Using the Models Above

```
fcst2 <- fit2 |>
  forecast(h = forecast_horizon)
fcst2
# A fable: 12 x 4 [1M]
# Key:     .model [1]
   .model          Months `total fertiliser`  .mean
   <chr>            <mth>           <dist>  <dbl>
 1 automatic_arima 2023 Jan    N(57962, 1e+09) 57962.
 2 automatic_arima 2023 Feb    N(47066, 1e+09) 47066.
 3 automatic_arima 2023 Mar  N(58059, 1.1e+09) 58059.
 4 automatic_arima 2023 Apr  N(52663, 1.1e+09) 52663.
 5 automatic_arima 2023 May  N(46846, 1.1e+09) 46846.
 6 automatic_arima 2023 Jun  N(58027, 1.1e+09) 58027.
 7 automatic_arima 2023 Jul  N(53196, 1.1e+09) 53196.
 8 automatic_arima 2023 Aug  N(72906, 1.1e+09) 72906.
 9 automatic_arima 2023 Sep  N(63793, 1.1e+09) 63793.
10 automatic_arima 2023 Oct  N(61628, 1.1e+09) 61628.
11 automatic_arima 2023 Nov  N(67813, 1.1e+09) 67813.
12 automatic_arima 2023 Dec  N(58710, 1.1e+09) 58710.
```

Estimate Models' Accuracy Metrics

In this case, we use the accuracy() function to estimate the models' accuracy as shown below.

```
fcst_accuracy <- fcst2 |>
 accuracy(test,
          measures = list(point_accuracy_measures,
                          interval_accuracy_measures,
                          distribution_accuracy_measures))
fcst_accuracy
# A tibble: 1 × 15
  .model       .type    ME   RMSE    MAE   MPE  MAPE  MASE RMSSE  ACF1 winkler
  <chr>        <chr> <dbl>  <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl>
1 automatic_ar… Test  12418. 62548. 51934. -64.8  112.   NaN   NaN 0.685 495675.
# □ 4 more variables: pinball <dbl>, scaled_pinball <dbl>, percentile <dbl>,
#   CRPS <dbl>
```

The table give various accuracy metrics but we can just extract a few that we interested in, because we do not need everything here.

Extract a Few Accuracy Metrics of Interest

```
acc <-fcst_accuracy |>
 select(.model, RMSE, MAPE, winkler, CRPS)
acc
# A tibble: 1 × 5
  .model          RMSE  MAPE winkler   CRPS
  <chr>          <dbl> <dbl>   <dbl>  <dbl>
1 automatic_arima 62548.  112. 495675. 37527.
View(acc)
```

Appendix 4: Colours representing Years in plot

Year	Color
2010	
2011	
2012	
2013	
2014	
2015	
2016	
2017	
2018	
2019	
2020	
2021	
2022	
2023	