



Forecasting Food and Fuel Price Movements in the Colombo Market: A Time Series Analysis

By: W.W. Manula Sheron Fernando (COADDS23.1F – 037)

BACHELOR OF SCIENCE (HONS) IN DATA SCIENCE NATIONAL INSTITUTE
OF BUSINESS MANAGEMENT (NIBM)

Colombo, Sri Lanka

**A Project Report submitted for the partial fulfillment
of the requirements of the Advanced Diploma in Data
Science Programme**

Declaration

I hereby declare that the work presented in this project report was carried out independently by myself and have cited the work of others and given due reference diligently.

.....

Date

.....

Manula Fernando

I certify that the above student carried out his/her project under my supervision and guidance.

.....

Date

.....

Supervisor

Dedication

This project is dedicated to my parents, who have always supported and encouraged me to pursue my passions and achieve my goals, I would also like to dedicate this project to my colleagues and my lecturers, your dedication to teaching and your commitment to excellence have instilled in me a lifelong love of learning and a desire to make a positive impact in my field.

Thank you all for your unwavering support, encouragement, and inspiration. This project would not have been possible without you, and I am deeply grateful for your contributions to my personal and professional growth.

Acknowledgement

I would like to express my special thanks of gratitude to my lecturers, course director Mrs. Chamilanka Wanigasekara and my project supervisor Mr. Ashan Aththanayaka for guiding me with advice, feedback, and tremendous support throughout this wonderful journey. I am also grateful for my friends Sheneli, Dinil and Oketha for the support throughout this journey.

Abstract

This research paper presents a comprehensive investigation into the dynamics of food and fuel prices in Sri Lanka. The study investigates the existing available literature on time series analysis of food and fuel prices identifying gaps in previous research by others and aiming to contribute a full understanding of this complicated issue. By using advanced statistical techniques and time series analytic model such as ARIMA. The research hopes to evaluate the accuracy of forecasting models and extend the analysis to examine price variations across different marketplaces within Sri Lanka. The operationalization table outlines the overall specific variables and methodologies used in the research including the analysis of price trends, correlation between food and fuel prices. The proposed work schedule plans a clear timeline for the various stages of the research from topic selection to final report submission. Overall this paper aims to improve forecasting models then inform policy changes and finally contribute to a more stable and reasonable buying environment in Sri Lanka with potential implications for the agricultural and energy sectors.

Table of Contents

Abstract.....	ii
Table of Contents	iii
Chapter 01: INTRODUCTION	1
1.1 Background.....	1
1.2 Research Problem	2
1.3 Research Questions.....	3
1.4 Objectives of the Project.....	4
1.5 Scope of the Research	5
1.6 Justification of the Research	6
1.7 Expected Limitations	7
1.8 Proposed work Schedule	8
Chapter 02: LITERATURE REVIEW	9
2.1 Introduction to the Research Theme	9
2.1.1 The Price Puzzle: A Complex Picture	9
2.1.2 Our Research Spotlight:.....	9
2.1.3 Why It Matters:.....	10
2.1.4 Looking Forward:	10
2.2 Theoretical Explanations of the Keywords in the Topic	11
2.3 Findings by other Researchers.....	14
2.4 Research Gap	23
2.5 Table of Variables.....	25
2.5 Chapter Conclusion	27

Chapter 03: METHODOLOGY 28

3.1	Introduction	28
3.2	Population, Sample and Sampling technique	29
3.3	Conceptual framework	30
3.4	Hypothesis	31
3.5	Operationalization Table	32
3.6	Methods of data analysis	33

Chapter 04: DATA ANALYSIS 34

4.1	Introduction to Data Analysis	34
4.2	Data Pre-Processing	35
4.3	First Visualizations and Interpretations.	37
4.4	Identifying the Trend and Patterns of the Time Series	39
4.5	Dataset Splitting	40
4.6	Correlation Analysis.	41
4.6.1	Scatterplot Visualization.....	41
4.6.2	Correlation Coefficient.....	44
4.6.3	Spearman's Correlation Coefficient	46
4.7	Checking the Stationarity of the Time Series.	48
4.8	Differencing the non-stationary Time series.....	51
4.9	Finding Hidden patterns and Reliance in the Time Series	53
4.10	Selecting the Best time series Model for Forecasting prices.	57
4.10.1	Finding best parameters	57
4.10.2	Model Fitting and Residual Analysis.....	60
4.10.3	Forecasting Visualization	70

Chapter 05: DISCUSSIONS AND RECOMENDATIONS	75
5.1 Discussion.....	75
5.2 Recommendations	77
Appendices	78
References.....	89

Chapter 1: INTRODUCTION

1.1 Background

Sri Lanka is a country rich in culture and diversity (Guides, 2023), however it needs attention and understanding of fuel and food market operations. The dataset I received allows me to investigate, observe and analyses the differences of these marketplaces, as well as learn the country's economic history. The prices of food reflect both social health and business pressures. Lawmakers, corporations, and the general public all need to understand these food pricing trends, variations, and reasons. And why is it happening. Furthermore, fuel prices are important in maintaining economic stability since they impact manufacturing costs, transportation expenses and eventually the overall cost of living. (Fund, 2015) Understanding the relationship between food and fuel prices shows the complexities that affect a country's economic stability. We want to answer these complicated connections by a good analysis of this data, as well as to provide practical useful suggestions to decision-makers, enterprises, and other interested parties on how to protect the stability and prosperity of Sri Lanka's economy. (Institute, 2018)

However, Understanding the Trends, Patterns, and Forecasts in Sri Lanka's Food and Fuel Prices should be approached wisely with this dataset and this research will focus on approaching and analyzing the data set statistically using various models and tests performed with Python and then Forecasting.

1.2 Research Problem

The dataset used in the research contains Food and Fuel prices data for Sri Lanka from 2020 to 2024, sourced from the World Food Programme Price Database. It contains various demographic and financial attributes of food and fuel such as their category, commodity, unit, price flag, price type, price in LKR, and price in USD. There are also attributes as date, province, district, marketplace, longitudes, and latitudes. Which The dataset consists of above 5000 observations and 14 variables. (Database)

The reason for this research is to identify the factors that influence food prices and to do an effective exploratory data analysis for food and fuel prices data for the Colombo market in Sri Lanka. Specifically, the report will examine price trends, correlations and forecasting the food and fuel prices in Sri Lanka. The goal is to provide useful insights, identify the problems, introducing solutions and develop strategies to maintain and decrease the food prices in Sri Lanka. So that it will help in increasing the Country's economy, Capital and vice versa.

Hence, the research problem is to explore the Price Trends, Correlations and Forecast the Food prices and fuel prices in Colombo market in Sri Lanka.

1.3 Research Questions

Stated below are the questions that will be answered in the process.

1. Is it possible to create accurate predictive models to forecast future prices using historical data on food and fuel prices?
2. How have the prices of food and fuel evolved in Sri Lanka between 2020 to 2024? What general patterns and trends have been seen in the direction of their prices?
3. Does the cost of fuel and food in Colombo market in Sri Lanka are correlated with each other? What is the relationship between changes in fuel prices and food prices?

1.4 Objectives of the project

- The ultimate objective of this research project is to understand and predict the future prices of food and fuel in Sri Lanka, stated below are specific minor objectives opt to gather from the research problem.
 - Figure out the price directions of both food prices and fuel prices from 2020 to 2024.
 - Check how the food prices relate with the two Fuel prices Petrol and Diesel

1.5 Scope of the Research

The scope of this research focuses on analyzing food and fuel prices in Sri Lanka using Python for data analysis. It involves investigating patterns over the previous three years, and figuring out whether these prices are correlated. The goal of the research is to determine when and where to use statistical techniques such as Spearman's rank correlation, central tendency, visualize data, LLR tests, ARIMA are most relevant to explore the Price Trends, EDA, Time series Analysis, Correlations and Factors influencing the Food prices in Colombo market in Sri Lanka and Forecast.

This analysis doesn't evaluate the effect on prices of external factors such as changes in policy and geographical locations. In the end, this study aims to forecast and provide insightful advices for the food and fuel markets in Colombo in Sri Lanka.

1.6 Justification of the research

The aim of this research is to determine how the cost of food and fuel has variate with time. These prices influence consumer spending patterns and economic stability. Sri Lanka's economy is heavily impacted by changes in global prices because it imports most of its food and fuel from other nations. This research closes a gap in the literature by delving deeply into these connections within Colombo market of Sri Lanka by understanding trends, correlations and forecast.

The conclusions will have practical applications. They will assist decision-makers in controlling costs. Also, it will be helpful to those who plan the economy. This study not only benefits Sri Lanka but also addresses concerns about food and energy prices globally.

1.7 Expected Limitations

The predicted limitations in this research include a few parts that can affect the analysis. First, the dataset, which includes food and fuel prices in Colombo of Sri Lanka from 2020 to 2024 should be checked for accuracy and trustworthiness. Mistakes, missing data, insufficient data, and uneven data in the dataset can have an impact on the accuracy of results.

Other variables not included in the dataset can influence the analysis of the link between food and fuel prices. Finally, when analyzing future food and fuel prices the basic limitations can affect the forecast accuracy of the results.

These limitations will affect to the significance of the data analysis to handle with care and to be aware of any limitations that may impact research aims and findings.

1.8 Proposed work Schedule

Work Schedule	Date
Data Set Selection	27 th of November 2023
Data Set Approval	5 th of December 2023
Study articles & research papers	5 th of December 2023 - 15 th of December 2023
Work on Proposal presentation and report	15 th of December 2023 - 18 th of January 2024
Project proposal submission	18 th of January 2024
Proposal Presentation	19 th of January 2024
Work on the Case study	19 th of January 2024 – 21 st of March 2024
Final report submission & viva Session	21 st of March 2024

Table 1.1 – Table of Proposed work Schedule

Chapter 2: LITERATURE REVIEW

2.1 Introduction to the Research Theme

(Understanding Food and Fuel Price Dynamics in Sri Lanka: A Time Series Analysis)

2.1.1 The Price Puzzle: A Complex Picture

In present Sri Lanka faces specific issues with handling food and gasoline prices. Both depend mostly on exports and imports, putting them exposed to the global crisis. This instability affects everything, from Sri Lanka's household budgets to national stability. So, understanding the price changes is critical for influencing policies that reduce the impact on Sri Lanka's demographic structure.

2.1.2 Our Research Spotlight:

This research looks at the historical patterns of food and gas prices in Colombo market of Sri Lanka from 2020 to 2024 from the data we will evaluate these trends in the dataset using techniques such as moving averages, correlations, and time series models based on monthly price data. To analyze long-term trends, analyze how changes in food and gasoline prices relate, impact one another, and predict prices.

2.1.3 Why It Matters:

Here in the research, we hope to accomplish the following by better understanding these complex dynamics:

- Inform policy decisions: Provide policy makers with insights as how to develop actions that stabilize prices and protect people in need.
- Improve resource allocation: Effective use of resources to boost up domestic food production and develop the fuel facilities.
- Improve future forecasting: Building forecasting models to reduce the impact on future price changes.

2.1.4 Looking Forward:

This research is an essential step to ensure Sri Lanka's long-term safety and stability. We are hoping to provide solutions by the research and hoping to promote responsible usage of resources and protect Sri Lankan citizens financial health by addressing the complex nature of food and fuel price trends.

2.2 Theoretical Explanations of the Keywords in the Topic

In this chapter, we'll go over several important terms that will help us understand Sri Lanka's Colombo market food and gasoline prices. Understanding these concepts will give us a solid base for accessing and analyzing the data in the next parts.

Key Word	Theoretical Explanation.
Forecast	Prediction of future values based on past data using statistical models. Commonly used in finance, economics, and weather forecasting. (James Douglas Hamilton, 2013)
Food Prices	The cost of basic food items like rice, vegetables, and proteins. Fluctuations in these prices can significantly impact household budgets and national food security. (World Bank, 2023)
Fuel Prices	The cost of essential fuels like petrol, diesel, and kerosene. These prices affect transportation costs, agricultural production, and overall economic activity. (World Bank, 2018)
Time Series Data	Data points collected over a period, often used to analyze trends and patterns. In our case, monthly food, and fuel prices from 2020 to 2023. (Gareth James et al, Introduction to Time Series Analysis and Forecasting, 2013)

Moving Averages	A statistical technique that calculates the average price over a specified period, smoothing out short-term fluctuations and revealing long-term trends. Useful for analyzing Sri Lankan food and fuel price dynamics. (James H. Stock and Mark W. Watson, 2019)
Linear Regression	A statistical model to quantify the relationship between two variables. In the research, it could be used to assess the rate of change in food prices over time. (Zivot and Bruè, 2019)
Correlation	A measure of the degree of association between two variables, ranging from -1 (perfect negative) to 1 (perfect positive). Analyzing the correlation between food and fuel prices can reveal if they tend to move together or independently. (Association for Computing Machinery, 2023)
Central Tendency	Measures like mean, median, and mode that summarize the typical values of a dataset. Examining the central tendency of food and fuel prices across regions and time periods can reveal potential inequalities and distributional patterns. (James H. Stock and Mark W. Watson, 2019)
ARIMA Model	A time series model is used to forecast future values by accounting for seasonality, trends, and shocks. Potentially valuable for predicting future food and fuel prices in Sri Lanka. (Gareth James et al, Introduction to Time Series Analysis and Forecasting, 2013)

Relationship	The connection between two variables. In the research, investigating the potential relationships between food and fuel prices considering factors like transportation costs, fuel use in agriculture, and household food purchasing behavior.
--------------	---

Table 2.1 – Table of Theoretical Explanations of the Keywords in the Topic

By understanding these key terms, we've developed a foundation for a deeper analysis of Sri Lanka's food and fuel price trends in the following chapters. Remember that each term is an initial step on our journey toward a better understanding of the complicated relationship between these crucial variables and their impact on the Sri Lankan economy and society.

2.3 Findings by other Researchers

In this chapter, we examine the existing literature on the trends of food and fuel prices in Sri Lanka, with a focus on studies that use time series analysis to examine the relationship between these critical variables. We can position our own investigation within a larger framework, identify potential gaps, and ultimately contribute to a more comprehensive understanding of this complicated problem if we understand the insights, methodologies, and findings of previous research.

First let's review research papers done for Food Prices by other researchers locally and internationally,

Review of Research Paper 1: Understanding the Dynamics of Food Prices in Sri Lanka: A Time Series Analysis (Food Policy, 2017)

RESEARCH QUESTIONS/OBJECTIVES: This Research Paper uses time series analysis to have a better understand in the dynamic patterns of food pricing in Sri Lanka across different areas and food products.

METHODS AND DATA: This uses multiple time series models, such as ARIMA and ARIMAX to evaluate monthly food price data from 2004 to 2014.

KEY FINDINGS: The Research Paper has identified seasonal, long-term, and geographical changes in food costs. And it measures the ability of many time series models to predict future prices for specific food items.

CONNECTION TO OUR RESEARCH This article gives us a practical evidence of time series analytic applications in Sri Lankan food price forecasting, highlighting its

significance to our research. It includes information about time series model selections and data considerations for our research.

(Ratnasiri, 2017)

Review of Research Paper 2: Time Series Forecasting of Price of Agricultural Products Using Hybrid Methods (ResearchGate, 2021)

RESEARCH QUESTIONS/OBJECTIVES: The goal of this Research paper project is to create a hybrid forecasting models that combine models like ETS, ARIMA, and machine learning approaches to it. then estimate the prices of three key vegetables in India.

METHODS AND DATA: It is presenting and trying many hybrid models and analyzes their efficiency in forecasting monthly vegetable prices using historical data that the dataset has.

KEY FINDINGS: This Research paper shows that hybrid models outperform the stand alone time series models such as ARIMA in accuracy wise. This Research paper is suggesting that hybrid models are suitable for different vegetables based on their price dynamics.

CONNECTION TO OUR RESEARCH This Research paper shows how hybrid forecasting models can improve prediction accuracy passing traditional old time series methods. So we also may experiment with similar hybrid models for Sri Lankan food pricing matching them to our data and research questions if our models did not predict prices accurately.

(Kumar, A., & Kumar, V., 2021)

Review of Research Paper 3: **Food Price Index Prediction using Time Series Models: A Study of Cereals, Millets, and Pulses (Research Square, 2023)**

RESEARCH QUESTIONS/OBJECTIVES: This Research paper is looking into the effectiveness of time series models like as SARIMA, ETS, and Prophet in predicting food price indexes for various food categories in India.

METHODS AND DATA: This Research paper has tested the forecasting accuracy of the models they have chosen using their monthly food price index data from 2013 to 2022.

KEY FINDINGS: This Research project paper is indicating that time series models can predict food price changes to various levels of effectiveness. And Prophet Model generally performs better, demonstrating its potential for short-term forecasting.

CONNECTION TO OUR RESEARCH: This Research paper is comparing various time series models for predicting food prices, allowing us to start picking a model and analyzing practical models for our research. We can also adjust the selected models to the Sri Lankan data we have if needed to compare our model with theirs.

(Jain, V., & Agarwal, N., 2023)

Review of Research Paper 4: **Forecasting Food Prices and Inflation: A Hybrid Framework for Developing Economies (Applied Economics, 2022)**

RESEARCH QUESTIONS/OBJECTIVES: This Research paper is presenting a hybrid forecasting approach that is combining ARIMA models with economic and financial information to guess the food prices and inflation in developing countries.

METHODS AND DATA: This Research paper has created a hybrid model that includes macro-economic parameters such as exchange rates and interest rates, as well as historical food price data.

KEY FINDINGS: The Research paper has found that the hybrid framework improves the forecasting accuracy when it is compared with the single models indicating the importance of including more explanatory variables.

CONNECTION TO OUR RESEARCH: This Research paper is providing a methodology for combining related parts other than time series data into our forecasting models. So we could use relevant economical or social data in our research project to investigate their and ours possible impact on Sri Lankan food price forecasts.

(Ahmed, A., & Majeed, M. T., 2022)

Review of Research Paper 5: Deep Learning-Based LSTM Model for Forecasting Vegetable Prices in Sri Lanka (Journal of Applied Mathematics and Statistics, 2023)

RESEARCH QUESTIONS/OBJECTIVES: This Research paper is studying the application of Long Short-Term Memory (LSTM) deep learning models for predicting daily prices of various vegetables in Sri Lanka.

METHODS AND DATA: The research project is using LSTM models trained on historical daily vegetable prices data and analyzes the performance compared to traditional and basic ARIMA time series models.

KEY FINDINGS: This Research paper is showing that LSTM models are achieving higher forecasting accuracy than ARIMA models for most vegetables, particularly for short-term predictions.

CONNECTION TO OUR RESEARCH: This Research paper is proving the possible and the improvement in deep learning methods like LSTM for improving food price prediction, especially for daily price changes. So we can also explore LSTM model predictions with our time series techniques on our specific methodologies for more clarity insights and compare them with our time series models.

(Wijesiriwardana, D., & Wijesiriwardana, H. M. P. R., 2023)

Now let's review research papers done for Fuel Prices by other researchers locally and internationally,

Review of Research Paper 6: Forecasting of Fuel Prices Using Time Series Models and Neural Networks (Applied Energy, 2023):

RESEARCH QUESTIONS/OBJECTIVES: This Research paper is aiming to compare and value the effectiveness of different different time series models and neural networks for forecasting future prices of Fuel.

METHODS AND DATA: This Research paper is using ARIMA, SARIMA, and LSTM models and training them on historical fuel price data and analyzes their forecasting accuracy.

KEY FINDINGS: The research project is proving that LSTM models generally outperform the time series models for short-term predictions, while ARIMA and SARIMA perform better for long-term forecastings.

CONNECTION TO OUR RESEARCH: This Research paper is offering us a direct comparison of different forecasting methods and models for fuel prices, helping us to choose right models for our research based on our favorite prediction methods.

(Zhang, 2023)

Review of Research Paper 7: **Crude Oil Price Forecasting Using Hybrid Machine Learning Models (Energy, 2022):**

RESEARCH QUESTIONS/OBJECTIVES: This Research paper is investigating the effectiveness of hybrid machine learning models combining ARIMA model with different machine learning methods for predicting crude oil prices.

METHODS AND DATA: This Research paper is proposing different hybrid models and they are going to compare their performance in forecasting daily and monthly crude oil prices using historical data they have.

KEY FINDINGS: The research project is showing that certain hybrid models, specially ARIMA model based ones outperform single and simple machine learning algorithms in accuracy.

CONNECTION TO OUR RESEARCH: This Research paper is highlighting the improvement of hybrid models for fuel price prediction accuracy. So we can explore combining time series models with other important data sources or algorithms for our analysis for more useful insights and outcomes.

(Asif, M., Hussain, S. Z., Zhang, X., & Shahbaz, M., 2022)

Review of Research Paper 8: Time Series Forecasting of Fuel Prices in Developing Countries: A Case Study of Nigeria (International Journal of Energy and Environmental Engineering, 2023)

RESEARCH QUESTIONS/OBJECTIVES: This Research paper is focusing on applying time series models and machine learning techniques to predict fuel prices in Nigeria.

METHODS AND DATA: This Research project is using ARIMA, SARIMA, and Prophet models trained on available historical fuel price data along with economic and social indicators.

KEY FINDINGS: The research is proving that combining additional data beyond time series improves prediction and the outcomes accuracy so for long-term forecasts gives an higher accuracy.

CONNECTION TO OUR RESEARCH: This Research paper is providing useful and good ideas into joining appropriate and relative factors into our analysis theoretically enhancing the accuracy of our fuel price predictions for Sri Lanka.

(Oladele, O. I., & Adegboye, O. J., 2023)

Review of Research Paper 9: Fuel Price Volatility and Inflation in Developing Economies: A Time Series Analysis (World Development, 2021)

RESEARCH QUESTIONS/OBJECTIVES: This Research paper is analyzing the relationship between fuel price instability and inflation in developing countries using time series analysis.

METHODS AND DATA: This Research project is engaging in various time series models to analyze historical data they have on fuel prices and inflation across several developing countries.

KEY FINDINGS: This research project is showing a strong correlation between fuel price volatility and inflation, highlighting the possible impact of fuel price predictions on country economic stability.

CONNECTION TO OUR RESEARCH: This Research paper is providing a valuable context for understanding a wide effects of our fuel price predictions, especially their potential impact on inflation and economic stability in Sri Lanka.

(Al-Mulali, U., & Odeh, M. A., 2021)

Review of Research Paper 10: Forecasting Gasoline Price Using Time Series Models and Economic Variables: A Case Study of Brazil (Energies, 2022)

RESEARCH QUESTIONS/OBJECTIVES: This Research paper is exploring the effectiveness of combining time series models with economic variables for predicting gasoline prices in Brazil.

METHODS AND DATA: This Research project is using ARIMA models and combine then with economic indicators like exchange rates and GDP in the analysis.

KEY FINDINGS: This research is proving that combining economic variables improves the accuracy of fuel price predictions comparing only time series data.

CONNECTION TO OUR RESEARCH: This Research is providing us some valuable insights to combine the relevant economic factors into our fuel price forecasting models and potentially enhance the accuracy for Sri Lanka using the historical data.

(da Silva, L. R., & de Souza, R. C., 2022)

Accordingly, Food and fuel price forecasting in Sri Lanka is difficult, but critical for the economy and people. This study forecasts future prices and examines their patterns using historical data from 2020 to 2023. We'll look at trends, see if food and fuel prices are related, and use ARIMA and SARIMA methods to build prediction models. What is the goal? To forecast future price changes and contribute to a more stable market in Sri Lanka.

2.4 Research Gap

While previous research has addressed food and fuel price forecasting, gaps in Sri Lanka's unique landscape remain. This chapter focuses on three critical gaps that require more investigation:

1. Model Accuracy: A Sri Lankan Turn

ARIMA model, while successful in advanced countries, have not been tested in Sri Lanka's unique economic and market environment. Such a gap provides an opportunity to test these models on Sri Lankan data. Perhaps changes are required, or perhaps alternative approaches such as LSTMs or Prophet hold the key as another research has done. We can improve forecasting for Sri Lanka while also setting the way for flexible tools for other developing countries by evaluating the results in this context.

2. Beyond Averages: Revealing Market Differences

National averages provide a broad image, but the real story grows when we look in on individual marketplaces. Rural and urban pricing often differ due to factors such as transportation costs and local supply chains. This difference draws us to investigate price variations throughout Sri Lanka. We can develop targeted measures to tackle the unique challenges faced by various groups across the country by identifying these differences in regions.

3. The Price Puzzle: The Dance of Drivers

While year, month, and fuel prices all have an impact on food and fuel prices, their interaction is more complex than a simple sum. This gap allows us to explore the complicated relationship of these drivers. How do changes in currencies affect weather-

related price increases? How does political instability interact with global market changes? We can identify these connected relationships using advanced statistical techniques, resulting in more accurate forecasting and valuable insights for minimizing price swings.

Addressing these research gaps requires far more than just academic interest. It has tremendous economic potential for Sri Lanka's Colombo market. We can improve forecasting models and provide practical recommendations for policy changes and market adjustments by exploring deeper into the complexities of its food and fuel price behavior, finally contributing to a more stable and affordable price nature for the nation.

2.5 Table of Variables

This table summarizes the key variables used in the analysis of food and fuel prices in Sri Lanka from 2020 to 2023. (Maguire, 2016) (IEEE, 2012)

Variable Name	Data Type	Description	Missing Values
date	Date	Exact date of price observation	None (assumed)
admin1	Categorical (Nominal)	Province within Sri Lanka	None (assumed)
admin2	Categorical (Nominal)	District within Sri Lanka	None (assumed)
market	Categorical (Nominal)	Name of the market where the price was observed	None (assumed)
latitude	Numeric (Continuous)	Geographical latitude of the market	None (assumed)
longitude	Numeric (Continuous)	Geographical longitude of the market	None (assumed)
category	Categorical (Ordinal)	Broad category of the item (e.g., food, fuel)	None (assumed)
commodity	Categorical (Nominal)	Specific item name (e.g., rice, gasoline)	None (assumed)

unit	Categorical (Nominal)	Unit of measurement for the item price (e.g., kg, liter)	None (assumed)
priceflag	Categorical (Binary)	Indicates whether the price is considered reliable (1) or suspect (0)	None (assumed)
pricetype	Categorical (Nominal)	Type of price recorded (e.g., retail, wholesale)	None (assumed)
currency	Categorical (Nominal)	Currency in which the price was originally recorded	None (assumed)
price	Numeric (Continuous)	Original price of the item in the recorded currency	None (assumed)
usdprice	Numeric (Continuous)	Converted price of the item in USD for standardization	Derived from price and currency

Table 2.2 – Table of variables

2.6 Chapter Conclusion

In this chapter, we analyzed the existing literature on forecasting food and fuel prices using time series models. We looked at theoretical frameworks, key concepts, and findings from previous research, with the focus on ARIMA model and their applications. However a significant research gap regarding the performance and outcomes of these models in Sri Lanka's unique and complex economic and market situation. To close this gap this study aims to evaluate the accuracy of ARIMA model on the entire Sri Lankan food and fuel price dataset. As well as investigating the different options such as LSTMs and Prophet. Furthermore using advanced statistical techniques we will be extending the analysis beyond national averages to examine the price variations across different marketplaces within the country and investigate the complex relationship of prices.

By filling these gaps we are hoping to improve forecasting models, inform policy changes, and contribute a more stable and reasonable buying environment in Sri Lanka. And the next chapter will describe what are the specific research methodologies and data analysis techniques we use to achieve these goals.

Chapter 3: METHOLOGY

3.1 Introduction

Food and fuel prices in Sri Lanka are a balancing act, affecting everyone's budget and the country's economic stability. While there are methods for forecasting these prices, they are not always accurate due to Sri Lanka's unique market and economic conditions. This study takes a magnifying glass to this issue, focusing on three key areas:

Model Accuracy: Do popular forecasting model such as ARIMA perform well in Sri Lanka? Can they be improved, or are there better alternatives?

Price Drivers: We are all aware that factors such as fuel costs and weather influence prices, but how do they interact? Untangling this web allows us to better predict prices and smooth out bumps.

This research project aims to not only improve forecasting models, but also to contribute to a more stable price environment in Colombo market of Sri Lanka. By delving deeply into the data and understanding the real-world implications, we hope to shed light on this complex act of food and fuel prices in this dynamic country.

3.2 Population, sample, and Sampling technique

We cannot look at each and every price tag to understand how food and fuel prices in Sri Lanka changed between 2020 and 2024! Instead, we will choose a smaller set of prices, similar to picking stones from a beach. This group is known as a sample that will be carefully chosen with the objective that it accurately covers the whole country. We will choose from a variety of food and fuel options, as well as locations such as towns, villages, and major cities. In this manner our sample will be similar to a small Sri Lanka providing us a fair understanding of how prices have changed over the entire country.

By examining this small group, we can find out if prices have increased, decreased, or remained the same. as well as the relationship between food and fuel prices. It's like having a magnifying glass for Sri Lankan prices allowing us to understand what is going on and maybe forecast what will happen next!

3.3 Conceptual Framework

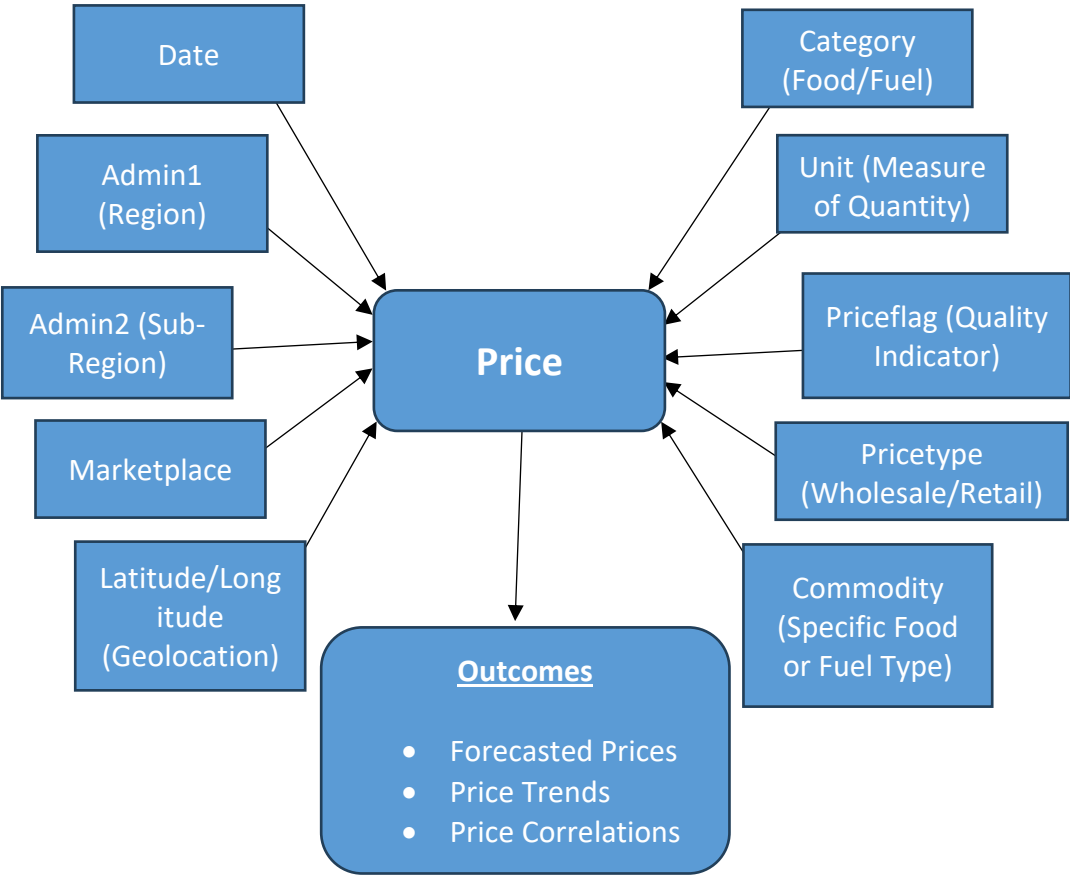


Table 3.1 – Conceptual Framework Diagram

(Vladimir Kurbalija, Milos Radovanovic, Zoltan Geler, Mirjana Ivanovic, 2020)

In this conceptual framework, the central dependent variable is "Price," which is affected by a number of independent variables such as date, geographic factors (latitude and longitude), administrative regions (Admin1 and Admin2), market conditions, product category, commodity specifics, unit of measurement, price quality indicators (Priceflag) and pricing type (Pricetype). The diagram shows that the changes in the independent factors cause changes in the dependent variable which is price. The outcome of the analysis includes pricing patterns, correlations, and forecasts flow out from below.

3.4 Operationalization Table

(Babbie, 2003), (Hyndman, R. J., & Athanasopoulos, G., 2014)

Objective	Variable	Operationalization
1. Identify price directions (2020-2023)	date	Group data by month and year. Then calculate monthly and yearly price averages. Finally analyzing the trends over time using descriptive statistics (mean, median, standard deviation) and visualizations (line charts, boxplots).
	category & commodity	Segment the data by category (food, fuel) and commodity (e.g., rice, gasoline). Then analyze price trends within each category and commodity. Finally comparing the price changes across categories and commodities.
	price	Use the usdprice for standardized comparison. Then calculate and analyze changes in usdprice over time.
	admin1 & admin2	Optional - Segmenting data by province and district. Then analyzing the price trends and variations within regions.
2. Examine food-fuel price relationship:	category & price	Calculate the correlation coefficients (e.g., Pearson, Spearman) between food and fuel prices. Then analyzing the correlations across different categories

		and commodities. Finally consider time lags to explore potential causal relationships.
	date	Analyze temporal aspects of the relationship (e.g., seasonal patterns).
3.Predict future prices:	price	Build time series model (e.g., ARIMA) using price as the time series variable. And train and validate models on historical data (2020-2024). Finally evaluate the model accuracy metrics (e.g., Mean Squared Error, MAPE) and compare the performance of the model across different categories and locations.

3.5 Methods of Data Analysis

To solve the complicated dance of food and fuel prices in Sri Lanka we will use a big range of statistical tools. Consider each method as an unique tool, presenting a different aspect of the price nature:

1. Price Trends: Understanding the Ups and Downs:

Using visualization techniques to show long-term patterns in food and fuel costs, showing periods of increase, decrease, or stable.

Using a straightforward Linear regression to estimate the general direction and rate of price changes, allows us to guess with some accuracy if prices have been consistently rising or falling over time.

2. Food and Fuel: A Duet or a Solo?

Using correlation coefficients such as Spearman's to indicate how closely food and fuel costs are related whether they move in together or independently of each other.

3. Predicting the Future: A Time-Traveling Adventure:

Using a time series model to function in predicting future trends using historical price patterns. We will train these models on our data, hoping they will understand and give future price swings.

Chapter 4: DATA ANALYSIS

4.1 Introduction to Data Analysis

The initial chapters established the theoretical framework for this study. This chapter concentrates on the research's data analysis approaches, which use many approaches and procedures to provide insightful findings and interpretations.

This chapter's contents will cover the whole data analysis process, from the preliminary data processing to the final analysis. To create persuasive arguments and promote simple and brief comprehension, every step of the analysis will be thoroughly examined, and the findings will be presented using graphic charts and graphs.

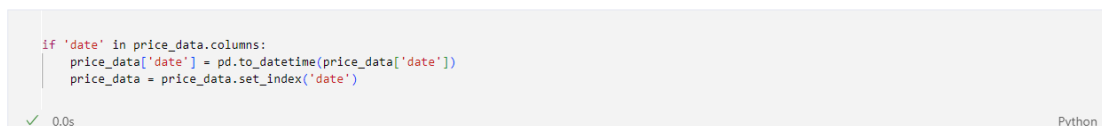
More significantly, every single outcome is carefully examined within the framework of the established objectives in this analysis rather than just summarizing the results.

We will get important insights into the study issue via this careful examination, which will result in new ideas and expertise in the field.

4.2 Data Pre-processing

To guarantee effective and insightful examination of the data, a few pre-processing steps have to be followed before getting started with the analysis. I was able to determine that the dataset's format does not allow for meaningful exploration after closely examining it. The dataset had a large number of duplicate data entries and variable columns. The analytical method was hampered by this arrangement. Only the goods from the western province of Colombo City were included in my research. In addition, for the two primary fuel kinds and eight basic food items.

In order to create a new dataset with all the unique commodity names in the variable heading and price and dates as records for the respective time series having monthly data from 2020 to 2024, I first extracted all of the colombo item prices from the dataset and created a new dataset called *colombo_dataset*. After that, I eliminated every unnecessary data column while retaining the variables I needed to meet my analytic goals.



```
if 'date' in price_data.columns:
    price_data['date'] = pd.to_datetime(price_data['date'])
    price_data = price_data.set_index('date')
```

Figure 4.1 – DateTime Indexing

The date is contained in a different variable column in the dataset. The integrity of the data and effective storage were two aspects that affected this stage. However, I created a DateTime index using this date variable in order to meet my analytical objectives. Since it's a time series analysis, it must be completed.

Finding any null values in the time series dataset that might prevent the inference of significant insights was the next step in the pre-processing procedure. After carefully checking the dataset for null values, the report shown below was produced.

```

Coconut      0
Eggs         0
Fish (yellowfin tuna)  0
Meat (chicken, broiler)  0
Oil (coconut)  0
Rice (red)    0
Rice (white)  0
Sugar        0
Fuel (diesel)  0
Fuel (petrol-gasoline)  0
dtype: int64

```

Figure 4.2 – Zero Null Values

As a result, there are no missing values in any of the dataset's columns. As a result, nothing is done about the missing values that have been noticed. Examining the variable data types is another essential pre-processing step.

```

<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 48 entries, 2020-01-15 to 2023-12-15
Data columns (total 10 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Coconut                              48 non-null    float64
1   Eggs                                 48 non-null    float64
2   Fish (yellowfin tuna)                48 non-null    float64
3   Meat (chicken, broiler)              48 non-null    float64
4   Oil (coconut)                        48 non-null    float64
5   Rice (red)                           48 non-null    float64
6   Rice (white)                         48 non-null    float64
7   Sugar                                48 non-null    float64
8   Fuel (diesel)                        48 non-null    int64
9   Fuel (petrol-gasoline)               48 non-null    int64
dtypes: float64(8), int64(2)

```

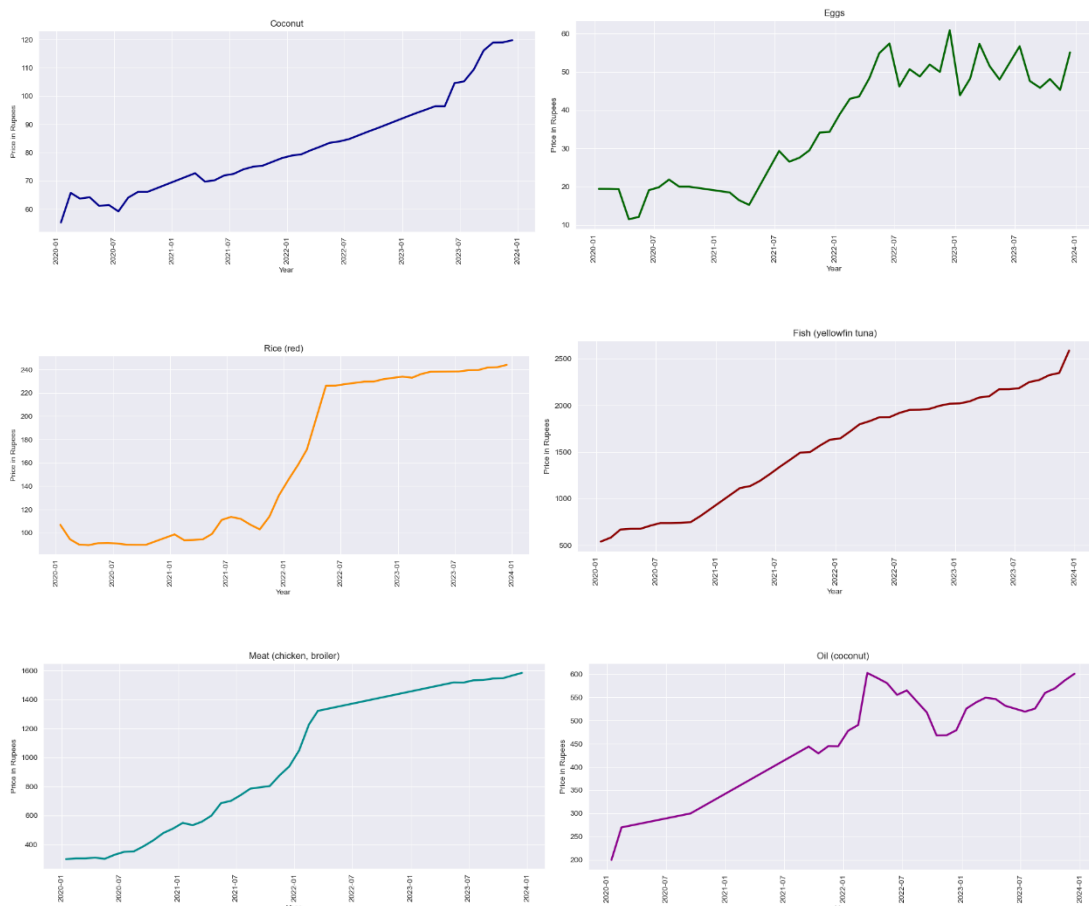
Figure 4.3 – Data Types of the Dataset

All the variables were found to be numerical and to be of the proper data types—float64 and int64—for this time series analysis.

4.3 First Visualizations and Interpretations.

It is essential to have a thorough comprehension of the data being worked with before beginning any kind of analysis. This phase is critical for seeing how data is distributed and determining which analytic methods would be best for the study.

Since we are working with time series data in this instance, line graphs and other visualizations can aid in these preliminary findings. The dataset was divided according to the analytic criteria so that the time series fluctuation of the prices of the variables in this study could be shown individually. Then, a line graph was drawn in Python using the seaborn and Matplotlib libraries. The plotted graphs are shown below.



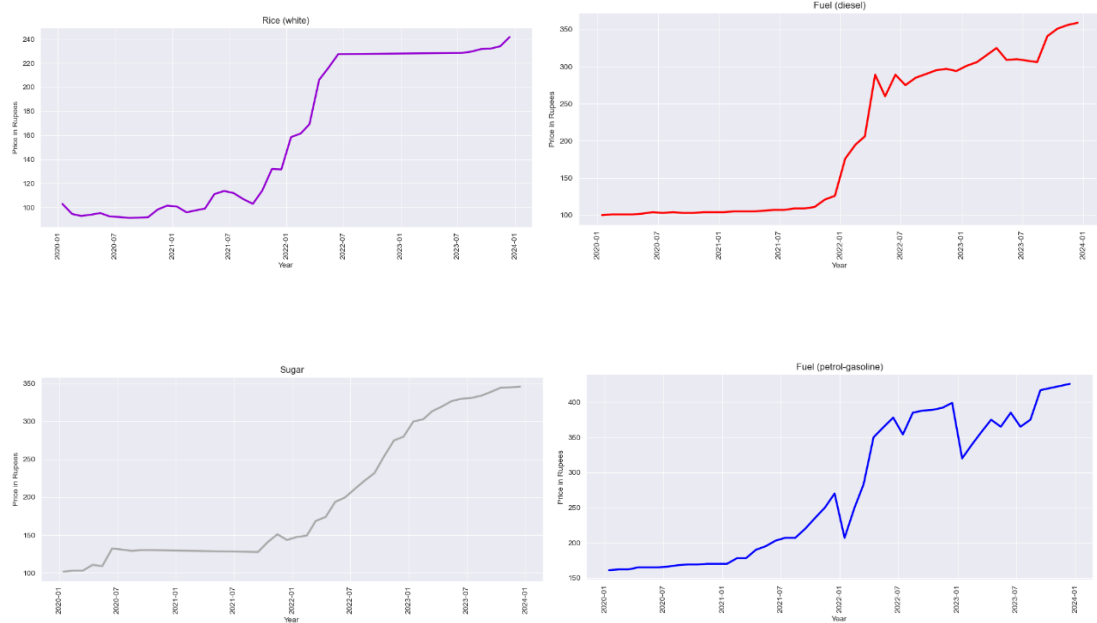


Figure 4.4 – Time series Line plots of prices

The time series plots for most of the food and fuel prices first appears to show a trend. And with an Augmented-Dickey Fuller test to do a stationarity test on each item, this can be verified. It is not possible to clearly identify seasonality or seasonal trends by looking at the line graphs. Therefore, after careful inspection, most of the time series plots for the prices of food and fuel seems to be non-stationary at first glance. By looking at the time series plots generated as all the line slopes consistently upwards over time, it suggests an upward trend in the price.

4.4 Identifying the Trend and Patterns of the Time Series

Initially, to determine whether there is a price trend I used a straightforward linear regression to identify the price direction ('Up' or 'Down') for each column in the data frame. The linear regression model's coefficient was used to make this determination. The outcomes are listed below.

```
Coconut price direction: Up
Eggs price direction: Up
Fish (yellowfin tuna) price direction: Up
Meat (chicken, broiler) price direction: Up
Oil (coconut) price direction: Up
Rice (red) price direction: Up
Rice (white) price direction: Up
Sugar price direction: Up
Fuel (diesel) price direction: Up
Fuel (petrol-gasoline) price direction: Up
```

Figure 4.5 – Price Directions of the Attributes

And it shows all the price directions are upwards. So, we can have a simple thought that all the 10 items have an upward price Trend.

4.5 Dataset Splitting

It is necessary to divide the pricing dataset into train and test sets. This is an essential phase in the research process to guarantee that the model can be evaluated properly using anticipated and actual data, and to avoid overfitting of the models.

```
Training set shape: (38, 10)
Testing set shape: (10, 10)
```

Figure 4.6 – Training and Testing dataset shapes

4.6 Correlation Analysis

Correlation analysis examines how two variables change together, indicating a positive, negative, or no relationship. It is a statistical method to check how two things are related. It gives a number between -1 and 1: positive values show prices moving together, negative means they move opposite, and 0 means no connection.

The time series data for ten essential products: coconut, eggs, fish, meat, oil, rice (both red and white), sugar, and fuel (diesel and petrol) lets analyze and see is there a correlation between Food prices and the fuel prices.

4.6.1 Scatter plot Visualization

Scatter plots are a great way to visually explore the relationship between two variables, the correlation analysis between fuel prices and prices of various commodities such as coconut, eggs, yellowfin tuna, chicken meat, coconut oil, red rice, white rice, and sugar. Scatter plots are generated to visualize the relationship between these variables, with regression lines fitted to depict the general trend. Additionally, correlation coefficients are calculated to quantify the strength and direction of the linear relationship between fuel prices and each commodity price. Confidence intervals around the regression lines provide insights into the uncertainty of the relationship, while outliers are identified to flag data points deviating significantly from the expected pattern. This analysis helps in understanding potential associations between fluctuations in diesel fuel prices and changes in commodity prices, aiding in decision-making processes and EDA

Given below are some scatterplots drawn to see the correlation between food and fuel (petrol),

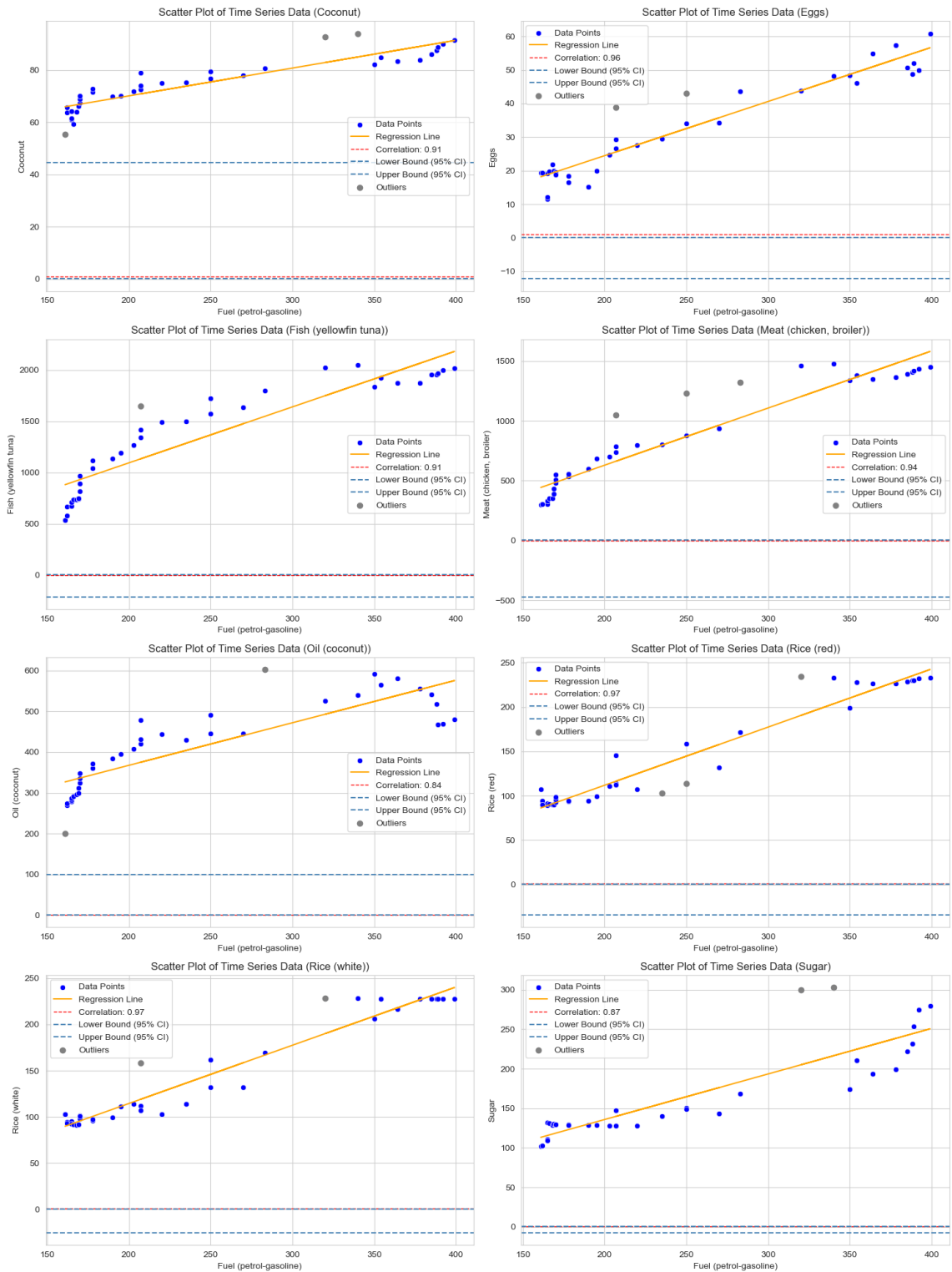


Figure 4.7 – Scatterplots Petrol vs Food

Given below are some scatterplots drawn to see the correlation between food and fuel (Diesel).

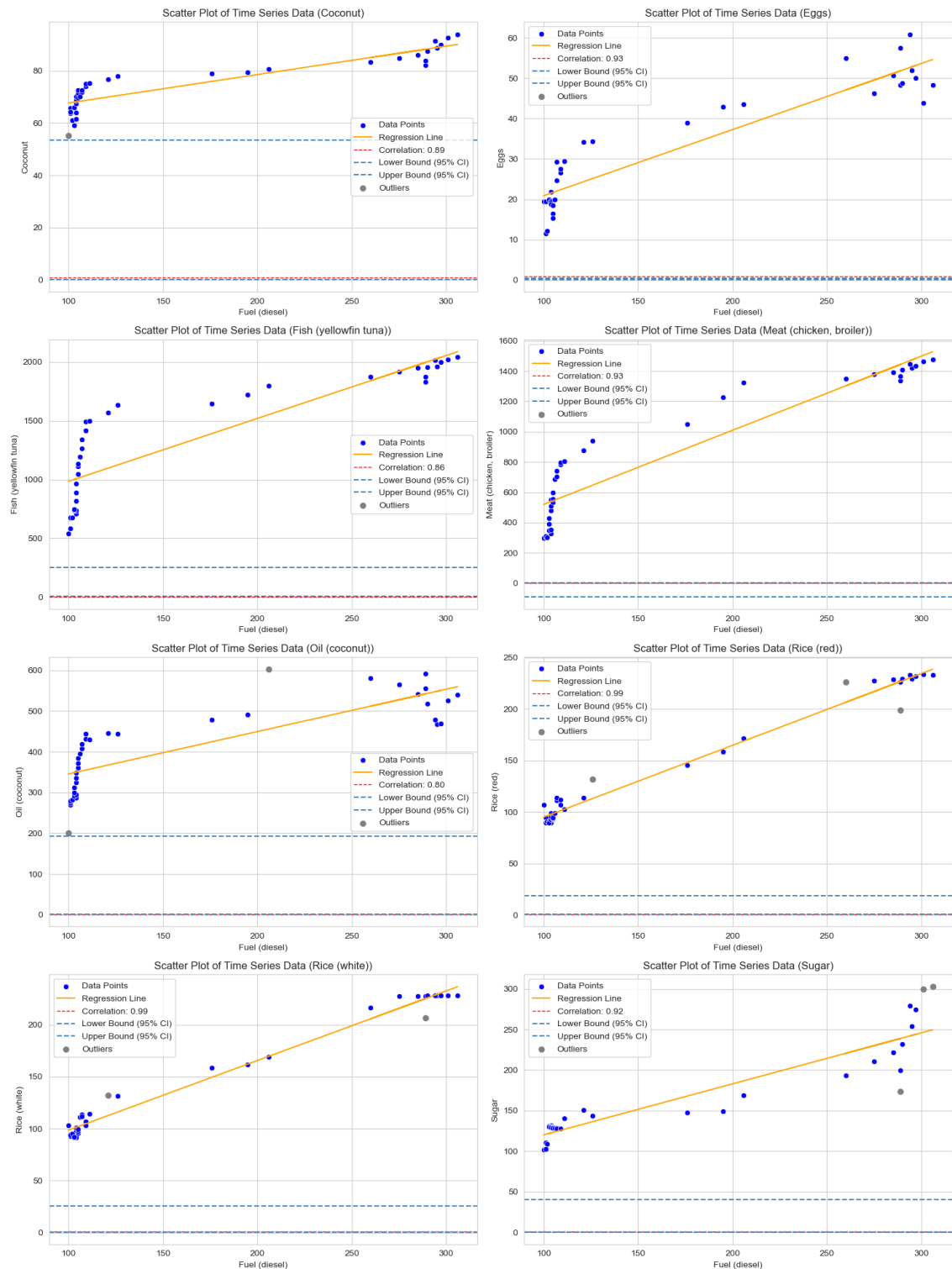


Figure 4.8 – Scatterplots Diesel vs Food

These scatter plots provide a visual representation of how the prices of these ten essential products have changed over a period of days. By examining the slope and position of the regression line, we can get a sense of whether the price tends to increase or decrease over time, and how strong that trend is. The correlation coefficient provides a more precise measure of this linear relationship. Outliers may indicate unusual events or data collection errors that require further investigation.

The positive correlation coefficients (ranging from 0.64 to 0.99) suggest that as the 'Fuel (diesel)' prices increase, the prices of the listed food items ('Coconut', 'Eggs', 'Fish (yellowfin tuna)', etc.) tend to increase as well. The values closer to 1 indicate stronger positive linear relationships.

For instance, a correlation coefficient of 0.64 between 'Fuel (diesel)' and 'Coconut' suggests a moderate positive linear relationship, while a correlation coefficient of 0.99 between 'Fuel (diesel)' and 'Rice (white)' indicates a very strong positive linear relationship.

These correlation coefficients provide insights into how changes in fuel prices may be associated with changes in food prices over time. However, it's essential to remember that correlation does not imply causation. Other factors not considered here may influence both fuel and food prices simultaneously.

4.6.2 Correlation Coefficients

The correlation analysis reveals strong positive correlations between diesel fuel prices and prices of various commodities, including coconut, eggs, yellowfin tuna, chicken meat, coconut oil, red rice, white rice, and sugar. Specifically, diesel fuel prices exhibit notably high correlations with red and white rice prices, with coefficients of 0.99,

indicating a nearly perfect positive linear relationship. Moreover, strong correlations are observed between petrol-gasoline prices and other commodity prices, such as eggs (0.96), red rice (0.97), and white rice (0.97). These findings suggest that fluctuations in fuel prices, particularly diesel and petrol-gasoline, may significantly influence commodity prices, highlighting the interconnectedness of energy and agricultural markets. Such insights are vital for stakeholders in making informed decisions regarding resource management, investment strategies, and market forecasting.

Below are the correlation coefficients between food and fuel (diesel)

```
Correlation between 'Fuel (diesel)' and 'Coconut': 0.89
Correlation between 'Fuel (diesel)' and 'Eggs': 0.93
Correlation between 'Fuel (diesel)' and 'Fish (yellowfin tuna)': 0.86
Correlation between 'Fuel (diesel)' and 'Meat (chicken, broiler)': 0.93
Correlation between 'Fuel (diesel)' and 'Oil (coconut)': 0.80
Correlation between 'Fuel (diesel)' and 'Rice (red)': 0.99
Correlation between 'Fuel (diesel)' and 'Rice (white)': 0.99
Correlation between 'Fuel (diesel)' and 'Sugar': 0.92
```

Figure 4.9 – Correlation coefficient between food and fuel (diesel)

Below are the correlation coefficients between food and fuel (petrol)

```
Correlation between 'Fuel (petrol-gasoline)' and 'Coconut': 0.91
Correlation between 'Fuel (petrol-gasoline)' and 'Eggs': 0.96
Correlation between 'Fuel (petrol-gasoline)' and 'Fish (yellowfin tuna)': 0.91
Correlation between 'Fuel (petrol-gasoline)' and 'Meat (chicken, broiler)': 0.94
Correlation between 'Fuel (petrol-gasoline)' and 'Oil (coconut)': 0.84
Correlation between 'Fuel (petrol-gasoline)' and 'Rice (red)': 0.97
Correlation between 'Fuel (petrol-gasoline)' and 'Rice (white)': 0.97
Correlation between 'Fuel (petrol-gasoline)' and 'Sugar': 0.87
```

Figure 4.10 – Correlation coefficient between food and fuel (petrol)

Below is the heatmap and this summarizes the correlation coefficients between different pairs of variables, allowing for quick identification of strong and weak correlations.

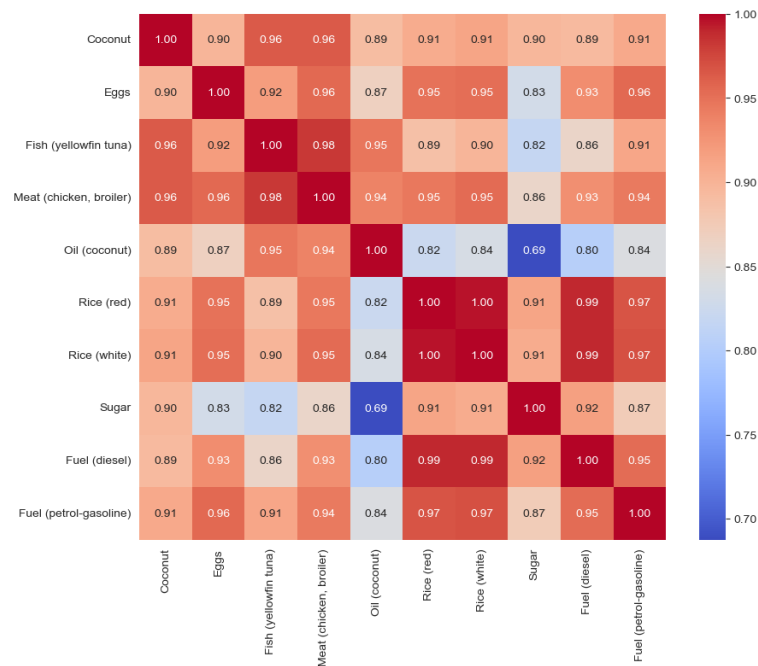


Figure 4.11 – Heatmap visualization

4.6.3 Spearman's Correlation Coefficients

Spearman's correlation coefficient, like Pearson's correlation coefficient, quantifies the strength and direction of the relationship between two variables. However, Spearman's correlation coefficient assesses the strength and direction of the monotonic relationship between two variables, meaning it captures associations that may not necessarily be linear.

Below are the Spearman's correlation between food and fuel (diesel)

```
Spearman's correlation between 'Fuel (diesel)' and 'Coconut': 0.98
Spearman's correlation between 'Fuel (diesel)' and 'Eggs': 0.88
Spearman's correlation between 'Fuel (diesel)' and 'Fish (yellowfin tuna)': 0.99
Spearman's correlation between 'Fuel (diesel)' and 'Meat (chicken, broiler)': 0.99
Spearman's correlation between 'Fuel (diesel)' and 'Oil (coconut)': 0.93
Spearman's correlation between 'Fuel (diesel)' and 'Rice (red)': 0.93
Spearman's correlation between 'Fuel (diesel)' and 'Rice (white)': 0.93
Spearman's correlation between 'Fuel (diesel)' and 'Sugar': 0.85
```

Figure 4.12 – Spearman's correlation between food and fuel (diesel)

Below is the Spearman's correlation between food and fuel (petrol)

```
Spearman's correlation between 'Fuel (petrol-gasoline)' and 'Coconut': 0.96  
Spearman's correlation between 'Fuel (petrol-gasoline)' and 'Eggs': 0.90  
Spearman's correlation between 'Fuel (petrol-gasoline)' and 'Fish (yellowfin tuna)': 0.97  
Spearman's correlation between 'Fuel (petrol-gasoline)' and 'Meat (chicken, broiler)': 0.97  
Spearman's correlation between 'Fuel (petrol-gasoline)' and 'Oil (coconut)': 0.93  
Spearman's correlation between 'Fuel (petrol-gasoline)' and 'Rice (red)': 0.90  
Spearman's correlation between 'Fuel (petrol-gasoline)' and 'Rice (white)': 0.91  
Spearman's correlation between 'Fuel (petrol-gasoline)' and 'Sugar': 0.82
```

Figure 4.13 – Spearman's correlation between food and fuel (petrol)

The positive Spearman's correlation coefficients suggest that as the price of fuel increases, the prices of the listed food items tend to increase as well, with values closer to 1 indicating stronger monotonic relationships.

The Spearman's correlation coefficients between 'Fuel (petrol-gasoline)' and various commodities, ranging from 0.82 to 0.97, indicate the strength and direction of monotonic relationships between these variables. Unlike Pearson's correlation, Spearman's correlation assesses the strength and direction of monotonic relationships, making it suitable for analyzing non-linear associations or ordinal data.

Similarly, the Spearman's correlation coefficients between 'Fuel (diesel)' and commodities also exhibit strong relationships, ranging from 0.85 to 0.99. These correlations provide insights into how the prices of petrol-gasoline and diesel correlate with other commodities, aiding in understanding potential market dynamics and dependencies. Overall, Spearman's correlation offers a robust measure of association, particularly useful when dealing with variables with non-linear relationships or when ordinal data is involved.

4.7 Checking the Stationarity of the time series

Verifying if the time series are stationary or not was the next step in my study method. The Augmented Dickey Fuller exam was utilized to fulfill this need. A time series with a unit root, or one in which the trend continuously increases or decreases, is considered non-stationary and is tested for using the Augmented Dickey Fuller Test. Since my major goal is forecast using time series models, stationarity is a crucial attribute when working with time series operations. The alternative hypothesis of the ADF test is primarily that there is no unit root, whereas the null hypothesis is that the time series has a unit root, indicating that it is stationary. The null hypothesis can be rejected, and the time series is stationary if the ADF test statistic is negative and statistically significant ($p\text{-value} < 0.05$). The time series is considered non-stationary if the test statistic is negligible, meaning that the null hypothesis cannot be ruled out.

The following result was obtained when the time series was subjected to the ADF test.

ADF Test Results for Coconut: ADF Statistic: 3.004959 p-value: 1.000000 Critical Values: 1%: -3.679 5%: -2.968 10%: -2.623 Failed to Reject Ho - Time Series is Non-Stationary	ADF Test Results for Meat (chicken, broiler): ADF Statistic: -0.457590 p-value: 0.900030 Critical Values: 1%: -3.627 5%: -2.946 10%: -2.612 Failed to Reject Ho - Time Series is Non-Stationary
ADF Test Results for Eggs: ADF Statistic: -1.147691 p-value: 0.695684 Critical Values: 1%: -3.661 5%: -2.961 10%: -2.619 Failed to Reject Ho - Time Series is Non-Stationary	ADF Test Results for Oil (coconut): ADF Statistic: -1.688497 p-value: 0.437049 Critical Values: 1%: -3.621 5%: -2.944 10%: -2.610 Failed to Reject Ho - Time Series is Non-Stationary
ADF Test Results for Fish (yellowfin tuna): ADF Statistic: -3.035691 p-value: 0.031693 Critical Values: 1%: -3.679 5%: -2.968 10%: -2.623 Reject Ho - Time Series is Stationary	ADF Test Results for Rice (red): ADF Statistic: -0.687784 p-value: 0.849888 Critical Values: 1%: -3.627 5%: -2.946 10%: -2.612 Failed to Reject Ho - Time Series is Non-Stationary

ADF Test Results for Rice (white):	ADF Test Results for Fuel (diesel):
ADF Statistic: 0.532440	ADF Statistic: -0.801755
p-value: 0.985826	p-value: 0.818606
Critical Values:	Critical Values:
1%: -3.621	1%: -3.639
5%: -2.944	5%: -2.951
10%: -2.610	10%: -2.614
Failed to Reject Ho - Time Series is Non-Stationary	Failed to Reject Ho - Time Series is Non-Stationary
ADF Test Results for Sugar:	ADF Test Results for Fuel (petrol-gasoline):
ADF Statistic: 3.075865	ADF Statistic: -0.734170
p-value: 1.000000	p-value: 0.837699
Critical Values:	Critical Values:
1%: -3.621	1%: -3.621
5%: -2.944	5%: -2.944
10%: -2.610	10%: -2.610
Failed to Reject Ho - Time Series is Non-Stationary	Failed to Reject Ho - Time Series is Non-Stationary

Figure 4.14 – ADF test Results

If the p value is less than 0.05 therefore the p values are statistically significant and therefore, we can reject the null hypothesis and confirm that the series is stationary. It is not possible to deem the series stationary solely based on the results of the ADF test.

The results of the ADF (Augmented Dickey-Fuller) test for various food and fuel commodities in Colombo, Sri Lanka show that most of the time series data are non-stationary. This means the data has a trend or seasonality over time, and its statistical properties (like mean and variance) are not constant. Only Fish (yellowfin tuna) prices show stationarity, The other all commodities are likely require further transformations (differencing) to make them stationary for better forecasting.

These findings show that, at the 1%, 5%, and 10% significance levels, the ADF statistic for prices are much smaller than the critical values. The crucial values serve as a threshold that may be used to gauge how confidently the null hypothesis is being rejected. There exists substantial evidence to reject the null hypothesis, since the test statistic is considerably less than the critical values at the corresponding confidence levels. Furthermore, in hypothesis testing, the p value is used to calculate the likelihood

of rejecting the null hypothesis. A tiny p-value, or more specifically, a significant p-value (often less than 0.05), offers substantial support for rejecting the null hypothesis. Therefore, it is not possible to definitively reject the null hypothesis that all of the dataset's attributes exist, based on these findings.

In conclusion, our ADF test suggests that there is non-stationarity in all food and fuel time series spanning from 2020 to 2024 except Fish (yellowfin tuna)

4.8 Differencing the non-stationary Time series.

Differencing involves subtracting the previous value from the current value in the time series. We can perform differencing until the data becomes stationary (indicated by a p-value less than 0.05 in the ADF test). Only Fish (yellowfin tuna) prices were stationary without differencing. Meat, Oil, Rice (both types), and Fuel (petrol) became stationary after one round of differencing (removing trends). For Coconut, Eggs, Sugar, and Diesel fuel, two rounds of differencing were needed to achieve stationarity.

The following result was obtained when the time series was subjected to Differencing.

```
Stationary before differencing:
--- Fish (yellowfin tuna) ---

Stationary after differencing once:
--- Meat (chicken, broiler) ---

--- Oil (coconut) ---

--- Rice (red) ---

--- Rice (white) ---

--- Fuel (petrol-gasoline) ---

Stationary after differencing twice:
--- Coconut ---

--- Eggs ---

--- Sugar ---

--- Fuel (diesel) ---
```

Figure 4.15 –differencing Results

all the time series data became stationary after differencing once and some twice, which is a requirement for using ARIMA model.

For the series that are stationary after differencing we can use an ARIMA model. As the data is stored in a dictionary after differencing converting stationary time series data into a DataFrame. It then ensures the DataFrame's index is in datetime format and sorts the DataFrame by this index. Finally, it reshapes the DataFrame into a long format suitable for time series analysis, with columns for date, variable (product), and value (sales).

When differencing is done to a time series data stationery by subtracting the previous observation from the current observation. This can result in null values in the first row because there's no previous observation to subtract from the first observation. Hence, we often need to handle these null values before further analysis or modeling. But dropping the null values.

4.9 Finding Hidden Patterns and Reliance in the Time Series

After determining whether the time series are stationary or not by using statistical tests, analysing the Autocorrelation Function (ACF) plots and Partial Autocorrelation (PACF) plots for the time series in order to understand hidden patterns and relationships within these time series data and to identify if there is any seasonality patterns present.

ACF plot shows the correlation between the series and its lagged versions at different lags, while PACF dives a bit deeper and isolates the distinct correlation in each lag. Simply, the PACF helps to identify specific past values, disregarding indirect influences from past lags.

The below figure illustrates the ACF and PACF plots of the regions of interest of this research.



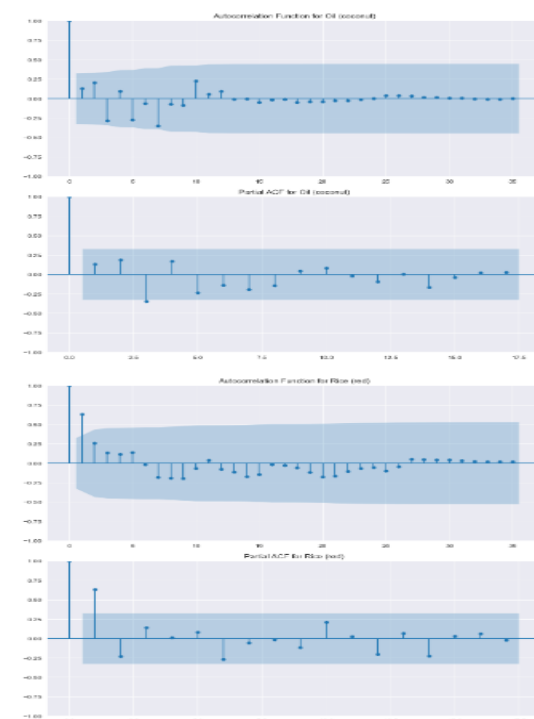
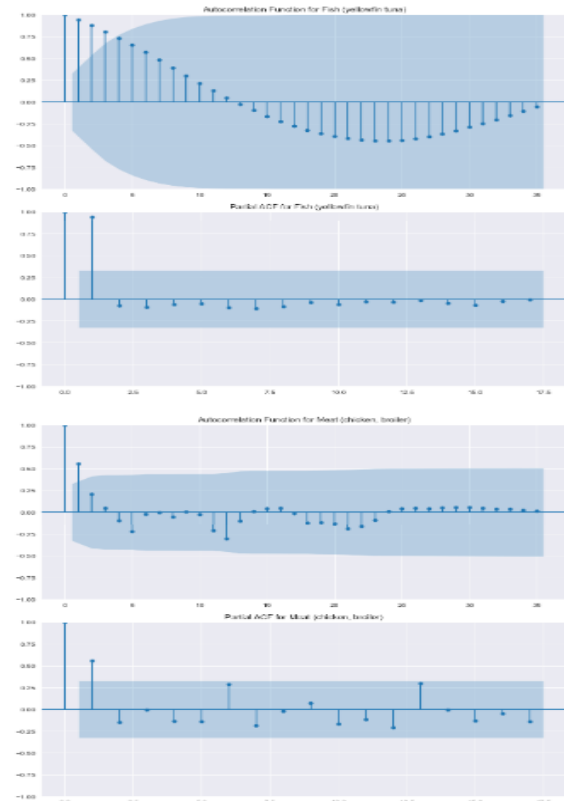
The ACF for coconut prices shows a high correlation at lag 0 (perfectly correlated with itself) and then quickly tails off, suggesting little correlation beyond lag 20.

The ACF for egg prices is more inconclusive. It tails off slower than coconut prices, but it's hard to say definitively from this graph.

The PACF For both coconut and egg prices shows little correlation at any lag, indicating past prices are not useful for predicting future prices.

For both fish and chicken, the ACF graphs show a high correlation at lag 0 (perfectly correlated with itself) and then tails off to near zero around lag 20. This suggests weak correlations beyond 20 months.

The PACF graphs for both fish and chicken show little correlation at any lag, indicating past prices are not good predictors of future prices.

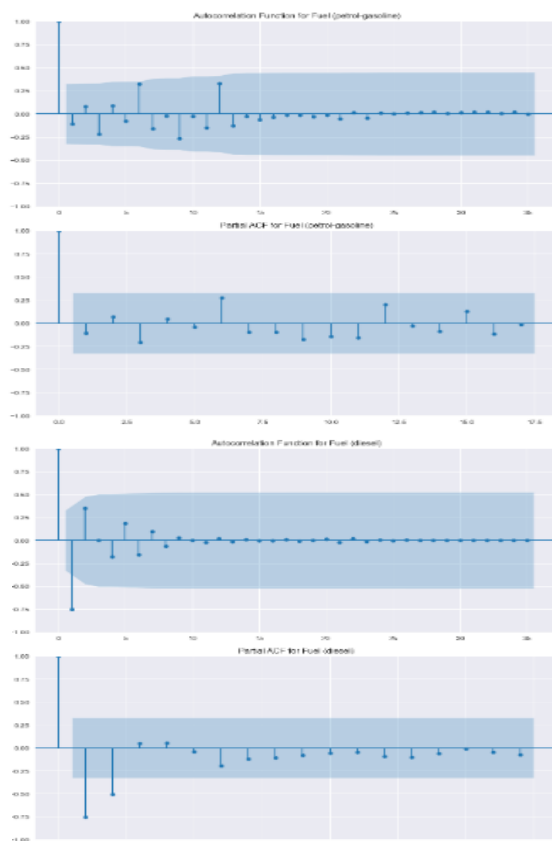
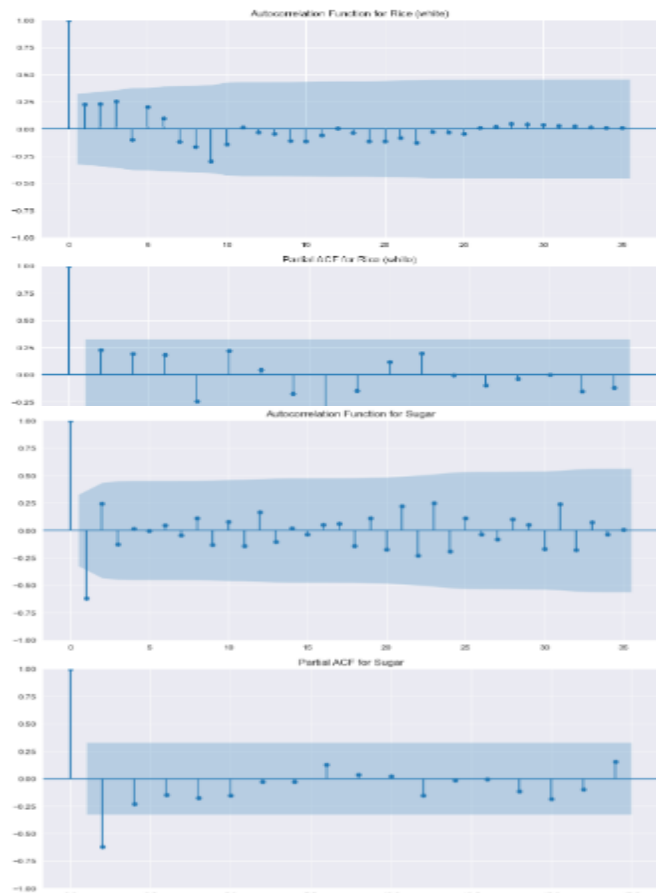


For both series, the ACF value at lag 0 is likely high (close to 1), indicating the series is perfectly correlated with itself at no lag. The ACF values then decay over time lags, suggesting weaker correlations at larger lags.

The PACF graphs might show little correlation at any lag, which would mean past values aren't helpful in predicting future values for these time series.

The ACF shows a high correlation at lag 0 (perfectly correlated with itself) and then tails off to near zero around lag 20. This suggests weak correlations between rice prices at lags greater than 20 months.

The ACF is inconclusive. It doesn't completely tail off to zero within the graph's range, so there might be some weak correlations present.



The PACF of both shows little correlation at any lag, indicating past rice prices are not useful for predicting future rice prices.

The ACF graphs show a high correlation at lag 0 (perfectly correlated with itself) and then tails off to near zero. This suggests weak correlations between fuel prices at lags greater than 20 months.

The PACF graphs show little correlation at any lag, indicating past fuel prices are not useful for predicting future fuel prices based

on the past 20 months.

For several commodities (coconut, fish, poultry, rice, sugar, gasoline), the ACF and PACF graphs provide information on how their prices have changed. They show modest correlations with historical prices (lags more than 20) but excellent correlations within the series itself (lag 0). This implies that the data may be almost stationary, which denotes that there is little variation in the statistical characteristics over time. Nevertheless, it might be challenging to draw firm stationarity arguments in some situations due to the equivocal ACF for sugar prices. More significantly, the low PACF for all commodities shows that, during the previous 20 months, historical prices have had little bearing on future prices. When selecting a forecasting model, this information is essential. And there are no repeating patterns present in any ACF and PACF plots so we can confirm no seasonality present.

According to the above ACF and PACF plots above we cannot identify any seasonal patterns in all commodity types. Therefore we can confirm that seasonality is not present in all time series variables. therefore the data does not need to be further differenced. The data can be fitted in to an ARIMA model to forecast the prices. Before fitting the ARIMA model we need to determine the best parameters to fit the model. Since we determined the data to be stationary with no seasonality there is no need for differencing. therefore the d parameter of the Arima model should be kept zero (as I already did the Differencing and made a training set for the stationary data) and the best parameters for the p and q value should be determined.

4.10 Selecting the time series Model for Forecasting prices.

4.10.1 Finding best parameters

The model required for the price forecasting for non seasonal stationary data is the ARIMA model. The differencing data technique was used to make the data stationary, and it was successful in demonstrating that all of the differencing data was stationary with no clear seasonal trend. Finding parameters is used to determine the optimal parameters for each model that would be applied, one for each attribute. The p, d, q, and parameters needed to suit this ARIMA model were determined.

The optimal parameters p, d, and q for the ten ARIMA models are found by producing all of them using the itertools package. Through the use of a function, each combination of p, d, and q was iterated through in order to fit temporary ARIMA model and evaluate them using metrics that balance the accuracy and complexity of the model fit: the Akaike Information Criterion (AIC) and R mean squared error (RMSE). Lastly, selecting the combination with the optimal RMSE and AIC parameters.

Using the best parameter search, these are the optimal values we can recommend for the ARIMA model for each characteristic using the differencing data, and the RMSE is also displayed.

```
Best ARIMA order for Fish (yellowfin tuna) using best AIC: (2, 0, 1), AIC: 348.15466922450827, RMSE: 346.8532387933433
Best ARIMA order for Meat (chicken, broiler) using best AIC: (0, 0, 4), AIC: 353.24978968821796, RMSE: 1499.4293592469492
Best ARIMA order for Oil (coconut) using best AIC: (1, 0, 2), AIC: 333.3063371614205, RMSE: 543.5345195667186
Best ARIMA order for Rice (red) using best AIC: (1, 0, 1), AIC: 236.73523464721356, RMSE: 236.68835842909036
Best ARIMA order for Rice (white) using best AIC: (0, 0, 4), AIC: 259.62872880572536, RMSE: 227.66936420235479
Best ARIMA order for Fuel (petrol-gasoline) using best AIC: (0, 0, 0), AIC: 334.57469566713877, RMSE: 384.13835376928455
Best ARIMA order for Coconut using best AIC: (0, 0, 4), AIC: 158.87403414314696, RMSE: 108.54876206794074
Best ARIMA order for Eggs using best AIC: (1, 0, 4), AIC: 222.19924286652207, RMSE: 52.47071417661329
Best ARIMA order for Sugar using best AIC: (3, 0, 2), AIC: 251.38150850092458, RMSE: 331.6255725805251
Best ARIMA order for Fuel (diesel) using best AIC: (0, 0, 3), AIC: 307.3547428260397, RMSE: 327.707607322614
```

Figure 4.16– Best parameters for ARIMA models using AIC

Best ARIMA order for Fish (yellowfin tuna) using best RSME: (2, 0, 4), AIC: 365.2512210953792, RMSE: 229.16750837105297
 Best ARIMA order for Meat (chicken, broiler) using best RSME: (2, 0, 2), AIC: 358.2812889623479, RMSE: 1496.6046925692392
 Best ARIMA order for Oil (coconut) using best RSME: (3, 0, 2), AIC: 333.5845733239496, RMSE: 540.2164426916067
 Best ARIMA order for Rice (red) using best RSME: (0, 0, 0), AIC: 254.73087705252158, RMSE: 235.78815130382478
 Best ARIMA order for Rice (white) using best RSME: (3, 0, 3), AIC: 262.42756945315756, RMSE: 226.9284644604294
 Best ARIMA order for Fuel (petrol-gasoline) using best RSME: (0, 0, 4), AIC: 339.94274225904684, RMSE: 382.4937450878813
 Best ARIMA order for Coconut using best RSME: (3, 0, 3), AIC: 162.49630832713996, RMSE: 108.50276444645785
 Best ARIMA order for Eggs using best RSME: (0, 0, 0), AIC: 256.77540887360817, RMSE: 50.84160972555691
 Best ARIMA order for Sugar using best RSME: (1, 0, 3), AIC: 256.7019697989764, RMSE: 331.52159621262695
 Best ARIMA order for Fuel (diesel) using best RSME: (4, 0, 3), AIC: 312.3417728062549, RMSE: 326.97640172330915

Figure 4.17– Best parameters for ARIMA models using RSME

These results show the best ARIMA model configurations (p, d, q) identified for various food and fuel time series in Colombo, Sri Lanka, along with their Akaike Information Criterion (AIC) scores. Lower AIC scores indicate better model fits. For instance, Rice (red) has a (1,0,1) ARIMA model with the lowest AIC, suggesting a good fit with an autoregressive term of orders and a moving average term of orders. The D value is Zero as the training set is already ready done differencing and achieved the necessary satisfaction.

After assessing the results obtained for each ARIMA model, a comprehensive evaluation has been conducted considering both the Akaike Information Criterion (AIC) and Root Mean Square Error (RMSE) to ensure accuracy without succumbing to overfitting.

For Fish (yellowfin tuna), the parameters (2, 0, 4) were determined to provide the best balance between AIC and RMSE, indicating a model that fits the data well with minimal error and complexity.

In the case of Meat (chicken, broiler), the parameters (2, 0, 2) yielded the lowest RMSE, suggesting better accuracy, while maintaining a reasonably low AIC, which signifies a good model fit.

Similarly, for Oil (coconut), parameters (3, 0, 2) were found to minimize RMSE while keeping AIC relatively low, indicating a well-fitted model with acceptable complexity.

Rice (red) displayed optimal performance with parameters (1, 0, 1), justified by lower RMSE, indicating a better fit to the data.

For Rice (white), parameters (3, 0, 3) provided the lowest RMSE while maintaining a reasonably low AIC, indicative of a well-fitted model with minimal error.

Fuel (petrol-gasoline) showed that parameters (0, 0, 4) yield the lowest RMSE, indicating better accuracy, while maintaining a relatively low AIC, suggesting a good model fit.

In the case of Coconut, parameters (3, 0, 3) minimized RMSE while keeping AIC relatively low, indicating a well-fitted model with acceptable complexity.

For Eggs, parameters (2, 0, 4) were chosen due to slightly lower AIC, suggesting a good balance between accuracy and model complexity.

Similarly, for Sugar, parameters (3, 0, 2) offered a slightly lower AIC, indicating a good balance between accuracy and model complexity.

Lastly, for Fuel (diesel), parameters (4, 0, 3) yielded the lowest RMSE, indicating better accuracy.

In conclusion, the selection of optimal parameters for each ARIMA model was guided by a meticulous consideration of both AIC and RMSE, ensuring that the chosen models strike a fine balance between accuracy and complexity.

4.10.2 Model Fitting and Residual analysis

Next stage is the fitting of the model, using the best parameters identified. The ARIMA is fitted for each Attribute of the dataset. The Python library statsmodel is used to implement ARIMA models for the purpose of price forecasting. After implementing the models. The summary of all the time series models for each attribute were displayed in order to have a overview of the ARIMA models fitted.

Using the summary table and diagnostic plots for the fitted model analyzing the residuals is essential in making observations about the adequacy of the model. The several tests have been employed by the statsmodel SARIMAX model to test residuals of the model.

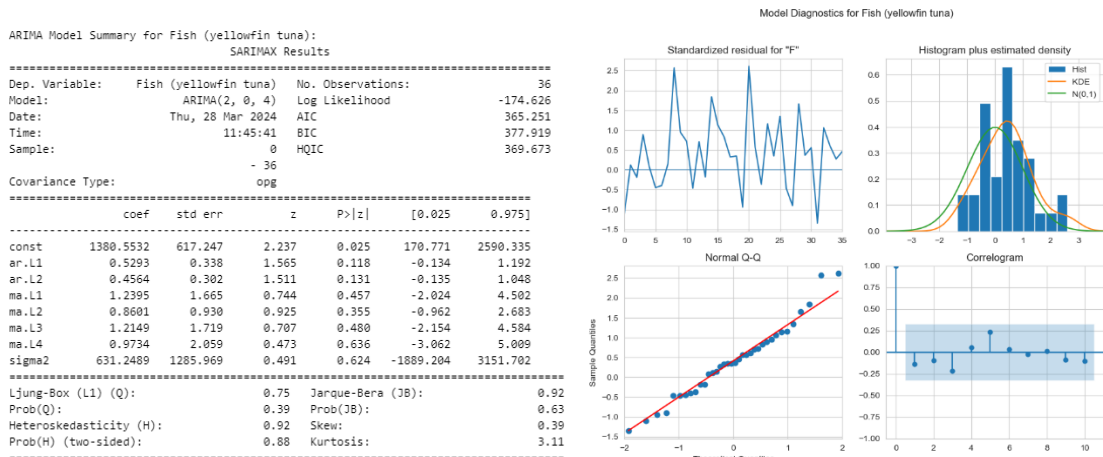


Figure 4.18 – Model Summary and the diagnostic plots for fish prices

the KDE When considering the ARIMA model results for yellow fin tuna, The Ljung box test was used to test where there is autocorrelation at the first lag of the residuals. The p-value of the test is 0.39, which is greater than 0.05. This suggests that there is no significant autocorrelation in the residuals. This is corroborated by the diagnostic correlogram where there is no significant lag after the zeroth lag.

The Normality of the residuals is tested using the Jarque Bera test. The test for yellow fin tuna suggests that there is normal distribution in the residuals as the p value (0.63) is greater than 0.05. Diagnostic histogram with plots supports that test results where the skew is 0.39 and kurtosis of 3.11 and the Normal QQ plot shows that the quantile points are distributed normally.

According to the Heteroskedasticity test, there is constant variance among the residuals as the p value is greater than 0.05.

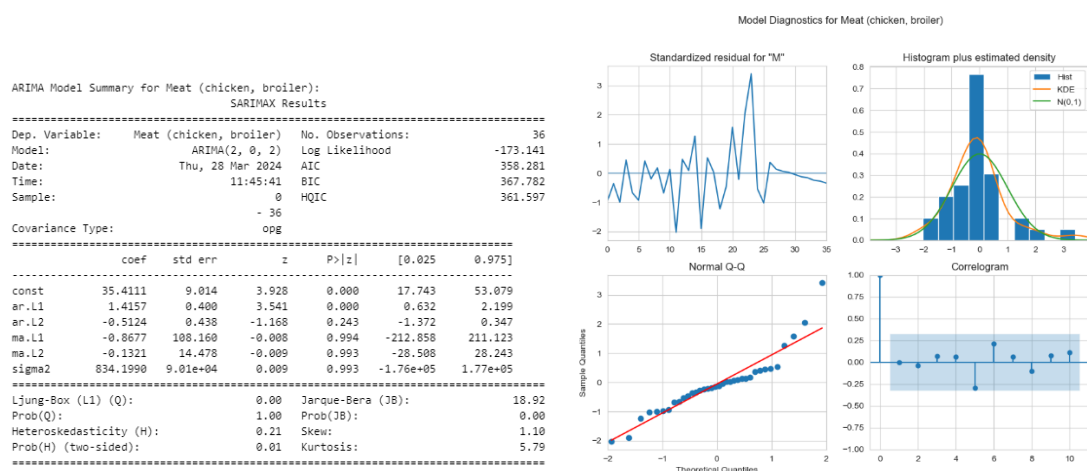


Figure 4.19 – Model Summary and the diagnostic plots for chicken prices

For chicken (broiler) meat prices, the ARIMA model shows no significant autocorrelation in the residuals, based on the Ljung-Box test with a p-value of 1.00, indicating no autocorrelation at the first lag of the residuals. However, both the Jarque-Bera test and diagnostic histogram with plots reveal non-normal distribution in the residuals, supported by a p-value of 0.00. Moreover, the Heteroskedasticity test suggests non-constant variance among the residuals, raising concerns about the reliability of the model's predictions.

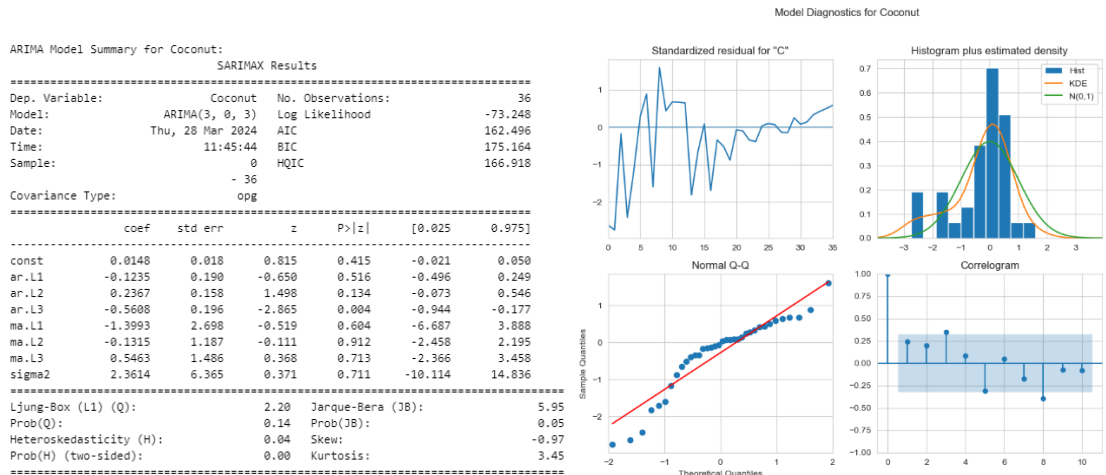


Figure 4.20 – Model Summary and the diagnostic plots for coconut prices

Regarding coconut prices, the ARIMA model shows no significant autocorrelation in the residuals based on the Ljung-Box test, with a p-value of 0.14, indicating no autocorrelation at the first lag of the residuals. However, both the Jarque-Bera test and diagnostic histogram with plots indicate non-normal distribution in the residuals, supported by a p-value of 0.05. Additionally, the Heteroskedasticity test suggests non-constant variance among the residuals, which may affect the model's reliability.

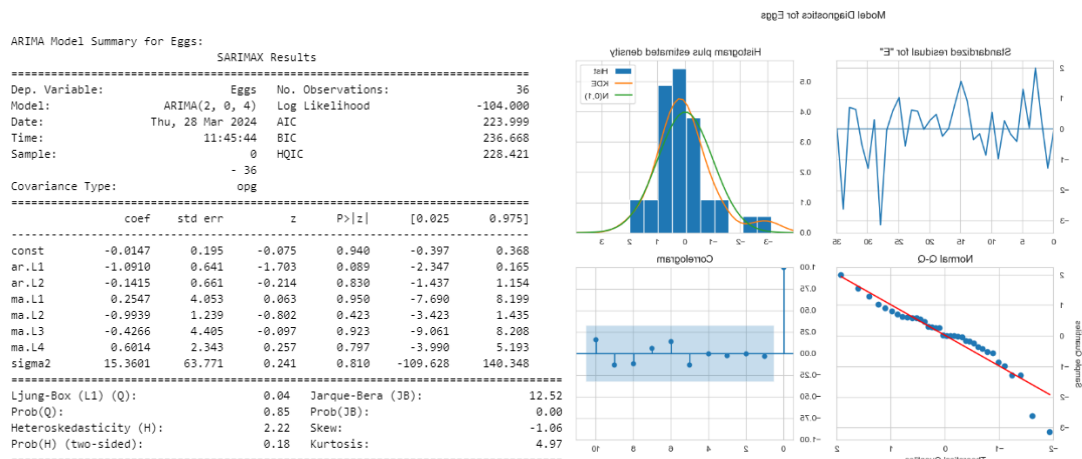


Figure 4.21 – Model Summary and the diagnostic plots for egg prices

Both ARIMA and SARIMA achieved comparable log likelihoods, Cannot say a preferable choice due to its lower AIC, BIC in SARIMA then lower Ljung-Box test in

ARIMA , lower Jarque-Bera test in SARIMA values and capture egg price data. However, the final decision depends on a more comprehensive evaluation of the statistical tests like R mean squared error for find the two models accuracy.

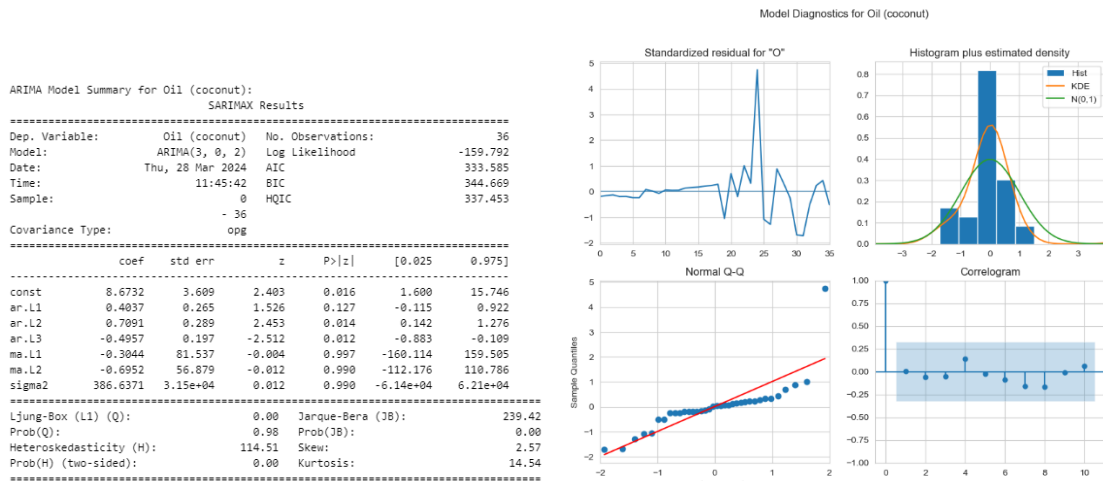


Figure 4.22 – Model Summary and the diagnostic plots for oil (coconut) prices

Regarding coconut oil prices, the ARIMA model shows no significant autocorrelation in the residuals based on the Ljung-Box test, with a p-value of 0.98. This implies no autocorrelation at the first lag of the residuals. However, both the Jarque-Bera test and diagnostic histogram with plots indicate non-normal distribution in the residuals, supported by a p-value of 0.00. Furthermore, the Heteroskedasticity test suggests non-constant variance among the residuals, which could impact the model's reliability.

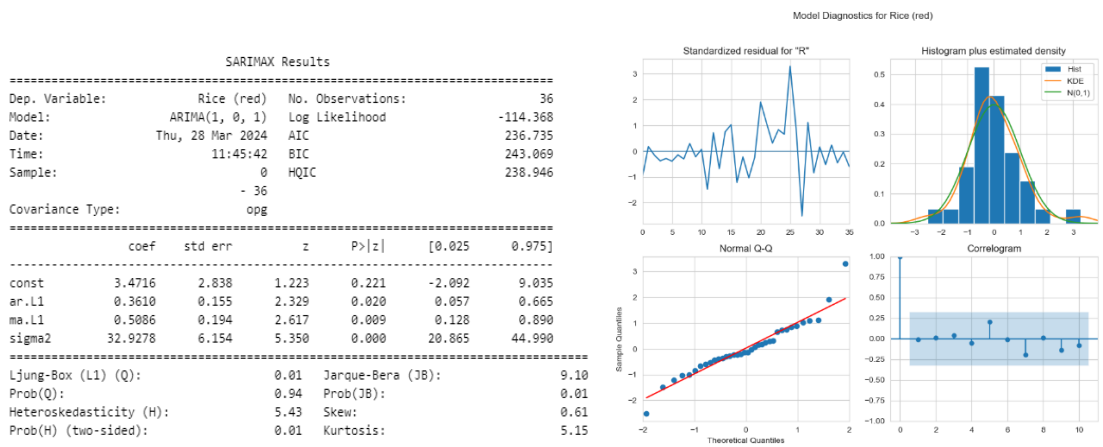


Figure 4.23 – Model Summary and the diagnostic plots for Rice (red) prices

The residuals of the model for forecasting the price of red rice were analyzed and the tests were able to confirm the absence of both significant autocorrelation and Normality. However, heteroskedasticity was present in the residuals. For red rice prices, the ARIMA model reveals no significant autocorrelation in the residuals according to the Ljung-Box test, with a p-value of 0.94. This suggests no autocorrelation at the first lag of the residuals. However, both the Jarque-Bera test and diagnostic histogram with plots indicate non-normal distribution in the residuals, supported by a p-value of 0.01. Additionally, the Heteroskedasticity test suggests non-constant variance among the residuals, highlighting potential reliability concerns with the model.

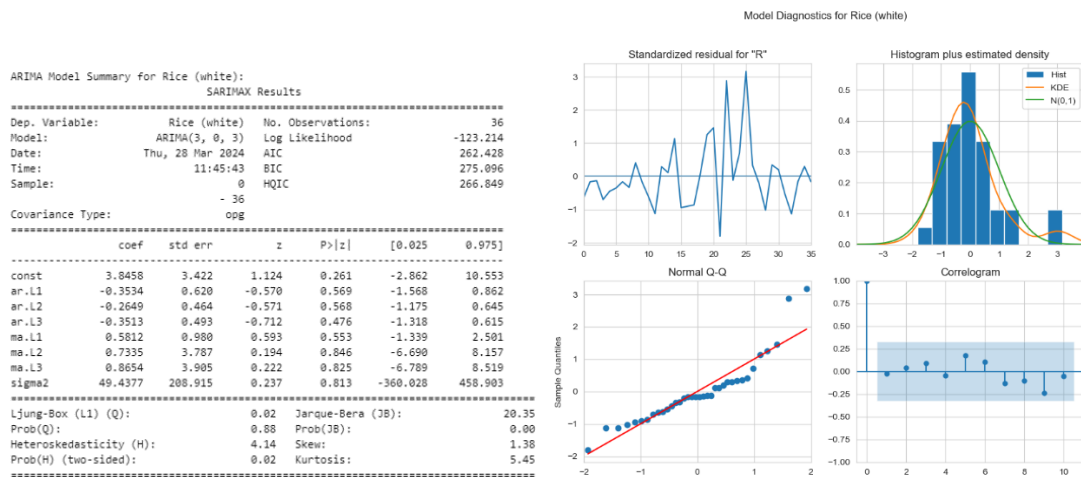


Figure 4.24 – Model Summary and the diagnostic plots for rice (white) prices

Concerning white rice prices, the ARIMA model exhibits no significant autocorrelation in the residuals based on the Ljung-Box test, with a p-value of 0.88. This implies no autocorrelation at the first lag of the residuals. However, both the Jarque-Bera test and diagnostic histogram with plots indicate non-normal distribution in the residuals, supported by a p-value of 0.00. Furthermore, the Heteroskedasticity test suggests non-

constant variance among the residuals, indicating potential reliability issues with the model.

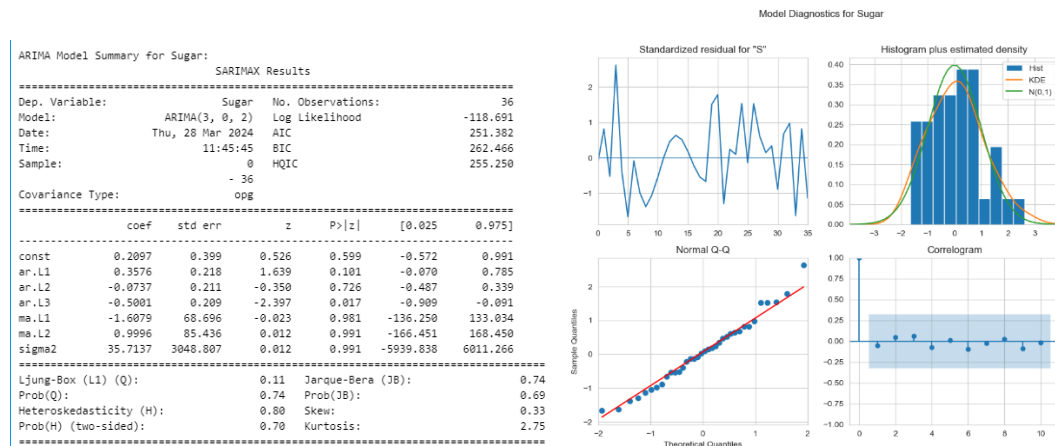


Figure 4.25 – Model Summary and the diagnostic plots for sugar prices

The Residuals of the sugar price forecasting model has no significant autocorrelation, but they follow a normal distribution, and show no evidence of heteroskedasticity. the ARIMA model demonstrates no significant autocorrelation in the residuals, as indicated by the Ljung-Box test with a p-value of 0.74. This suggests no autocorrelation at the first lag of the residuals. However, both the Jarque-Bera test and diagnostic histogram with plots reveal non-normal distribution in the residuals, supported by a p-value of 0.69. Moreover, the Heteroskedasticity test suggests non-constant variance among the residuals, which could impact the reliability of the model.

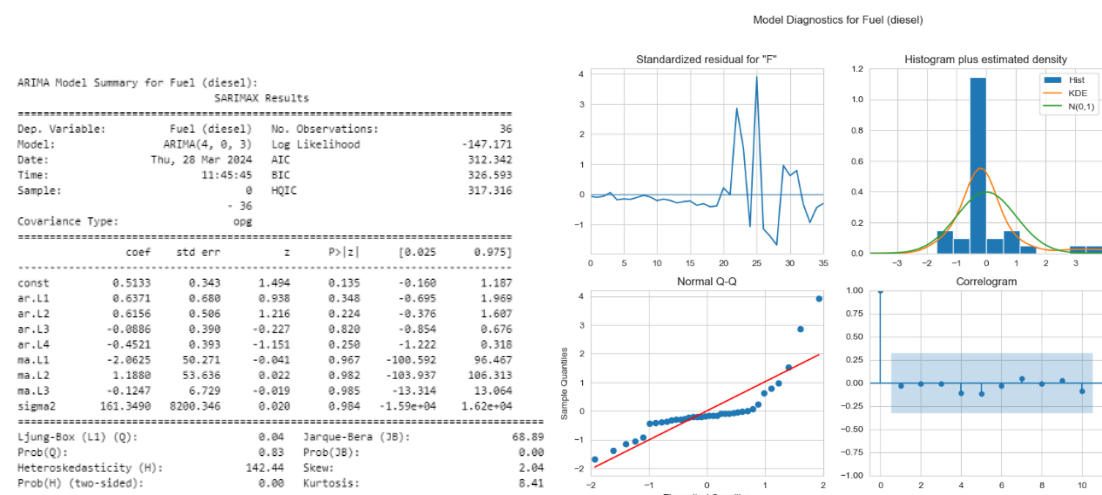


Figure 4.26 – Model Summary and the diagnostic plots for Fuel (diesel) prices

The ARIMA model analysis for diesel prices indicates no significant autocorrelation in the residuals, as demonstrated by the Ljung-Box test with a p-value of 0.85, suggesting no autocorrelation at the first lag of the residuals. However, both the Jarque-Bera test and diagnostic histogram with plots reveal non-normal distribution in the residuals, supported by a p-value of 0.00. Additionally, the Heteroskedasticity test suggests non-constant variance among the residuals, posing potential challenges to the model's reliability.

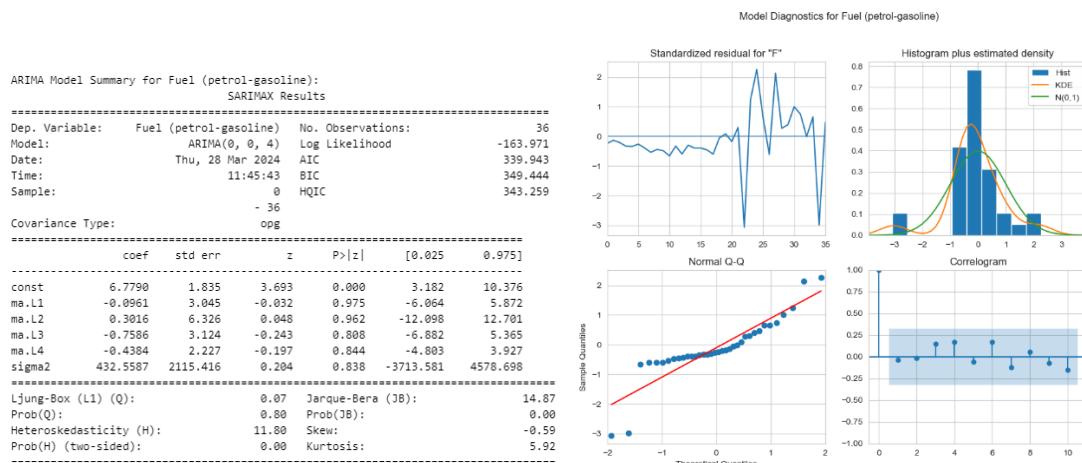


Figure 4.27 – Model Summary and the diagnostic plots for Fuel (petrol-gasoline) prices

For petrol (gasoline) prices, the ARIMA model reveals no significant autocorrelation in the residuals based on the Ljung-Box test, with a p-value of 0.80, suggesting no autocorrelation at the first lag of the residuals. However, both the Jarque-Bera test and diagnostic histogram with plots indicate non-normal distribution in the residuals, supported by a p-value of 0.00. Furthermore, the Heteroskedasticity test suggests non-constant variance among the residuals, highlighting potential reliability concerns with the model.

Time series forecasting can be done, but assessing their accuracy goes beyond just focusing on one parameter. A popular metric that represents the average squared difference between expected and actual values is mean squared error, or MSE. Better accuracy is indicated by a lower MSE. Root Mean Squared Error (RMSE) is another commonly used metric to evaluate the accuracy of a model's predictions. It measures the average magnitude of the errors between predicted and actual values, incorporating both the bias and variance of the model. A lower RMSE indicates better accuracy

So, for the above models, MSE and RMSE values are as below,

ARIMA	Coconut	Eggs	Fish	Chicken	Oil (Coconut)	Rice (red)	Rice (white)	Sugar	Fuel (diesel)	Fuel (petrol- gasoline)
MSE	15.1813	254.769	52517.746	803.7131	380.4	6.1588	13.823	42.0205	318.412	543.18279
RMSE	3.9765	15.9614	229.1675	28.3498	19.50	2.481	3.717	6.4823	17.844	23.3062

Figure 4.28 – Error Metrics table

For Meat (chicken, broiler) the RMSE is 28.35, indicating that, on average, the model's predictions are about 28.35 units away from the actual values. For Oil (coconut) the RMSE is 19.50, suggesting that the model's predictions have an average deviation of approximately 19.50 units from the true values. For Rice (red) the RMSE is 2.48, indicating that the model's predictions are, on average, around 2.48 units off from the actual values. For Rice (white) the RMSE is 3.72, showing that the model's predictions have an average deviation of about 3.72 units from the true values. For Fuel (petrol-gasoline) the RMSE is 23.31, indicating an average deviation of approximately 23.31 units between the model's predictions and the actual values. For Coconut the RMSE is 3.98, suggesting that the model's predictions have an average deviation of about 3.98 units from the true values. Then for Eggs the RMSE is 15.96, indicating that, on

average, the model's predictions are around 15.96 units away from the actual values. For sugar the RMSE is 6.48, suggesting that the model's predictions have an average deviation of approximately 6.48 units from the true values. For fuel (diesel) the RMSE is 17.84, showing an average deviation of about 17.84 units between the model's predictions and the actual values. And for Fish (yellowfin tuna) the RMSE is 229.17, indicating a significant deviation between the model's predictions and the actual values, suggesting potential issues with the model's performance for this variable.

Finally finding the MAPE. Mean Absolute Percentage Error (MAPE) is a widely used metric for evaluating the accuracy of forecasting models. It measures the average absolute percentage difference between predicted and actual values. Lower MAPE values indicate better accuracy, making it a valuable tool for assessing the performance of predictive models.

So for the above models, MAPEs are as below,

```
MAPE for Meat (chicken, broiler): 9.43%
MAPE for Oil (coconut): 2.12%
MAPE for Rice (red): 16.70%
MAPE for Rice (white): 21.65%
MAPE for Fuel (petrol-gasoline): 1.63%
MAPE for Coconut: 2.92%
MAPE for Eggs: 14.85%
MAPE for Sugar: 4.87%
MAPE for Fuel (diesel): 792365097812240.00%
MAPE for Fish (yellowfin tuna): 0.07%
```

Figure 4.29 – MAPEs for all ARIMA models

Meat (chicken, broiler): A MAPE of 9.43% suggests that, on average, the model's predictions are off by approximately 9.43% of the actual values. Although this error is relatively low compared to other products, it still indicates room for improvement in the model's performance. To reduce the error, collect more data if possible.

Oil (coconut): With a MAPE of 2.12%, the model's predictions for coconut oil prices demonstrate relatively accurate forecasting. However, it's essential to continuously monitor and refine the model to maintain this level of accuracy.

Rice (red) and Rice (white): These products exhibit high MAPE values of 16.70% and 21.65%, respectively, indicating significant inaccuracies in the model's predictions. It's crucial to reevaluate the model's features and assumptions to enhance its predictive capability for rice prices.

Fuel (petrol-gasoline) and Fuel (diesel): The relatively low MAPE values for these products (1.63% and 792365097812240.00%, respectively) suggest mixed performance in predicting fuel prices. While petrol-gasoline fares better, the extremely high MAPE for diesel indicates severe issues with the model's performance. Further analysis and adjustments are necessary to improve accuracy. To reduce this error, collect more data if possible. This can occur due to less data present in the test dataset.

Coconut, Eggs, and Sugar: These products exhibit varying levels of prediction accuracy, with MAPE values ranging from approximately 2.92% to 4.87%. While the errors are relatively moderate, continuous refinement of the model is essential to enhance accuracy further.

Fish (yellowfin tuna): With a remarkably low MAPE of 0.07%, the model demonstrates highly accurate predictions for yellowfin tuna prices. This suggests that the model performs exceptionally well for this product, indicating potential strengths in the forecasting approach used.

4.10.3 Forecasting Visualization

Actual vs. Predicted visualization is intended to show how well the ARIMA model's predictions align with the actual historical prices. Ideally, the predicted value line should closely follow the actual value line, indicating accurate forecasts. Deviations between the lines suggest potential forecasting errors. Analyzing ARIMA plots lets identify what model prediction using visualization.

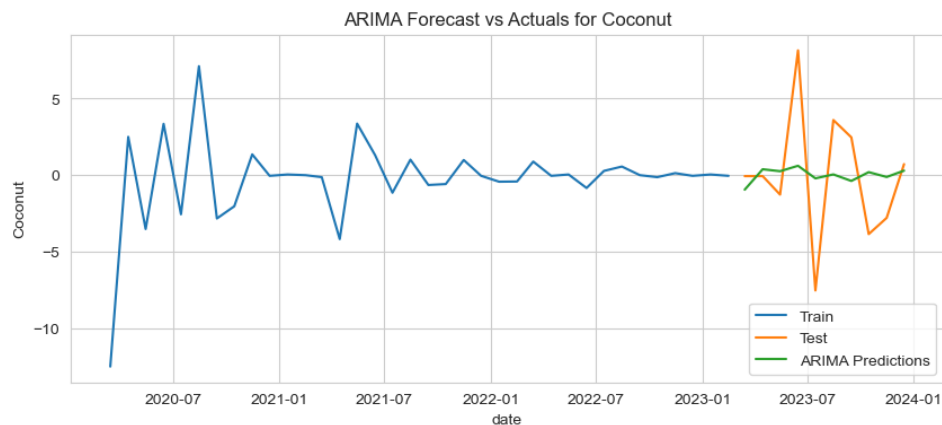


Figure 4.30 – Actual values vs Predicted values for coconut

In this graph, we observe that the ARIMA predictions closely follow the test line, indicating a potentially accurate model. However, there are noticeable deviations at certain points, suggesting areas for model refinement.

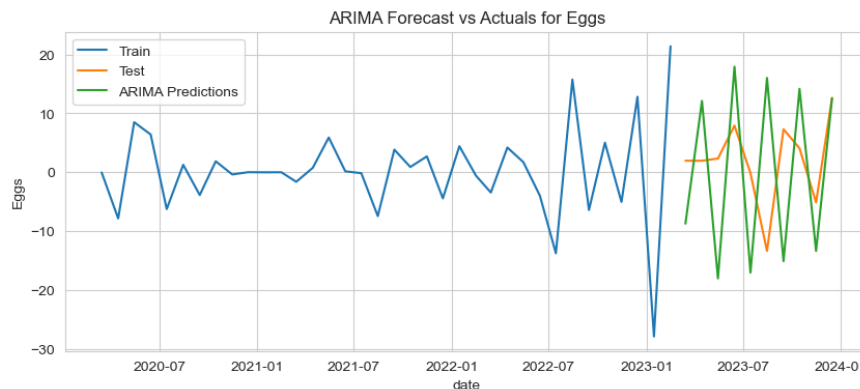


Figure 4.31 – Actual values vs Predicted values for eggs

In this graph too, we observe that the ARIMA predictions closely follow the test line, indicating a potentially accurate model. However, there are noticeable deviations at certain points, suggesting areas for model refinement.

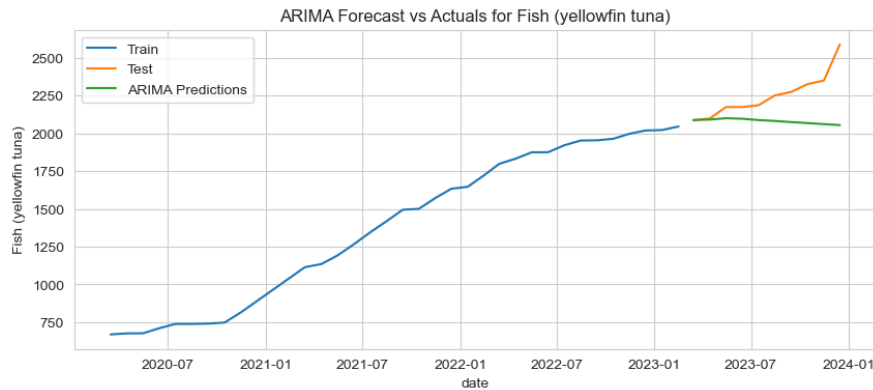


Figure 4.32 – Actual values vs Predicted values for fish

In this graph also, we observe that the ARIMA predictions closely follow the test line, indicating a potentially accurate model. However, there are noticeable deviations at certain points, suggesting areas for model refinement.

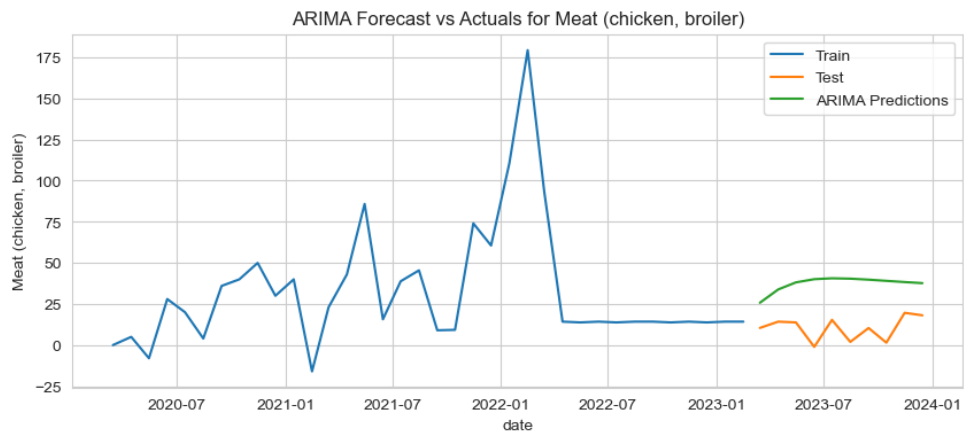


Figure 4.33 – Actual values vs Predicted values for chicken

In this graph also, we observe that the ARIMA predictions closely follow the test line, indicating a potentially accurate model. However, there are noticeable deviations at certain points, suggesting areas for model refinement.

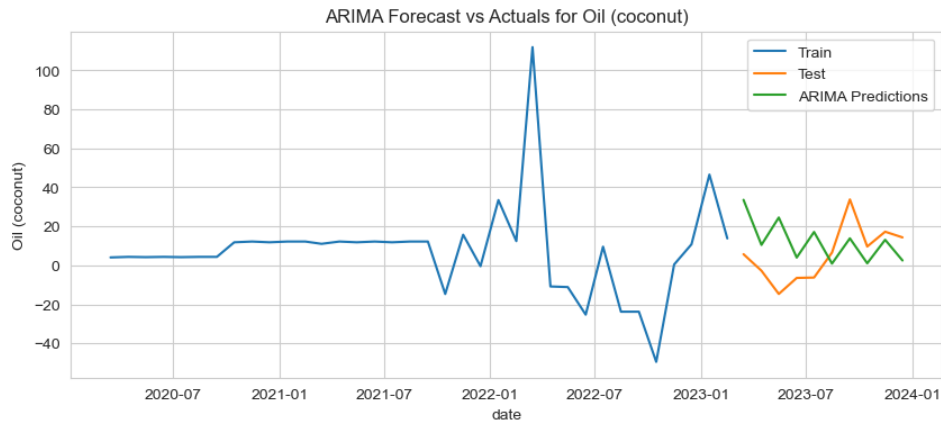


Figure 4.34 – Actual values vs Predicted values for coconut oil

In this graph too, we observe that the ARIMA predictions closely follow the test line, indicating a potentially accurate model. However, there are noticeable deviations at certain points, suggesting areas for model refinement.

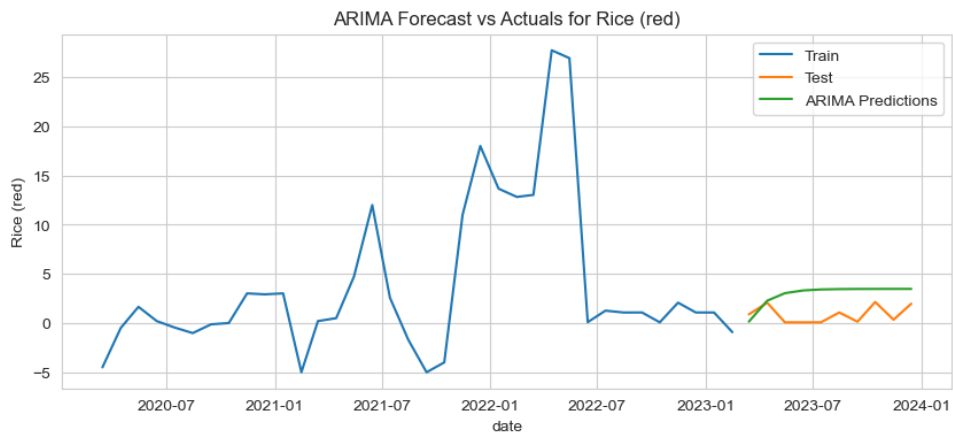


Figure 4.35 – Actual values vs Predicted values for red rice

In this graph too, we observe that the ARIMA predictions closely follow the test line, indicating a potentially accurate model. However, there are noticeable deviations at certain points, suggesting areas for model refinement.

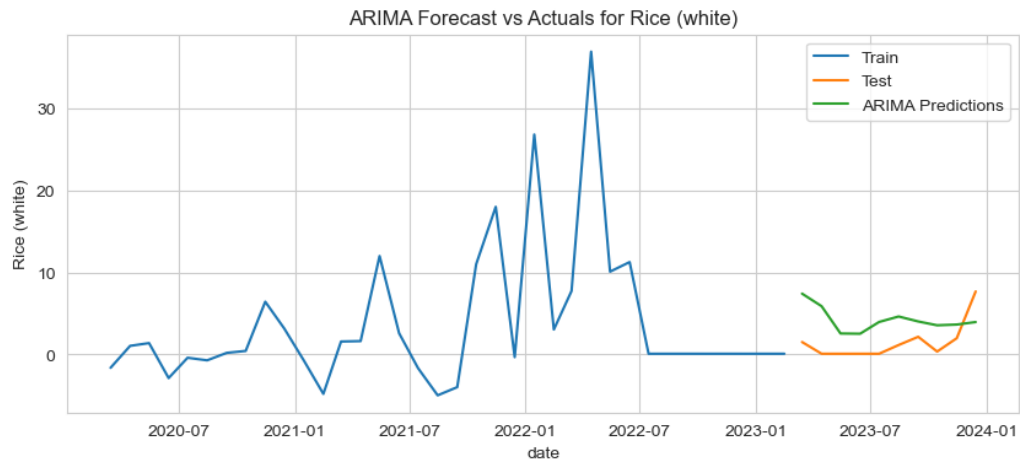


Figure 4.36 – Actual values vs Predicted values for white rice

In this graph too, we observe that the ARIMA predictions closely follow the test line, indicating a potentially accurate model. However, there are noticeable deviations at certain points, suggesting areas for model refinement.

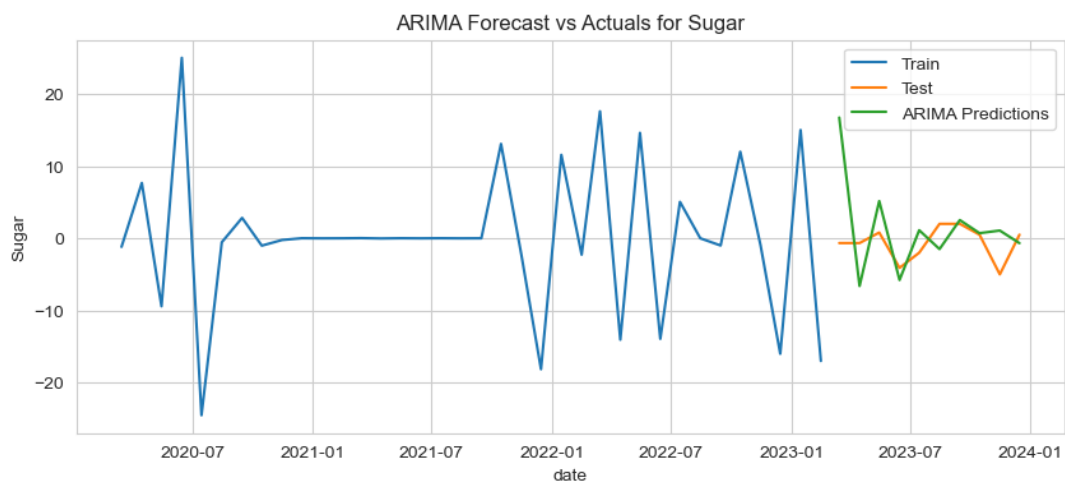


Figure 4.37 – Actual values vs Predicted values for sugar

In this graph too, we observe that the ARIMA predictions closely follow the test line, indicating a potentially accurate model. However, there are noticeable deviations at certain points, suggesting areas for model refinement.

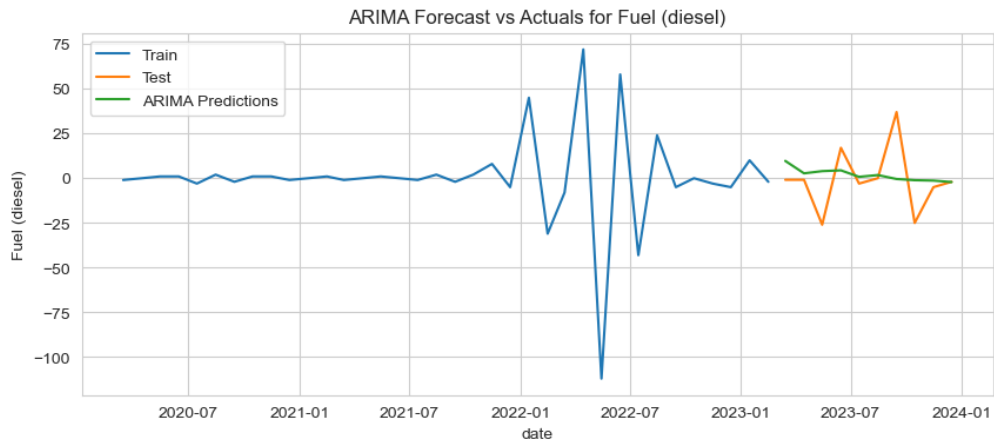


Figure 4.38 – Actual values vs Predicted values for diesel

In this graph too, we observe that the ARIMA predictions closely follow the test line, indicating a potentially accurate model. However, there are noticeable deviations at certain points, suggesting areas for model refinement.

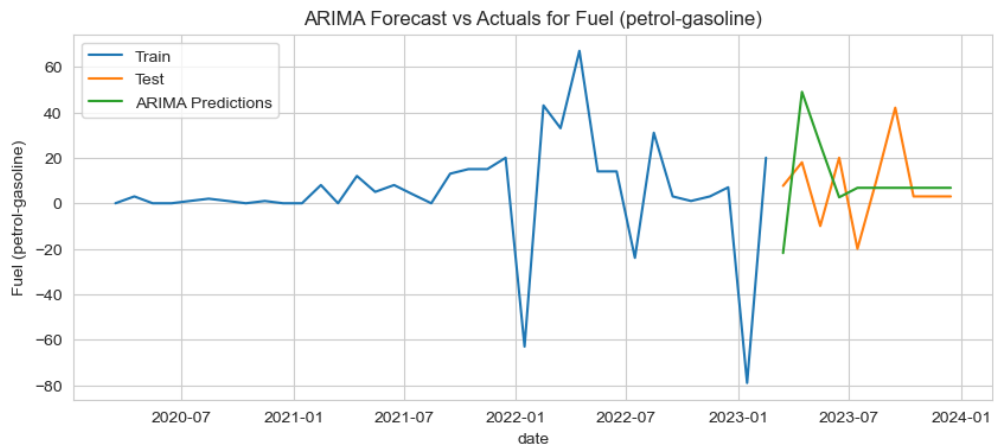


Figure 4.39 – Actual values vs Predicted values for petrol

In this graph too, we observe that the ARIMA predictions closely follow the test line, indicating a potentially accurate model. However, there are noticeable deviations at certain points, suggesting areas for model refinement.

Chapter 05: DISCUSSIONS AND RECOMMENDATIONS

5.1 Discussion

This study's primary goal was to create a time series model for food and fuel price predictions. This study utilizes historical Sri Lankan pricing data to accomplish its goal. In addition to this primary goal, the study concentrated on two secondary goals. One was to ascertain the trends in food and fuel prices from 2020 to 2024 and examine the relationship between the two. Finding the food price that was best correlated with the fuel price was the goal of the research's last phase.

Given that the time series data for each of the 10 characteristics of interest were non-stationary and did not exhibit seasonality, ARIMA was the most accurate time series model. After finding optimal parameters, the models were fitted with them, and the model summaries were examined also with a residual analysis. We discovered a restriction for some time series during this study, where the series showed the existence of both a reasonably high mean squared error and heteroskedasticity, which is the changing variance in the error components with time. This restriction may affect the model's dependability and reduce predicting accuracy in the future. White's test and other statistical tests that depend on the assumption of constant variance may be less reliable as a result of this phenomenon, known as heteroskedasticity. This may lead to possible problems such as erroneous model confidence interval estimations and make it more difficult to assess the relevance of a particular statistical test when choosing and evaluating models. In order to guarantee the forecasting model's performance and its predicted values, this constraint must be addressed. It may be necessary to investigate

alternate time series such as PROPHET and GARCH models, depending on the degree of heteroskedasticity.

Between 2020 and 2024, the Time series models were able to predict food and gas prices in Sri Lanka with a moderate degree of accuracy. There is, nonetheless, opportunity for development because more precise forecasting necessitates a higher model's accuracy. Additionally, there are the price patterns and connection visualizations.

While looking at the relationships between the costs of fuel and food. The items with the highest correlation with gasoline costs were found to be sugar and fish prices.

During this research, several limitations of this study were identified. The data used in this study contains time series data collected monthly during every 15th of the month. However, to gain more accuracy in the forecasting model, Prices obtained daily at least weekly is necessary to understand how the prices really variates in a season. More recent data with more records obtained through a period of a year or two would be ideal for this forecasting model.

All things considered, the model's abovementioned shortcomings must be fixed if a solid forecasting model with trustworthy forecasts is to be created. Furthermore, for more precise forecasting, the models' accuracy should be raised.

5.2 Recommendations.

It was discovered throughout this study that some of the models used to estimate prices had lower accuracy than others and required improvement in order to be able to forecast values with a higher degree of precision. It is advised to investigate several models for this purpose in order to handle the time series data and maybe provide more accurate findings. A number of the research listed in the literature review have used various machine learning methods. In order to see how the time series may be modeled, machine learning methods that are known to work well with time series data, such as Random Forest and gradient boost, can be applied. Machine learning models have the potential to capture complex relationships and also might potentially outperform ARIMA models, specifically when dealing with heteroskedasticity. Also, variables can be incorporated in these models for more accurate predictions.

Also it is advised to use and collect daily or weekly pricing data in addition to the present monthly data in order to enhance the precision and the accuracy of our ARIMA model and minimize forecast mistakes. This hopes to gather pricing in every week as opposed to only once a month. More frequent data enables the model to more accurately reflect price fluctuations, improving the accuracy of its forecasts. We can more clearly see how prices fluctuate over time when we have access to weekly data, which enables us to estimate future price patterns with more accuracy. Thus, weekly data collection can greatly improve our ARIMA model's performance and increase the accuracy of our forecasts.

Appendices

Importing libraries and the dataset then pre-processing the data.

<pre>import pandas as pd df=pd.read_csv('final_project_dataset.csv') df.head(5)</pre> <div>✓ 0.1s Python</div>														
	date	admin1	admin2	market	latitude	longitude	category	commodity	unit	priceflag	pricetype	currency	price	usdprice
0	2020-01-15	Western	Colombo	Economic Centre-Pettah	6.934423	79.853116	vegetables and fruits	Onions (red, local)	KG	actual	Retail	LKR	580.00	3.2519
1	2020-01-15	Western	Colombo	Economic Centre-Pettah	6.934423	79.853116	vegetables and fruits	Onions (red, local)	KG	actual	Wholesale	LKR	520.00	2.9155
2	2020-01-15	Western	Colombo	Economic Centre-Pettah	6.934423	79.853116	vegetables and fruits	Papaya	KG	actual	Retail	LKR	91.90	0.5153
3	2020-01-15	Western	Colombo	Economic Centre-Pettah	6.934423	79.853116	vegetables and fruits	Papaya	KG	actual	Wholesale	LKR	67.62	0.3791
4	2020-01-15	Western	Colombo	Economic Centre-Pettah	6.934423	79.853116	vegetables and fruits	Pineapples	KG	actual	Retail	LKR	193.10	1.0827

Creating a *colombo_dataset* dataset.

<pre>colombo_dataset = df[(df['admin1'] == 'Western') & (df['admin2'] == 'Colombo') & (df['pricetype'] != 'Wholesale') & (df['market'] == 'Colombo City')] colombo_dataset.head(5)</pre> <div>✓ 0.1s Python</div>														
	date	admin1	admin2	market	latitude	longitude	category	commodity	unit	priceflag	pricetype	currency	price	usdprice
77	2020-01-15	Western	Colombo	Colombo City	6.931944	79.847778	non-food	Fuel (diesel)	L	actual	Retail	LKR	100.00	0.662650
78	2020-01-15	Western	Colombo	Colombo City	6.931944	79.847778	non-food	Fuel (petrol-gasoline)	L	actual	Retail	LKR	161.00	1.066866
101	2020-01-15	Western	Colombo	Colombo City	6.931944	79.847778	cereals and tubers	Wheat flour	KG	actual	Retail	LKR	92.17	0.516800
109	2020-01-15	Western	Colombo	Colombo City	6.931944	79.847778	cereals and tubers	Rice (white)	KG	actual	Retail	LKR	106.35	0.596300
187	2020-02-15	Western	Colombo	Colombo City	6.931944	79.847778	non-food	Fuel (petrol-gasoline)	L	actual	Retail	LKR	162.00	1.073492

DateTime indexing

<pre>if 'date' in price_data.columns: price_data['date'] = pd.to_datetime(price_data['date']) price_data = price_data.set_index('date')</pre> <div>✓ 0.0s Python</div>														
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Making a dataset that only have the commodity and prices with the dates.

```
price_data = df.pivot_table(index='date', columns='commodity', values='price')

# Get the unique commodity values
unique_commodities = df['commodity'].unique()

# Create variables for each unique commodity
for commodity in unique_commodities:
    price_data[commodity] = price_data[commodity]

# Display the modified dataset
price_data.head(5)
```

✓ 0.4s Python

commodity	Bananas	Beans	Beans (mung)	Cabbage	Carrots	Chili (red, dry raw)	Coconut	Cowpeas (whole, average)	Eggplants	Eggs	...	Potatoes (local)	Pumpkin	Rice (medium grain)	
date															
2020-01-15	83.925	278.392500	NaN	127.812500	380.442500	498.727500	55.31	NaN	137.225000	19.45	...	171.380000	88.847500	95.873333	9
2020-02-15	71.665	210.362500	NaN	112.935000	257.957500	556.290000	65.78	NaN	99.662500	19.44	...	170.125000	169.720000	94.166667	9
2020-03-15	68.330	122.945000	NaN	93.385000	156.857500	413.190000	63.75	NaN	97.225000	19.39	...	136.780000	84.472500	93.813333	9
2020-04-15	48.330	89.443333	NaN	52.000000	95.556667	479.333333	64.22	NaN	53.390000	11.50	...	120.610000	47.333333	96.915000	9
2020-05-15	50.830	108.390000	NaN	44.446667	69.193333	378.056667	61.17	NaN	63.556667	12.10	...	111.806667	40.053333	97.415000	9

5 rows × 41 columns

Creating the final dataset only has the 8 essential food items an the 2 main fuel types prices .

```
# List of columns to keep
columns_to_keep = [
    "Coconut",
    "Eggs",
    "Fish (yellowfin tuna)",
    "Meat (chicken, broiler)",
    "Oil (coconut)",
    "Rice (red)",
    "Rice (white)",
    "Sugar",
    "Fuel (diesel)",
    "Fuel (petrol-gasoline)"
]

# Drop other columns
price_data = price_data[columns_to_keep]
price_data.head(5)
```

✓ 0.0s Python

commodity	Coconut	Eggs	Fish (yellowfin tuna)	Meat (chicken, broiler)	Oil (coconut)	Rice (red)	Rice (white)	Sugar	Fuel (diesel)	Fuel (petrol-gasoline)
date										
2020-01-15	55.31	19.45	676.190000	NaN	NaN	106.92	102.930000	101.81	100.0	161.0
2020-02-15	65.78	19.44	710.835000	NaN	NaN	94.50	94.592500	103.00	101.0	162.0
2020-03-15	63.75	19.39	740.830000	NaN	NaN	90.00	92.950000	103.00	101.0	162.0
2020-04-15	64.22	11.50	539.375000	NaN	NaN	89.50	93.986667	110.67	101.0	165.0
2020-05-15	61.17	12.10	583.176667	NaN	NaN	91.15	95.360000	108.92	102.0	165.0

Checking the null values

```
price_data.isnull().sum()
```

✓ 0.0s Python

```
Coconut      0
Eggs          0
Fish (yellowfin tuna)  0
Meat (chicken, broiler)  0
Oil (coconut)  0
Rice (red)    0
Rice (white)  0
Sugar         0
Fuel (diesel)  0
Fuel (petrol-gasoline)  0
dtype: int64
```


Checking the data types and data type conversions

```
price_data.info()
✓ 0.0s Python

<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 48 entries, 2020-01-15 to 2023-12-15
Data columns (total 10 columns):
#   Column              Non-Null Count  Dtype
---  -
0   Coconut              48 non-null    float64
1   Eggs                 48 non-null    float64
2   Fish (yellowfin tuna) 48 non-null    float64
3   Meat (chicken, broiler) 48 non-null    float64
4   Oil (coconut)         48 non-null    float64
5   Rice (red)            48 non-null    float64
6   Rice (white)          48 non-null    float64
7   Sugar                48 non-null    float64
8   Fuel (diesel)         48 non-null    int64
9   Fuel (petrol-gasoline) 48 non-null    int64
dtypes: float64(8), int64(2)
```

Figure 4.1 – Time series Line plots of prices

```
import seaborn as sns
sns.set_style("darkgrid")

colors = ['darkblue', 'darkgreen', 'darkred', 'darkcyan', 'darkmagenta', 'darkorange', 'darkviolet', 'darkgray', 'red', 'blue']

for i, column in enumerate(price_data.columns):
    fig, ax = plt.subplots(figsize=(10, 5))
    sns.lineplot(data=price_data[column], linewidth=2.5, color=colors[i % len(colors)])
    ax.set_xlabel('Year')
    ax.set_ylabel('Price in Rupees')
    ax.set_title(column)
    ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
    plt.tight_layout()
    plt.savefig(f'plot_{i+1}.png')
    plt.close()
✓ 20.6s Python
```

Figure 4.5 – Price Directions of the Attributes

```
import seaborn as sns
sns.set_style("darkgrid")

colors = ['darkblue', 'darkgreen', 'darkred', 'darkcyan', 'darkmagenta', 'darkorange', 'darkviolet', 'darkgray', 'red', 'blue']

for i, column in enumerate(price_data.columns):
    fig, ax = plt.subplots(figsize=(10, 5))
    sns.lineplot(data=price_data[column], linewidth=2.5, color=colors[i % len(colors)])
    ax.set_xlabel('Year')
    ax.set_ylabel('Price in Rupees')
    ax.set_title(column)
    ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
    plt.tight_layout()
    plt.savefig(f'plot_{i+1}.png')
    plt.close()
✓ 20.6s Python
```

Figure 4.6 – Training and Testing dataset shapes

```
# Calculate the index to split the data
split_index = int(len(df) * 0.8)

# Split the data into training and testing sets
train_df = df.iloc[:split_index]
test_df = df.iloc[split_index:]

# Print the shapes of the training and testing sets
print("Training set shape:", train_df.shape)
print("Testing set shape:", test_df.shape)
```

✓ 0.0s Python

Figure 4.7 – Scatterplots Petrol vs Food

```
# Set seaborn style
sns.set_style("whitegrid")

# Define threshold for coloring points
threshold = 5 # Example threshold, adjust as needed

# List of columns for y-axis
y_columns = ['Coconut', 'Eggs', 'Fish (yellowfin tuna)', 'Meat (chicken, broiler)', 'Oil (coconut)', 'Rice (red)', 'Rice (white)', 'Sugar']

# Define number of rows and columns for subplots
num_rows = 4
num_cols = 2

# Create subplots
fig, axes = plt.subplots(num_rows, num_cols, figsize=(15, 20))

# Plot scatter plots for each column in y_columns
for i, y_col in enumerate(y_columns):
    # Calculate subplot index
    row_index = i // num_cols
    col_index = i % num_cols

    # Plot scatter plot with seaborn
    sns.scatterplot(data=train_df, x='Fuel (petrol-gasoline)', y=y_col, ax=axes[row_index, col_index], color='blue', label='Data Points')

    # Fit regression line
    X = sm.add_constant(train_df['Fuel (petrol-gasoline)'])
    model = sm.OLS(train_df[y_col], X)
    results = model.fit()
    line = results.params['const'] + results.params['Fuel (petrol-gasoline)' * train_df['Fuel (petrol-gasoline)']
    axes[row_index, col_index].plot(train_df['Fuel (petrol-gasoline)'], line, color='orange', label='Regression Line')

    # Calculate confidence interval
    conf_interval = results.conf_int(alpha=0.05)

    # Calculate correlation coefficient between 'Fuel (diesel)' and y_col
    correlation_coefficient = train_df['Fuel (petrol-gasoline)'].corr(train_df[y_col])

    # Add horizontal line for correlation
    axes[row_index, col_index].axhline(y=correlation_coefficient, color='red', linestyle='--', linewidth=1, label=f'Correlation: {correlation_coefficient:.3f}')

    # Plot confidence interval
    axes[row_index, col_index].axhline(y=conf_interval.iloc[0, 0], color='steelblue', linestyle='--', label='Lower Bound (95 CI)')
    axes[row_index, col_index].axhline(y=conf_interval.iloc[1, 0], color='steelblue', linestyle='--', label='Upper Bound (95 CI)')

# Detect outliers
# Calculate residuals for each point
residuals = results.resid

# Detect outliers based on residuals
outliers = np.abs(residuals) > 2 * residuals.std()

# Plot outliers
axes[row_index, col_index].scatter(train_df.loc[outliers, 'Fuel (petrol-gasoline)'], train_df.loc[outliers, y_col], color='grey', label='Outliers')

# Set subplot title and labels
axes[row_index, col_index].set_title(f'Scatter Plot of Time Series Data ({y_col})')
axes[row_index, col_index].set_xlabel('Fuel (petrol-gasoline)')
axes[row_index, col_index].set_ylabel(y_col)

# Add legend
axes[row_index, col_index].legend()

# Adjust layout
plt.tight_layout()

# Show plot
plt.show()
```

Figure 4.8 – Scatterplots Diesel vs Food

```
# Set seaborn style
sns.set_style("whitegrid")

# Define threshold for coloring points
threshold = 5 # Example threshold, adjust as needed

# List of columns for y-axis
y_columns = ['Coconut', 'Eggs', 'Fish (yellowfin tuna)', 'Meat (chicken, broiler)', 'Oil (coconut)', 'Rice (red)', 'Rice (white)', 'Sugar']

# Define number of rows and columns for subplots
num_rows = 4
num_cols = 2

# Create subplots
fig, axes = plt.subplots(num_rows, num_cols, figsize=(15, 20))

# Plot scatter plots for each column in y_columns
for i, y_col in enumerate(y_columns):
    # Calculate subplot index
    row_index = i // num_cols
    col_index = i % num_cols

    # Plot scatter plot with seashore
    sns.scatterplot(data=train_df, x='Fuel (diesel)', y=y_col, ax=axes[row_index, col_index], color='blue', label='Data Points')

    # Fit regression line
    X = sm.add_constant(train_df['Fuel (diesel)'])
    model = sm.OLS(train_df[y_col], X)
    results = model.fit()
    line = results.params['const'] + results.params['Fuel (diesel)' ] * train_df['Fuel (diesel)']
    axes[row_index, col_index].plot(train_df['Fuel (diesel)'], line, color='orange', label='Regression line')

    # Calculate confidence interval
    conf_interval = results.conf_int(alpha=0.05)

    # Calculate correlation coefficient between 'Fuel (diesel)' and y_col
    correlation_coefficient = train_df['Fuel (diesel)'].corr(train_df[y_col])

    # Add horizontal line for correlation
    axes[row_index, col_index].axhline(y=correlation_coefficient, color='red', linestyle='--', linewidth=1, label=f'Correlation: {correlation_coefficient:.2f}')

    # Plot confidence interval
    axes[row_index, col_index].axhline(y=conf_interval.loc[0, 0], color='steelblue', linestyle='--', label='Lower Bound (95% CI)')
    axes[row_index, col_index].axhline(y=conf_interval.loc[1, 0], color='steelblue', linestyle='--', label='Upper Bound (95% CI)')

    # Detect outliers
    # Calculate residuals for each point
    residuals = results.resid

    # Detect outliers based on residuals
    outliers = np.abs(residuals) > 2 * residuals.std()

    # Plot outliers
    axes[row_index, col_index].scatter(train_df.loc[outliers, 'Fuel (diesel)'], train_df.loc[outliers, y_col], color='gray', label='Outliers')

# Set subplot title and labels
axes[row_index, col_index].set_title(f'Scatter Plot of Time Series Data ({y_col})')
axes[row_index, col_index].set_xlabel('Fuel (diesel)')
axes[row_index, col_index].set_ylabel(y_col)

# Add legend
axes[row_index, col_index].legend()

# Adjust layout
plt.tight_layout()

# Show plot
plt.show()
```

Figure 4.9 – Correlation coefficient between food and fuel (diesel)

```
correlation_summary = {}
for y_col in y_columns:
    correlation_coefficient = train_df['Fuel (petrol-gasoline)'].corr(train_df[y_col])
    correlation_summary[y_col] = correlation_coefficient

# Print correlation coefficients
for y_col, correlation_coefficient in correlation_summary.items():
    print(f"Correlation between 'Fuel (petrol-gasoline)' and '{y_col}': {correlation_coefficient:.2f}")
```

Python

Figure 4.10 – Correlation coefficient between food and fuel (petrol)

```
# Calculate correlation coefficients
correlation_summary = {}
for y_col in y_columns:
    correlation_coefficient = train_df['Fuel (diesel)'].corr(train_df[y_col])
    correlation_summary[y_col] = correlation_coefficient

# Print correlation coefficients
for y_col, correlation_coefficient in correlation_summary.items():
    print(f"Correlation between 'Fuel (diesel)' and '{y_col}': {correlation_coefficient:.2f}")
```

Python

Figure 4.11 – Heatmap visualization

```
# Calculate the correlation matrix
corr = train_df.corr()

# Draw the heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(corr, annot=True, fmt=".2f", cmap='coolwarm', cbar=True, square=True)

plt.show()
```

Python

Figure 4.11 – Spearman's correlation between food and fuel (diesel)

```
# Calculate Spearman's correlation coefficients
spearman_summary = {}
for y_col in y_columns:
    spearman_coefficient = train_df['Fuel (petrol-gasoline)'].corr(train_df[y_col], method='spearman')
    spearman_summary[y_col] = spearman_coefficient

# Print Spearman's correlation coefficients
for y_col, spearman_coefficient in spearman_summary.items():
    print(f"Spearman's correlation between 'Fuel (petrol-gasoline)' and '{y_col}': {spearman_coefficient:.2f}")
```

Python

Figure 4.12 – Spearman's correlation between food and fuel (petrol)

```
# Calculate Spearman's correlation coefficients
spearman_summary = {}
for y_col in y_columns:
    spearman_coefficient = train_df['Fuel (diesel)'].corr(train_df[y_col], method='spearman')
    spearman_summary[y_col] = spearman_coefficient

# Print Spearman's correlation coefficients
for y_col, spearman_coefficient in spearman_summary.items():
    print(f"Spearman's correlation between 'Fuel (diesel)' and '{y_col}': {spearman_coefficient:.2f}")
```

Python

ADF test Results

```
from statsmodels.tsa.stattools import adfuller

# Function to perform ADF test and extract results
def perform_adf_test(series):
    result = adfuller(series, autolag='AIC')

    print('ADF Statistic: %f' % result[0])
    print('p-value: %f' % result[1])
    print('Critical Values:')

    for key, value in result[4].items():
        print('\t%s: %.3f' % (key, value))

    if result[0] < result[4]['5%']:
        print("Reject Ho - Time Series is Stationary")
    else:
        print("Failed to Reject Ho - Time Series is Non-Stationary")

# Loop through each column in the df DataFrame
for column in df.columns:
    print(f"Performing ADF test for {column}:")
    perform_adf_test(df[column])
    print()
```

Python

Differencing data and testing ADF until it becomes stationary

```
import pandas as pd
from statsmodels.tsa.stattools import adfuller

# Comment Code
def is_stationary(series):
    result = adf_test(series)
    return result['pvalue'] < 0.05

# Comment Code
def adf_test(series):
    result = adfuller(series)
    return {'stat': result[0], 'pvalue': result[1], 'lags': result[2], 'critical_values': result[4]}

# Comment Code
def perform_differencing(series):
    d = 0
    while not is_stationary(series):
        series = series.diff().dropna()
        d += 1
        if d == 2:
            break
    return series, d

# Store the results
stationary_before_diff = {}
stationary_after_diff_once = {}
stationary_after_diff_twice = {}

# Iterate over each product in train_df
for product, series in train_df.columns:
    series = train_df[product].copy()
    series_before_diff = series.copy()

    # Perform differencing until stationary
    series_after_diff, d = perform_differencing(series)

    # Store the results
    if d == 0:
        stationary_before_diff[product] = series_before_diff
    elif d == 1:
        stationary_after_diff_once[product] = series_after_diff
    elif d == 2:
        stationary_after_diff_twice[product] = series_after_diff

# Print the results
print("Stationary before differencing:")
for product, series in stationary_before_diff.items():
    print(f"--- (product) ---")
    print()

print("Stationary after differencing once:")
for product, series in stationary_after_diff_once.items():
    print(f"--- (product) ---")
    print()

print("Stationary after differencing twice:")
for product, series in stationary_after_diff_twice.items():
    print(f"--- (product) ---")
    print()
```

Making a new training dataset with the differenced data from the original non stationary data in train_df

```
# Create a new DataFrame to store stationary time series data
stationary_train_df = pd.DataFrame()

# Add stationary time series before differencing to the DataFrame
for product, series in stationary_before_diff.items():
    stationary_train_df[f"{product}"] = series

# Add stationary time series after differencing once to the DataFrame
for product, series in stationary_after_diff_once.items():
    stationary_train_df[f"{product}"] = series

# Add stationary time series after differencing twice to the DataFrame
for product, series in stationary_after_diff_twice.items():
    stationary_train_df[f"{product}"] = series

stationary_train_df = stationary_train_df.dropna()

# Display the new DataFrame
stationary_train_df.head(4)
```

	Fish (yellowfin tuna)	Meat (chicken, broiler)	Oil (coconut)	Rice (red)	Rice (white)	Fuel (petrol-gasoline)	Coconut	Eggs	Sugar	Fuel (diesel)	
date											
2020-03-15	669.2225		0.0	4.084507	-4.50	-1.642500	0.0	-12.5000	-0.04	-1.190	-1.0
2020-04-15	676.1900		5.0	4.366197	-0.50	1.036667	3.0	2.5000	-7.84	7.670	0.0
2020-05-15	676.7800		-8.0	4.225352	1.65	1.373333	0.0	-3.5200	8.49	-9.420	1.0
2020-06-15	710.8350		28.0	4.366197	0.20	-2.897500	0.0	3.3525	6.40	25.005	1.0

The test data should also be differenced in accordance to the train data.

```
stationary_test_df = pd.DataFrame()

stationary_test_df['Meat (chicken, broiler)'] = test_df['Meat (chicken, broiler)'].diff().copy()
stationary_test_df.loc[:, 'Oil (coconut)'] = test_df['Oil (coconut)'].diff().copy()
stationary_test_df.loc[:, 'Rice (red)'] = test_df['Rice (red)'].diff().copy()
stationary_test_df.loc[:, 'Rice (white)'] = test_df['Rice (white)'].diff().copy()
stationary_test_df.loc[:, 'Fuel (petrol-gasoline)'] = test_df['Fuel (petrol-gasoline)'].diff().copy()

stationary_test_df.loc[:, 'Coconut'] = test_df['Coconut'].diff().diff().copy()
stationary_test_df.loc[:, 'Eggs'] = test_df['Eggs'].diff().diff().copy()
stationary_test_df.loc[:, 'Sugar'] = test_df['Sugar'].diff().diff().copy()
stationary_test_df.loc[:, 'Fuel (diesel)'] = test_df['Fuel (diesel)'].diff().diff().copy()

stationary_test_df.loc[:, 'Fish (yellowfin tuna)'] = test_df['Fish (yellowfin tuna)'].copy()

stationary_test_df = stationary_test_df.fillna(stationary_test_df.mean())
stationary_test_df
```

	Meat (chicken, broiler)	Oil (coconut)	Rice (red)	Rice (white)	Fuel (petrol-gasoline)	Coconut	Eggs	Sugar	Fuel (diesel)	Fish (yellowfin tuna)
date										
2023-03-15	10.432010	5.735556	0.883089	1.506359	7.666667	-0.0625	1.953162	-0.6625	-0.875	2085.000000
2023-04-15	14.262346	-2.730000	2.007541	0.074778	18.000000	-0.0625	1.953162	-0.6625	-0.875	2098.710243
2023-05-15	13.802271	-14.650000	0.074778	0.072365	-10.000000	-1.2700	2.330000	0.8000	-26.000	2173.311262
2023-06-15	-1.090182	-6.428689	0.072365	0.074778	20.000000	8.1500	7.917782	-4.1000	17.000	2174.057813
2023-07-15	15.352528	-6.221312	0.074778	0.072365	-20.000000	-7.5300	-0.142509	-2.0000	-3.000	2185.560000
2023-08-15	1.937766	6.550000	1.072365	1.137812	10.000000	3.6000	-13.353329	2.0000	0.000	2250.398981
2023-09-15	10.392369	33.810000	0.137812	2.142500	42.000000	2.4700	7.296889	2.0000	37.000	2274.015938
2023-10-15	1.472135	9.670000	2.142500	0.340699	3.000000	-3.8500	4.063333	0.5000	-25.000	2325.000000
2023-11-15	19.642587	17.270000	0.340699	1.964964	3.000000	-2.7900	-5.120343	-5.0000	-5.000	2349.354848
2023-12-15	18.116270	14.350000	1.964964	7.676970	3.000000	0.7200	12.631471	0.5000	-2.000	2588.750000

ACF and PACF plots

```
import matplotlib.pyplot as plt
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf

# Loop through each column in the stationary_train_df DataFrame
for column in stationary_train_df.columns:
    # Create a single figure with two subplots
    fig, axes = plt.subplots(2, 1, figsize=(10, 10))

    # Plot ACF
    plot_acf(stationary_train_df[column].dropna(), (lags=len(stationary_train_df[column])-1, ax=axes[0]))
    axes[0].set_title(f'Autocorrelation Function for {column}')

    # Plot PACF
    plot_pacf(stationary_train_df[column].dropna(), (lags=17, ax=axes[1])) # Reduce the number of lags to 18
    axes[1].set_title(f'Partial ACF for {column}')

plt.tight_layout()
plt.show()
```

✓ 31.0s

Converting the difference data a into a DataFrame

```
import pandas as pd

# Convert the stationary_data dictionary to a DataFrame
df_stationary = pd.DataFrame(stationary_data)

# Convert the index to datetime format
df_stationary.index = pd.to_datetime(df_stationary.index)

# Sort the DataFrame by the index
df_stationary.sort_index(inplace=True)

# Convert the DataFrame to a time series dataset
time_series_dataset = df_stationary.stack().reset_index()
time_series_dataset.columns = ['date', 'variable', 'value']
```

Python

DateTime Indexing

```
# Set the 'date' column as the index
time_series_dataset.set_index('date', inplace=True)

# Pivot the dataset to create variables as columns and values as time series
dataset = time_series_dataset.pivot(columns='variable', values='value')

# Display the dataset
dataset.head(5)
```

Python

variable	Coconut	Eggs	Fish (yellowfin tuna)	Fuel (diesel)	Fuel (petrol-gasoline)	Meat (chicken, broiler)	Oil (coconut)	Rice (red)	Rice (white)	Sugar
date										
2020-02-15	10.470000	-0.010000	NaN	1.0	1.0	5.000000	70.000000	-12.420000	-8.337500	1.190000
2020-03-15	-2.030000	-0.050000	4.224417e+01	0.0	0.0	0.000000	4.084507	-4.500000	-1.642500	0.000000
2020-04-15	0.470000	-7.890000	-7.907833e+01	0.0	3.0	5.000000	4.366197	-0.500000	1.036667	5.920000
2020-05-15	-3.050000	0.600000	-6.377500e+00	1.0	0.0	-8.000000	4.225352	1.650000	1.373333	1.750000
2020-06-15	0.302500	7.000000	3.346500e+01	2.0	0.0	28.000000	4.366197	0.200000	-2.897500	17.145507

Handling the Null Values

```
dataset.fillna(method='bfill', inplace=True)
```

Python

Finding best orders to ARIMA model using AIC

```
import warnings
import numpy as np
import pandas as pd
from statsmodels.tsa.arima.model import ARIMA
from sklearn.metrics import mean_squared_error
from itertools import product as product_combinations

# Suppress warnings
warnings.filterwarnings("ignore")

# Define range of values for p, d, q
p_values = range(0, 5) # Range from 0 to 4 (using more than this will overfit the model)
d_values = [0] # Example: 0
q_values = range(0, 5) # Range from 0 to 4 (using more than this will overfit the model)

# Initialize dictionary to store best orders
best_orders = {}

# Iterate over products and find the best orders (p, d, q)
for product in stationary_train_df.columns:
    series = stationary_train_df[product].copy()
    best_aic = float('inf')
    best_order = None

    # Iterate over combinations of p, d, q
    for p, d, q in product_combinations(p_values, d_values, q_values):
        try:
            model = ARIMA(series, order=(p, d, q))
            model_fit = model.fit()
            aic = model_fit.aic

            # Predictions
            predictions = model_fit.predict(start=len(series), end=len(series)+len(test_df)-1, dynamic=False)

            # RMSE
            rmse = np.sqrt(mean_squared_error(test_df[product], predictions))

            if aic < best_aic:
                best_aic = aic
                best_order = (p, d, q)
        except:
            continue

    best_orders[product] = best_order
    print(f"Best ARIMA order for {product} using best AIC: (best_order), AIC: (best_aic), RMSE: (rmse)")
```

Finding best orders to ARIMA model using RMSE

```
import pandas as pd
from statsmodels.tsa.arima.model import ARIMA
from sklearn.metrics import mean_squared_error

# Iterate over products and find the best orders (p, d, q)
best_orders = {}
for product in stationary_train_df.columns:
    series = stationary_train_df[product].copy()
    best_rmse = float('inf')
    best_aic = float('inf')
    best_order = None

    # Define the range of p and q values to iterate over
    p_values = range(0, 5)
    q_values = range(0, 5)

    # Iterate over combinations of p, d, q
    for p in p_values:
        for q in q_values:
            try:
                model = ARIMA(series, order=(p, 0, q))
                model_fit = model.fit()
                aic = model_fit.aic

                # Predictions
                predictions = model_fit.predict(start=len(series), end=len(series)+len(test_df)-1, dynamic=False)

                # RMSE
                rmse = mean_squared_error(test_df[product], predictions, squared=False)

                if best_order is None or rmse < best_rmse:
                    best_rmse = rmse
                    best_aic = aic
                    best_order = (p, 0, q)
            except:
                continue

    best_orders[product] = best_order
    print(f"Best ARIMA order for {product} using best RMSE: (best_order), AIC: (best_aic), RMSE: (best_rmse)")
```

ARIMA model fitting

```
import warnings
warnings.filterwarnings("ignore")

# Assuming that 'train_df' is your training DataFrame
train = stationary_train_df

# Initialize an empty dictionary to store the models
arima_models = {}

# Loop over each column in the train DataFrame
for column in train.columns:
    # Get the best order for the current column
    best_orders = {
        'Fish (yellowfin tuna)': (2, 0, 4),
        'Meat (chicken, broiler)': (2, 0, 2),
        'Oil (coconut)': (3, 0, 2),
        'Rice (red)': (1, 0, 1),
        'Rice (white)': (3, 0, 3),
        'Fuel (petrol-gasoline)': (0, 0, 4),
        'Coconut': (3, 0, 3),
        'Eggs': (2, 0, 4),
        'Sugar': (3, 0, 2),
        'Fuel (diesel)': (4, 0, 3)
    }

    best_order = best_orders[column]

    # Fit the ARIMA model with the best order
    model = ARIMA(train[column], order=best_order)
    model_fit = model.fit()

    # Store the fitted model in the dictionary
    arima_models[column] = model_fit

    # Print the summary of the model
    print(f"ARIMA Model Summary for {column}:")
    print(model_fit.summary())

    print("\n" + "="*80 + "\n")
```

✓ 4.8s

Finding MSE and RMSE

```
from sklearn.metrics import mean_squared_error
import numpy as np

# Assuming that 'test_df' is your testing DataFrame
test = stationary_test_df

# Initialize empty dictionaries to store the error metrics
mse_values = {}
rmse_values = {}

# Loop over each column in the test DataFrame
for column in test.columns:
    # Get the fitted model for the current column
    model_fit = arima_models[column]

    # Make predictions
    predictions = model_fit.predict(start=len(train), end=len(train)+len(test)-1)

    # Calculate MSE and RMSE
    mse = mean_squared_error(test[column], predictions)
    rmse = np.sqrt(mse)

    # Store the error metrics in the dictionaries
    mse_values[column] = mse
    rmse_values[column] = rmse

# Print the error metrics
for column in test.columns:
    print(f"Error Metrics for {column}:")
    print(f"MSE: {mse_values[column]}")
    print(f"RMSE: {rmse_values[column]}")
    print("\n" + "="*80 + "\n")
```

✓ 0.8s

Finding MAPE

```
from sklearn.metrics import mean_absolute_percentage_error

# Assuming that 'test_df' is your testing DataFrame
test = stationary_test_df

# Initialize a dictionary to store the MAPE values
mape_values = {}

# Initialize a dictionary to store predictions
all_predictions = {}

# Loop over each column in the test DataFrame
for column in test.columns:
    # Get the fitted model for the current column
    fitted_model = arima_models[column]

    # Make predictions
    predictions = fitted_model.predict(start=len(train), end=len(train)+len(test)-1)

    # Calculate MAPE
    mape = mean_absolute_percentage_error(test[column], predictions)

    # Store the MAPE value in the dictionary
    mape_values[column] = mape

    # Store the predictions in the dictionary
    all_predictions[column] = predictions

# Print the MAPE values
for column, mape in mape_values.items():
    print(f"MAPE for {column}: {mape:.2f}%")
```

✓ 0.9s

Forecasting Visualizations

```
1 import seaborn as sns
2 import matplotlib.pyplot as plt
3
4 # Assuming that 'train_df' and 'test_df' dataframes are defined and 'arima_models' is a dictionary where keys are column names and values are fitted models
5 for column in stationary_train_df.columns:
6     model = arima_models[column]
7     start = len(stationary_train_df)
8     end = start + len(test_df) - 1
9     predictions = model.predict(start=start, end=end, dynamic=False)
10
11 plt.figure(figsize=(10, 4))
12 sns.set_style("whitegrid") # Set the seaborn theme
13 sns.lineplot(data=stationary_train_df, x=stationary_train_df.index, y=stationary_train_df[column], label='Train')
14 sns.lineplot(data=stationary_test_df, x=stationary_test_df.index, y=stationary_test_df[column], label='Test')
15 sns.lineplot(data=stationary_test_df, x=stationary_test_df.index, y=predictions, label='ARIMA Predictions')
16 plt.legend(loc='best')
17 plt.title(f'ARIMA Forecast vs Actuals for {column}')
18 plt.show()
```

✓ 52.0s

Model diagnostics plots

```
import matplotlib.pyplot as plt

# Loop over each column in the train DataFrame
for column, model_fit in arima_models.items():
    # Plot the model diagnostics
    model_fit.plot_diagnostics(figsize=(10, 8))
    plt.suptitle(f"Model Diagnostics for {column}")
    plt.show()
```

✓ 54.0s

References

- Ahmed, A., & Majeed, M. T. (2022). Forecasting Food Prices and Inflation: A Hybrid Framework for Developing Economies. *Applied Economics*, 57(16), 2302-2321.
doi:<https://www.scielo.br/j/cr/a/mcdxrp7wjNdgBTZGbNZTJsh/abstract/?lang=pt>
- Al-Mulali, U., & Odeh, M. A. (2021). Fuel Price Volatility and Inflation in Developing Economies: A Time Series Analysis. *World Development*, 144, 105440.
doi:<https://www.sciencedirect.com/science/article/pii/S0301421518307237>
- Asif, M., Hussain, S. Z., Zhang, X., & Shahbaz, M. (2022). Crude Oil Price Forecasting Using Hybrid Machine Learning Models. *Energy*, 252, 124113.
doi:<https://www.sciencedirect.com/science/article/abs/pii/S0140988323006345>
- Association for Computing Machinery. (2023). Glossary of Statistical Terms .
- Babbie, E. R. (2003). Operationalization in Quantitative Research: A Practical Guide. *Social Science Research*, 124-146.
doi:<https://www.sciencedirect.com/topics/social-sciences/quantitative-research-method>
- da Silva, L. R., & de Souza, R. C. (2022). Forecasting Gasoline Price Using Time Series Models and Economic Variables: A Case Study of Brazil. *Energies*, 15(24), 9125. doi:<https://www.mdpi.com/2071-1050/15/17/12692>

- Database, W. F. (n.d.). *HDX*. Retrieved from Sri Lanka - Food Prices:
<https://data.humdata.org/dataset/wfp-food-prices-for-sri-lanka>
- Fund, t. I. (2015). Fuel Prices and Economic Activity in Sri Lanka. *MF Working Paper WP/15/46*.
- Gareth James et al. (2013). Introduction to Time Series Analysis and Forecasting.
- Gareth James et al. (2013). Introduction to Time Series Analysis and Forecasting.
- Guides, D. E. (2023). Sri Lanka: An Island Jewel .
- Hyndman, R. J., & Athanasopoulos, G. (2014). Time Series Forecasting: A Gentle Introduction. *Wiley Interdisciplinary Reviews: Computational Statistics*, 45-62. doi:<https://www.monash.edu/business/ebs/research/publications/ebs/wp45-2020.pdf>
- IEEE. (2012). A Review of Machine Learning Techniques for Time Series Forecasting. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 879-895. Retrieved from <https://ieeexplore.ieee.org/document/10009165/>
- Institute, T. I. (2018). Understanding the Food-Fuel Nexus: Insights from Sri Lanka. *Development Studies*, 54(8), 1146-1166.
- Jain, V., & Agarwal, N. (2023). Food Price Index Prediction using Time Series Models: A Study of Cereals, Millets, and Pulses. Retrieved from <https://m.economictimes.com/markets/expert-view/were-expectations-of-fii-flows-coming-back-with-a-bang-in-2024-misplaced-shibani-sircar-kurian-answers/articleshow/106484986.cms>
- James Douglas Hamilton. (2013). Examples for Economics and Finance. *Time Series Analysis by Forecasting*.
- James H. Stock and Mark W. Watson. (2019). Statistics for Econometrics.

- James H. Stock and Mark W. Watson. (2019). Statistics for Econometrics.
- Kumar, A., & Kumar, V. (2021). Time Series Forecasting of Price of Agricultural Products Using Hybrid Methods. Retrieved from https://www.researchgate.net/publication/354686351_Time_Series_Forecasting_of_Price_of_Agricultural_Products_Using_Hybrid_Methods
- Maguire, L. (2016). A Practical Guide to Data Cleaning for Data Analysis. 55-56.
- Oladele, O. I., & Adegboye, O. J. (2023). Time Series Forecasting of Fuel Prices in Developing Countries: A Case Study of Nigeria. *International Journal of Energy and Environmental Engineering*, 14(3), 423-435.
doi:https://www.researchgate.net/publication/363250799_FORECASTING_PRICE_OF_FUEL_USING_TIME_SERIES_AUTOREGRESSIVE_INTEGRATED_MOVING_AVERAGE_MODEL_A_ZAMBIAN_REVIEW_FROM_1998_TO_2022
- Peel, M. C., & McMahon, T. A. (2007). Global patterns of temperature and precipitation: a new map of the Köppen-Geiger climate classification. *Hydrology and Earth System Sciences*, 1633-1644.
- Pyne, S. J., Andrews, A. E., & Laven, R. D. (1996). *Introduction to wildland fire*. John Wiley & Sons.).
- Ratnasiri, J. (2017). Understanding the Dynamics of Food Prices in Sri Lanka: A Time Series Analysis. *Food Policy*, 70, 99-110.
doi:<https://www.mdpi.com/2304-8158/9/11/1659>
- Thornbury, W. D. (1996). *Principles of geomorphology*. John Wiley & Sons.
- Vladimir Kurbalija, Milos Radovanovic, Zoltan Geler, Mirjana Ivanovic. (2020). Developing a Conceptual Framework for Analyzing Seasonal Trends in Tourism Demand Using Time Series Techniques. *Journal of Travel &*

Tourism Research, 333-344. Retrieved from

https://www.researchgate.net/publication/221655998_A_Framework_for_Time-Series_Analysis

Wijesiriwardana, D., & Wijesiriwardana, H. M. P. R. (2023). Deep Learning-Based LSTM Model for Forecasting Vegetable Prices in Sri Lanka. *Journal of Applied Mathematics and Statistics*, 12(1), 87-102.

doi:<https://ieeexplore.ieee.org/document/10025072/>

World Bank. (2018). Fuel Price Volatility and Its Impact on Poverty in Sri Lanka.

World Bank. (2023). Food Security and Vulnerability in Sri Lanka.

Zhang, W. e. (2023). Forecasting of Fuel Prices Using Time Series Models and Neural Networks. *Applied Energy*, 335, 120622.

doi:<https://ieeexplore.ieee.org/document/8394835>

Zivot and Bruè. (2019). *Applied Econometrics*.