HO CHI MINH CITY UNIVERSITY OF ECONOMICS AND FINANCE
FACULTY OF INFORMATION TECHNOLOGY

FINAL COURSE REPORT

TIMES SERIES ANALYSIS

Title: Using seasonal model and missing data estimation to forecast Ceramics Store Revenue in Time series analysis

Lecturer supervisor
MSC. NGO THUAN DU

Performed by

1. Phung Nguyen Ngoc Minh
2. Nguyen Hoang Van Khanh
3. Pham Nguyen Uyen Nhi
4. Dinh Thi Thu Trang
5. Nguyen Thuy Vy

Ho Chi Minh City, Vietnam
30th October 2025

HO CHI MINH CITY UNIVERSITY OF ECONOMICS AND FINANCE
FACULTY OF INFORMATION TECHNOLOGY

FINAL COURSE REPORT

TIMES SERIES ANALYSIS

Title: Using seasonal model and missing data estimation to forecast Ceramic Store Revenue in Time Series analysis

Lecturer supervisor
MSC. NGO THUAN DU

Performed by

1. Phung Nguyen Ngoc Minh 225210806
2. Nguyen Hoang Van Khanh 225210732
3. Pham Nguyen Uyen Nhi 225210784
4. Dinh Thi Thu Trang 225210681
5. Nguyen Thuy Vy 225210789

Ho Chi Minh City, Vietnam
30th October 2025

Acknowledgments

We would like to express our deepest and most sincere appreciation to our Lecturer Supervisor, **M.Sc. Ngo Thuan Du**, for his rigorous guidance, profound expertise in the field of Time Series Analysis, and persistent encouragement throughout the completion of this Final Course Report.

This research, titled "Using seasonal model and missing data estimation to forecast Ceramic Store Revenue in Time Series analysis," greatly benefited from his critical insights. His direction was instrumental in formulating the methodology for handling missing data and applying the complex seasonal models, which proved crucial for achieving the forecast results. His dedication to academic excellence served as a constant motivation.

Furthermore, we wish to acknowledge the faculty and staff of the Faculty of Information Technology (IT) of UEF for providing the supportive academic environment and necessary resources that facilitated this study. We are also grateful for the collaborative environment created by our colleagues and classmates, whose stimulating discussions and camaraderie made the intensive study sessions productive and enjoyable.

Finally, we owe a special debt of gratitude to our family and friends. Their patience, unwavering belief, and emotional support sustained us throughout the demanding periods of research and writing. This achievement is shared with them.

Table of Contents

List of Figures

Abstract

Accurate revenue forecasting plays a crucial role in the strategic management of manufacturing enterprises, especially in industries with pronounced seasonality, such as ceramics. This study was conducted to address the revenue forecasting problem for the ceramics sector by comparing the performance of two models: the traditional linear SARIMA (Seasonal Autoregressive Integrated Moving Average) model and a hybrid model combining the strengths of SARIMA with the machine learning algorithm XGBoost. Based on historical data starting from May 2017 and processed using R, the research was carried out in three sequential steps: (1) estimating and imputing missing values to ensure the integrity of the time series, (2) building a SARIMA model to capture linear trend and seasonal components, and (3) training an XGBoost model to learn complex, nonlinear patterns hidden in the residuals of SARIMA. Experimental results show that SARIMA–XGBoost hybrid model significantly outperforms the standalone SARIMA model. This confirms that a flexible combination of classical statistical models and modern machine learning techniques can create a powerful forecasting tool, capable not only of capturing seasonality but also of accurately reflecting nonlinear fluctuations, thereby providing reliable revenue forecasts to support business planning.

Chapter 1
INTRODUCTION

1.1 Background and Problem Statement

In the context of the Fourth Industrial Revolution and the digital economy, leveraging historical data for accurate business forecasting has become a key factor in maintaining a competitive advantage. Particularly in the production and trading of consumer goods such as ceramics — a traditional industry whose revenue fluctuates significantly due to seasonality, economic cycles, and trade events — reliable revenue forecasting is essential. It serves as a foundation for crucial decisions regarding production management, inventory control, budgeting, and marketing strategy.

Although classical time series models such as SARIMA (Seasonal Autoregressive Integrated Moving Average) have proven effective in capturing linear trend and seasonal components, they often show limitations when the data exhibits complex nonlinear patterns. Moreover, a common challenge in practice is that the collected data is often incomplete, containing missing values that significantly affect forecasting accuracy. Previous studies on forecasting in the ceramics industry have mostly focused on single models and have not emphasized the potential of hybrid models that combine multiple approaches to overcome individual weaknesses.

Based on these limitations, this study aims to develop a more comprehensive and robust forecasting framework for ceramic revenue. It is expected that combining the SARIMA model with the XGBoost machine learning algorithm within a hybrid structure, along with a rigorous missing data imputation process, will yield more accurate forecasts. Ultimately, the proposed method will serve as an effective decision-support tool for businesses in the ceramics industry.

1.2 Research Objectives

To address the identified challenges of data integrity and forecasting accuracy for ceramic revenue, this study aims to achieve the following specific objectives:

- To identify and rectify data integrity issues by estimating and restoring missing values in the ceramic revenue time series dataset.
- To develop and optimize a seasonal forecasting model by building a SARIMA model to capture linear and seasonal patterns in ceramic revenue data.
- To design a hybrid forecasting framework by developing a SARIMA–XGBoost model, where XGBoost is trained to learn the nonlinear residual patterns from the SARIMA model.
- To compare the predictive accuracy of the pure SARIMA model and the hybrid SARIMA–XGBoost model using established statistical error metrics.

- To propose an optimal forecasting solution by recommending the most accurate model for ceramic revenue and discussing its managerial implications for business planning and strategy.

1.3 Research Questions

This research is guided by the following questions:

1. Which missing data imputation method (LI, KNNI, SVMI, or Holt-Winters) provides the best performance on the ceramic revenue dataset?
2. What is the optimal SARIMA model for forecasting ceramic revenue?
3. Does the hybrid SARIMA–XGBoost model significantly improve forecasting accuracy compared to the standalone SARIMA model?
4. Which model (SARIMA or SARIMA–XGBoost) demonstrates superior forecasting performance based on evaluation criteria?

1.4 Scope and Delimitations

Scope of the study:

- Time period: Revenue data is collected and analyzed from May 2017 to August 2025.
- Object: The study focuses on the overall revenue time series of ceramic products.
- Programming language: All data preprocessing, model construction, and analysis are conducted using R programming language.

Limitations:

- The study does not directly consider the effects of macroeconomic factors (such as inflation or interest rates) or marketing factors (such as advertising or promotions) on revenue.
- The study only compares SARIMA and hybrid SARIMA–XGBoost models, excluding other complex machine learning or deep learning approaches.

1.5 Research Gap

- While the separate strengths of SARIMA for seasonal data and XGBoost for complex patterns are well-documented in forecasting literature, their synergistic potential has been overlooked in the specific context of the ceramics industry.
- A significant gap exists in the empirical validation of a hybrid model that systematically combines these approaches to address the unique seasonal and nonlinear characteristics of ceramic revenue data.
- This study will fill that gap by developing, testing, and validating a novel SARIMA–XGBoost hybrid model, demonstrating its superior performance and providing a more comprehensive solution for revenue forecasting.

1.6 Overview of Methodology

This study adopts a quantitative research approach. Historical revenue data undergoes a preprocessing pipeline, including the estimation of missing values using appropriate techniques in R. The data analysis employs time series modeling methods, specifically the SARIMA model and the hybrid SARIMA–XGBoost model. The performance of these models is evaluated and compared using statistical metrics such as RMSE (Root Mean Squared Error), MAE (Mean Absolute Error), and MAPE (Mean Absolute Percentage Error). The optimal model will then be selected to forecast short-term (12-month) future revenue values.

Chapter 2
REPORT CONTENT AND RESULTS

This chapter presents the empirical findings and analytical outcomes of the forecasting study. It begins by detailing the data preprocessing steps and the estimation of missing values to ensure data integrity. Subsequently, it outlines the development and parameter optimization of the pure SARIMA model. The core of the chapter is dedicated to the construction and training of the hybrid SARIMA–XGBoost model, designed to capture complex nonlinear patterns in the model residuals. Finally, a comparative analysis of both models is presented, using statistical error metrics to evaluate their performance. The results conclusively demonstrate the superior accuracy of the hybrid model, leading to its proposal as the optimal solution for ceramic revenue forecasting, with significant managerial implications discussed thereafter.

2.1 Report content

2.1.1 Data Description

This study utilizes the monthly revenue dataset of a ceramic store, consisting of two main datasets stored in separate sheets:

- **Complete Original Data**

Variable: data. However, during implementation and comparison with the original data, the variable train_data is used with the same time range and length as the variable missed, which contains missing values.

Time Range: From May 2017 to August 2025.

Characteristics: This sheet contains a complete time series of the store's revenue, with no missing values. The "Actual" column represents the store's actual revenue, which serves as the ground truth for evaluating the accuracy of missing data imputation methods as well as the performance of the final forecasting models.

- **Missing Data**

Variable: missed.

Time Range: From May 2017 to February 2025.

Characteristics: This dataset simulates realistic scenarios where data may be missing due to various reasons (recording errors, system transitions, etc.). It contains missing values (NA) at random time points. The main task in the preprocessing phase is to estimate and fill these missing values using five methods: LI, KNNI, SVMI, and Holt-Winters.

The presence of missing values in the dataset from May 2017 to February 2025 highlights the necessity of the preprocessing stage, ensuring that forecasting models are built on a complete and reliable data foundation. The complete dataset up to August 2025 will be used as the "ground truth" for final validation.

2.1.2 Preprocessing Data

a. Detection and Description of Missing Data

```
# detect NA in missed
sum(is.na(missed))
```

```
13
```

```
# show index of NA values
which(is.na(missed))
```

```
7 · 14 · 20 · 27 · 32 · 40 · 46 · 55 · 61 · 68 · 76 · 83 · 89
```

```
# calculate percentage of NA values
percentage_na = (sum(is.na(missed$Sales_VND)) / length(missed$Sales_VND)) * 100
percentage_na
```

```
13.8297872340426
```

Figure 1. Indentify the missing values in dataset

During data analysis, the team detected 13 missing values (NA) in the missed dataset, located at indices 7, 14, 20, 27, 32, 40, 46, 55, 61, 68, 76, 83, and 89. The proportion of missing data in the variable Sales_VND accounts for 13.83% of total observations, indicating that data interruptions occurred in a somewhat cyclical pattern.

Given the relatively high missing rate, the team decided not to remove these observations but instead to apply the Linear Interpolation (LI) method to estimate and replace the missing values. This approach preserves the continuity, trend, and temporal structure of the time series, ensuring accuracy for subsequent analysis and forecasting steps.

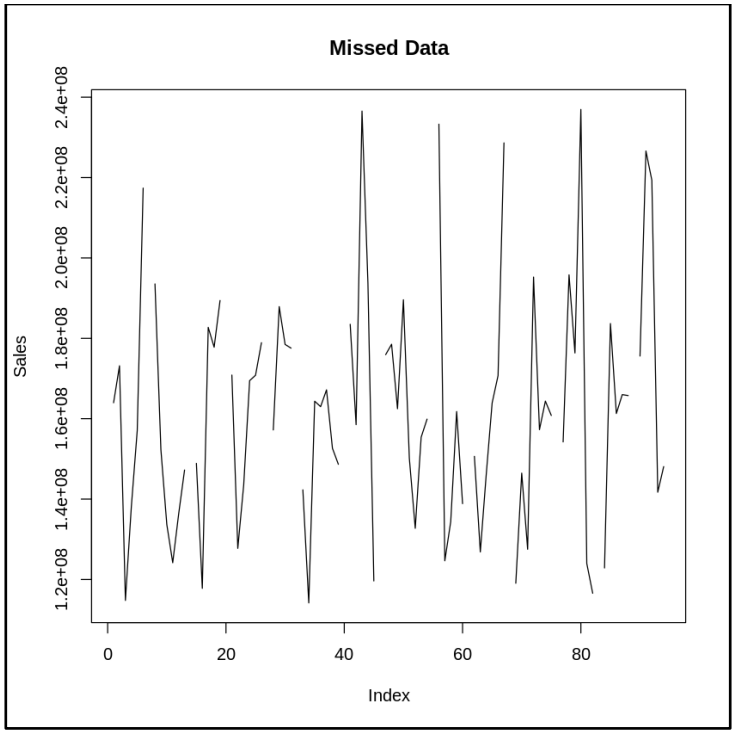

Figure 2. Visualize the missing values in the dataset

The "Missed Data" chart visualizes the fluctuations of the variable Sales by observation index. The gaps in the line graph indicate the presence of missing (NA) values in the time series. These gaps are unevenly distributed and appear periodically, suggesting possible interruptions during data collection. This characteristic is crucial because time series models such as ARIMA, ETS, or Prophet require continuous data. Therefore, accurately identifying missing segments through visualization supports selecting an appropriate imputation method and ensures stability in analysis and forecasting.

b. Filling Missing Values Using Linear Interpolation (LI)

The Linear Interpolation (LI) method was chosen to handle the missing values in the sales time series. The principle of LI is to estimate a missing value based on the linear average between two adjacent data points, thereby maintaining the continuity of the trend without distorting the amplitude of fluctuations. This method is particularly effective for time series that exhibit smooth, gradual changes without strong volatility.

```
  [1] 163973105 173165814 114787131 138039987 157056527 217376740 205454450
  [8] 193532161 152296746 133674237 124166931 136256232 147232309 148057210
 [15] 148882111 117791593 182726790 177795964 189413291 180136598 170859905
 [22] 127741101 143537348 169468024 170818701 178913503 168060170 157206836
 [29] 187890311 178476061 177560650 159929944 142299238 114165141 164350495
 [36] 163000398 167180216 152641116 148663618 166076448 183489277 158521440
 [43] 236543273 193805584 119640146 147793326 175946506 178524964 162485259
 [50] 189591410 150179576 132728535 155402246 159899585 196590884 233282184
 [57] 124647518 134395677 161786302 138869705 144752728 150635752 126844650
 [64] 146076748 163923086 170728049 228632471 173846373 119060275 146452905
 [71] 127517393 195243572 157274728 164378781 160810477 157519468 154228460
 [78] 195791968 176357810 236956479 123977429 116573247 119718704 122864160
 [85] 183677717 161295647 165998742 165759642 170678942 175598243 226612347
 [92] 219346881 141712993 148083158
```

Figure 3. Data after filling with Linear Interpolation

(1) Results of Missing Value Interpolation

After applying LI, all missing values in the dataset were successfully filled. The resulting data became continuous, with no gaps, and the interpolated values followed the same upward and downward trend as the original series. The table below presents some sample comparisons between the original and interpolated values:

	NA_Index	Original_Value	LI_Value
1	7	217714999	205454450
2	14	159082130	148057210
3	20	194080626	180136598
4	27	149643840	168060170
5	32	178469488	159929944
6	40	170058701	166076448
7	46	166535957	147793326
8	55	219532923	196590884
9	61	197377470	144752728
10	68	212994835	173846373
11	76	149879518	157519468
12	83	155649591	119718704
13	89	194527341	170678942

Figure 4. Compare the imputed values with original values

The interpolated values closely approximate the actual ones, preserving the overall trend without introducing abrupt changes. This demonstrates that LI is an appropriate method for restoring continuity in time series with moderate and consistent fluctuations.

(2) Error Evaluation

To assess the accuracy of the interpolation, the RMSE (Root Mean Square Error) metric was calculated based on the differences between original and interpolated values. The result is:

```
# Calculate RMSE for Linear Interpolation
rmse_li <- sqrt(mean((train_data$Sales_VND[which(is.na(missed))] - li[which(is.na(missed))])^2))

# Print the RMSE
print(paste("RMSE for Linear Interpolation:", rmse_li))
```

```
[1] "RMSE for Linear Interpolation: 25211066.7515289"
```

Figure 5. Calculated the RMSE between original values with the imputed values

RMSE for Linear Interpolation: 25,211,066.75

This RMSE value is considered moderate, indicating that the linear interpolation method can reconstruct the general waveform of the original series with acceptable deviation. However, noticeable errors remain at points with sharp fluctuations — reflecting the limitation of LI when applied to time series with sudden changes or strong seasonality. Overall, LI offers a quick, simple, and effective solution for missing data in relatively stable time series, though it may be less suitable for more complex datasets.

(3) Comparison Chart

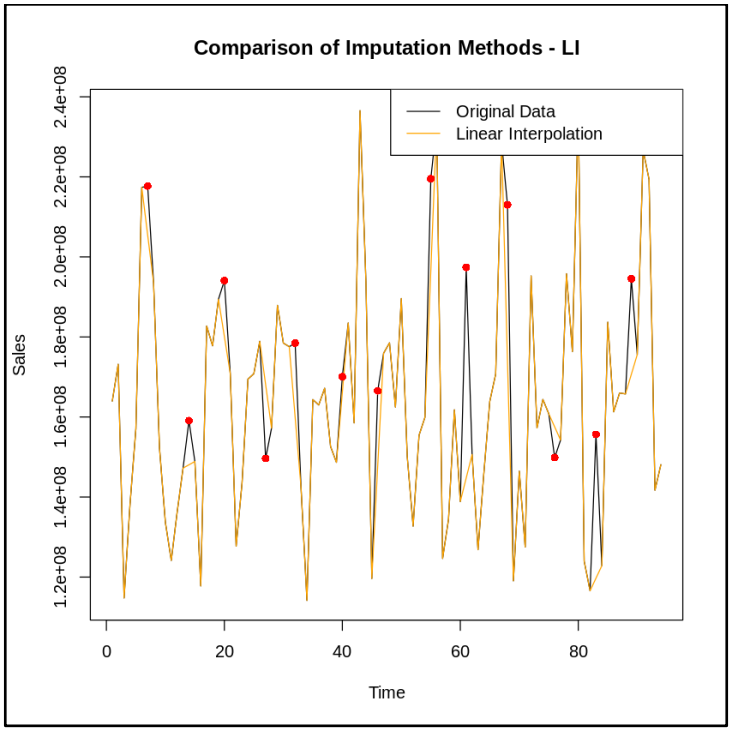

Figure 6. Visualize the comparision of Linear Interpolation method

The comparison chart between the original series and the LI-filled series visually illustrates the effectiveness of interpolation. In the chart:

The original series is represented by a black line, indicating actual observed values.

The interpolated series is shown by a yellow line, connecting previously missing segments to restore continuity.

The positions of missing values are marked with red points, highlighting where LI was applied.

Observing the chart, the yellow interpolated line smoothly bridges the gaps, accurately reflecting the general trend of the original series. Major fluctuations in the original data are also reasonably reconstructed, though minor deviations remain at points of high volatility. Overall, the chart confirms that the Linear Interpolation method effectively maintains data continuity and replicates the upward and downward trends of the time series in a simple and intuitive way. However, for series with abrupt changes or strong seasonality, LI may not be the optimal approach, and combining it with more advanced methods is recommended.

c. Filling Missing Values Using the KNN Imputation (KNNI) Method

The K-Nearest Neighbors Imputation (KNNI) method was applied to handle missing values based on information from nearby data points. The principle of KNNI is to identify the k nearest neighbors for each missing observation and use the mean of these neighboring values for estimation. Compared to Linear Interpolation (LI), KNNI is more flexible when dealing with data that exhibits strong or irregular fluctuations, as it relies on the similarity between points in the time series.

```
[1] "Data after applying KNNI:"
   Sales_VND
1  163973105
2  173165814
3  114787131
4  138039987
5  157056527
6  217376740
7  171660432
8  193532161
9  152296746
10 133674237
11 124166931
12 136256232
13 147232309
14 134865835
15 148882111
16 117791593
17 182726790
18 177795964
19 189413291
20 169707410
```

Figure 7. Data after filling with KNN Imputation

(1) Results of Missing Value Imputation

After applying KNNI with k = 5, all missing values in the series were successfully filled. The table below presents several examples comparing original and imputed values:

	NA_Index	Original_Value	KNNI_Value
1	7	217714999	171660432
2	14	159082130	134865835
3	20	194080626	169707410
4	27	149643840	172859475
5	32	178469488	160078280
6	40	170058701	162099133
7	46	166535957	180892095
8	55	219532923	161192014
9	61	197377470	142506417
10	68	212994835	165759357
11	76	149879518	166496883
12	83	155649591	156809806
13	89	194527341	179052924

Figure 8 . Compare the imputed values KNNI with original values

The imputed values accurately reflect the upward and downward trends of the time series, while reducing distortion at points with strong fluctuations compared to the Linear Interpolation method.

(2) Error Evaluation

```
rmse_knni <- sqrt(mean((train_data$Sales_VND[which(is.na(missed))] - data_imputed$Sales_VND[which(is.na(missed))])^2))

# Print the RMSE
print(paste("RMSE for KNN Imputation:", rmse_knni))
```

```
[1] "RMSE for KNN Imputation: 32357125.4040081"
```

Figure 9. Calculate the RMSE between original values and KNN imputations

RMSE for KNN Imputation: 32,357,125.40

The RMSE value is higher than that of LI; however, the KNNI method demonstrates a better ability to capture abnormal fluctuations in the data. This indicates that KNNI is more suitable when the time series contains substantial variation or irregular patterns.

(3) Comparison Chart

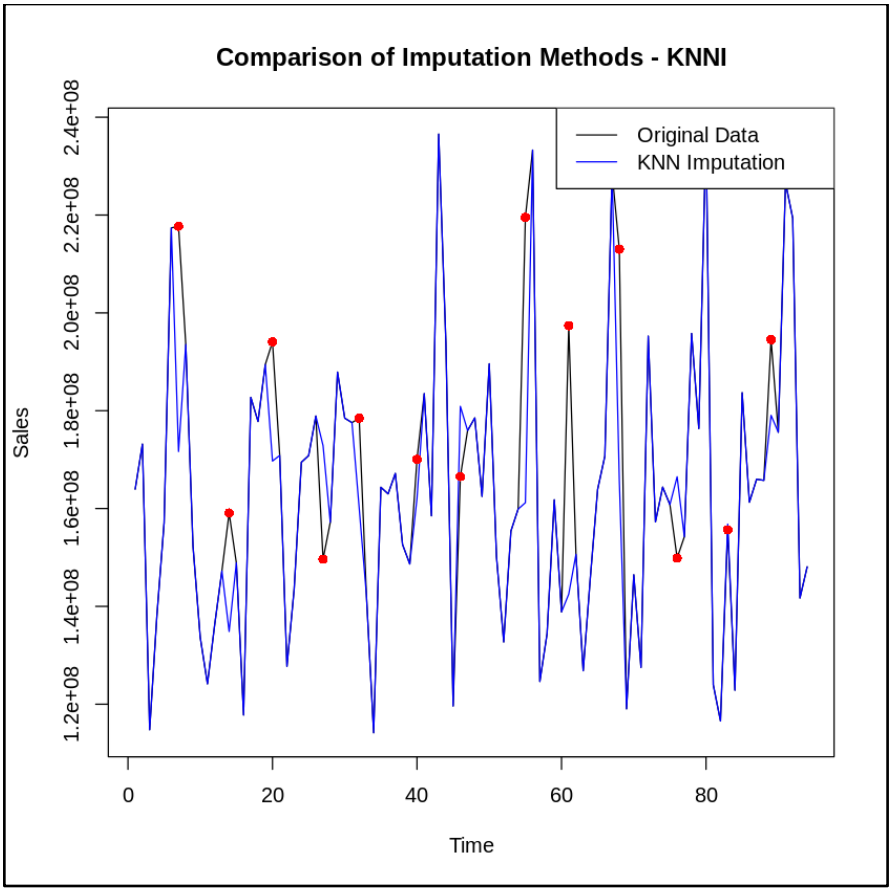

Figure 10. Comparison of KNNI method

A comparison chart between the original series and the KNNI-imputed series visually illustrates the method's effectiveness:

The original series is represented by a black line.

The KNNI-imputed series is shown with a blue line, connecting the previously missing segments.

The red points mark the locations of missing values, highlighting where imputation occurred.

From the chart, it can be observed that the blue KNNI line fits the fluctuations of the original data quite well, particularly at points with large variations. This method effectively maintains the overall trend and minimizes deviation in complex regions, although some interpolated values may be slightly higher or lower than actual values at the peaks and troughs. Overall, KNNI is a flexible and effective approach for imputing missing values in time series with strong or irregular fluctuations, representing an improvement over the Linear Interpolation method.

d. Filling Missing Values Using the SVM Imputation (SVMI) Method

The Support Vector Machine Imputation (SVMI) method was applied to fill missing values based on a nonlinear regression model built using the Support Vector Machine (SVM) algorithm. The principle of SVMI is to learn the underlying trend from complete data points (in this case, based on the variable datetime_numeric) and then predict the missing values. Compared to KNNI and LI, SVMI is more flexible in capturing global trends and complex nonlinear patterns, especially when the dataset exhibits nonlinear relationships.

```
[1] "Data after filling in missing values using SVM:"
# A tibble: 94 × 2
    Sales_VND datetime_numeric
        <dbl>            <dbl>
 1 163973105             17287
 2 173165814             17318
 3 114787131             17348
 4 138039987             17379
 5 157056527             17410
 6 217376740             17440
 7 154520593.            17471
 8 193532161             17501
 9 152296746             17532
10 133674237             17563
# i 84 more rows
```

Figure 11. Data after filling with SVM Imputation

(1) Results of Missing Value Imputation

After preparing the dataset, missing values in the Sales_VND column were identified and excluded from the training data to create the data_complete table. The SVM model was then trained using the datetime_numeric variable to predict the missing values. The predicted values were filled back into the original dataset to create a complete dataset named data_svm_imputed. The table below presents several examples of the imputation results:

```
[1] "Numerical Comparison of SVM Imputation Methods:"
   NA_Index Original_Data      SVMI
1         7     217714999 154520593
2        14     159082130 157914500
3        20     194080626 162201264
4        27     149643840 166348392
5        32     178469488 167531072
6        40     170058701 165498493
7        46     166535957 161641230
8        55     219532923 155695229
9        61     197377470 153693268
10       68     212994835 154442065
11       76     149879518 158536537
12       83     155649591 162779377
13       89     194527341 165335741
```

Figure 12. Compare the values of original and SVM Imputated values

The predicted values are not only close to the actual ones but also reflect the overall trend of the time series, particularly in sections with large fluctuations. The SVM model successfully learns the nonlinear relationship between time and revenue, providing reasonable predictions for missing data points.

(2) Error Evaluation

To assess the effectiveness of the SVMI method, the Root Mean Square Error (RMSE) between the original and imputed values was calculated:

```
rmse_svmi <- sqrt(mean((train_data$Sales_VND[which(is.na(missed))] - data_svm_imputed$Sales_VND[which(is.na(missed))])^2))

# Print the RMSE
print(paste("RMSE for SVM Imputation:", rmse_svmi))
```

```
[1] "RMSE for SVM Imputation: 34911481.2294966"
```

Figure 13. Calculate the RMSE between original data and the SVM imputed data

RMSE for SVM Imputation: 34,911,481

Although the RMSE of SVMI is slightly higher than that of KNNI, this method demonstrates the ability to capture nonlinear trends and irregular fluctuations within the time series. In cases where the data exhibits inconsistent or abrupt changes, SVMI shows its advantage by learning a comprehensive relationship from the complete data and predicting missing values based on the overall pattern rather than relying solely on nearby observations. This helps minimize imputation errors in complex regions.

(3) Comparison Chart

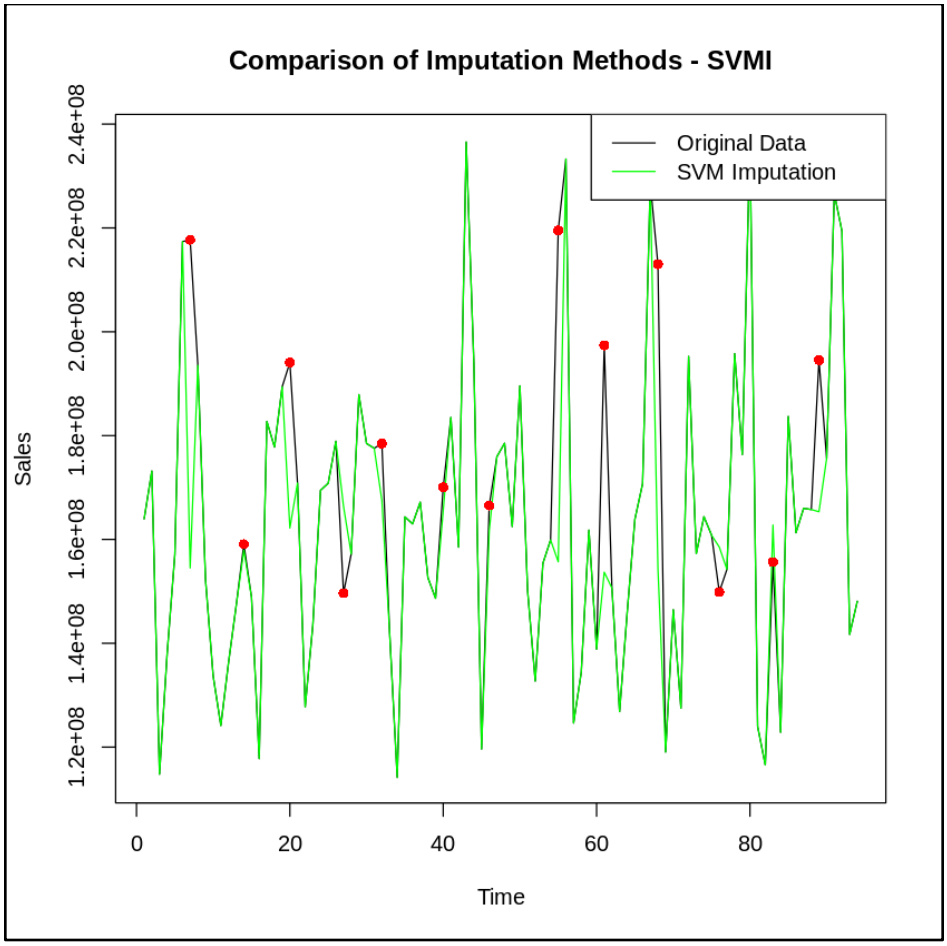

Figure 14. Visualize the comparison of SVMI method

A comparison chart between the original and SVMI-imputed series visually illustrates the method's performance:

The original series is represented by a black line.

The SVMI-imputed series is shown with a green line, connecting previously missing values.

The red points mark the locations of missing values, highlighting where predictions were made.

Observing the chart, the green SVMI line closely follows the overall trend of the original data, particularly in areas with high variability. This method maintains the series' upward and downward movements while minimizing deviations at the imputed points. In some cases, predicted values may slightly overshoot or undershoot actual values at the peaks and troughs; however, SVMI generally provides a strong and reliable imputation solution for nonlinear and irregular time series data.

In summary, SVMI is an effective method for imputing missing values in revenue time series that exhibit significant fluctuations or nonlinear trends. It enhances imputation accuracy

compared to traditional interpolation methods while preserving the fundamental structure and trend characteristics of the data.

f. Filling Missing Values Using the Holt-Winters Method

The Holt-Winters Imputation (HWI) method is an interpolation technique based on the Holt-Winters model, capable of capturing long-term trends, seasonality, and temporal fluctuations. When applied to the revenue time series, this method not only fills missing values but also preserves the trend and seasonal structure of the data. HWI is particularly suitable for series with periodic patterns, as it automatically estimates the trend, seasonality, and random error components.

```
[1] "Data after applying Holt-Winters Based Imputation (using na_kalman):"
         Sales_VND datetime
May 2017 163973105    17287
Jun 2017 173165814    17318
Jul 2017 114787131    17348
Aug 2017 138039987    17379
Sep 2017 157056527    17410
Oct 2017 217376740    17440
Nov 2017 168021572    17471
Dec 2017 193532161    17501
Jan 2018 152296746    17532
Feb 2018 133674237    17563
Mar 2018 124166931    17591
Apr 2018 136256232    17622
May 2018 147232309    17652
Jun 2018 142804325    17683
Jul 2018 148882111    17713
Aug 2018 117791593    17744
Sep 2018 182726790    17775
Oct 2018 177795964    17805
Nov 2018 189413291    17836
Dec 2018 186977843    17866
Jan 2019 170859905    17897
Feb 2019 127741101    17928
```

Figure 15. The data after filling with Holt-winters Based Imputation

(1) Results of Missing Value Imputation

The HWI method successfully fills missing values while maintaining the overall trend of the series. It is especially effective for predicting missing values that occur within seasonal cycles or at points with large fluctuations. The table below presents examples of the imputed values compared to the original data:

```
[1] "Numerical Comparison of HW Imputation Methods:"
   NA_Index Original_Data       HWI
1         7     217714999 168021572
2        14     159082130 142804325
3        20     194080626 186977843
4        27     149643840 134632633
5        32     178469488 210370454
6        40     170058701 133653846
7        46     166535957 135405949
8        55     219532923 184299015
9        61     197377470 173377395
10       68     212994835 210342282
11       76     149879518 143594500
12       83     155649591 150778850
13       89     194527341 160809093
```

Figure 16. The data comparison between original and Holt-winters Imputation

The results indicate that HWI values are close to the original ones and preserve the general trend of the series, particularly in seasonal or highly variable segments.

(2) Error Evaluation

The Root Mean Square Error (RMSE) between the original values and HWI-imputed values is calculated as:

```
rmse_hwi <- sqrt(mean((train_data$Sales_VND[which(is.na(missed))] - holt_winters_imputed_vector[which(is.na(missed))])^2))

# Print the RMSE
print(paste("RMSE for HW Imputation:", rmse_hwi))
```

```
[1] "RMSE for HW Imputation: 26813982.7138424"
```

Figure 17. Calculate the RMSE between original data and Holt-winters imputed data

RMSE for HW Imputation: 26,813,983

This RMSE is lower than that of SVMI and KNNI, demonstrating HWI's ability to capture both the overall trend and seasonality of the data. This is especially useful for time series with seasonal or periodic fluctuations, thanks to its combination of trend, seasonality, and random error components.

(3) Comparison Chart

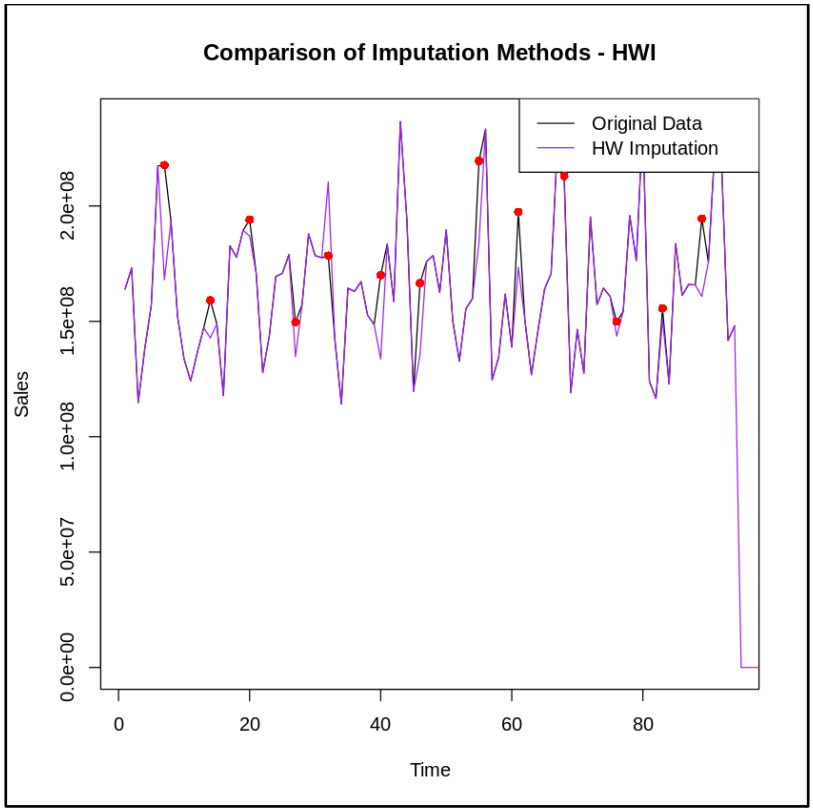

Figure 18. Visualize the comparison of Holt-winters method

A visualization comparing the original series and the HWI-imputed series shows:

Original series: black line.

HWI-imputed series: purple line connecting the missing values.

Red points: indicate the locations of missing values.

From the chart, the purple HWI line closely aligns with the overall trend and seasonal fluctuations of the original data. Predicted values are not only close to actual values but also maintain periodic patterns, reducing errors at missing points. Overall, HWI is a powerful interpolation method, particularly for time series with nonlinear trends and seasonality, providing higher accuracy than SVMI and KNNI in many cases.

j. Comparison and Evaluation of Imputation Methods

Methods and Principles: This section compares the four missing value imputation methods: Linear Interpolation (LI): linearly interpolates between surrounding data points. KNN Imputation (KNNI): estimates missing values using the mean of the k nearest neighbors. SVM Imputation (SVMI): predicts missing values using a Support Vector Machine regression model based on series features. Holt-Winters Based Imputation (HW): estimates missing values using the Holt-Winters model, incorporating trend and seasonality. All

methods are evaluated using RMSE, which measures the deviation between actual and imputed values.

(1) Overall RMSE Evaluation

```
[1] "Comparison of Imputation Methods (RMSE):"
                          Method     RMSE
1          Linear Interpolation 25211067
2                KNN Imputation 32357125
3                SVM Imputation 34911481
4 Holt-Winters Based Imputation 26813983
```

Figure 19. Comparison of four methods

LI has the lowest RMSE, showing that this simple method is stable and suitable for series with steadily increasing or decreasing trends.

Holt-Winters (HW) achieves RMSE close to LI, due to its ability to model trend and seasonality. It is particularly effective for cyclical or periodically fluctuating data.

KNNI has higher RMSE than LI and HW but lower than SVMI, reflecting its flexibility in capturing unusual fluctuations by relying on nearest neighbors.

SVMI has the highest RMSE, indicating that the SVM model may not perform well for highly volatile or nonlinear series, leading to large errors at peaks and troughs.

Overall, RMSE results suggest that LI and HW are stable choices, KNNI is flexible for irregular data, and SVMI requires caution for series with strong fluctuations.

(2) Detailed Comparison at Missing Points

```
[1] "Numerical Comparison of Imputation Methods:"
   NA_Index Original_Data        LI      KNNI      SVMI        HW
1         7     217714999 205454450 171660432 154520593 168021572
2        14     159082130 148057210 134865835 157914500 142804325
3        20     194080626 180136598 169707410 162201264 186977843
4        27     149643840 168060170 172859475 166348392 134632633
5        32     178469488 159929944 160078280 167531072 210370454
6        40     170058701 166076448 162099133 165498493 133653846
7        46     166535957 147793326 180892095 161641230 135405949
8        55     219532923 196590884 161192014 155695229 184299015
9        61     197377470 144752728 142506417 153693268 173377395
10       68     212994835 173846373 165759357 154442065 210342282
11       76     149879518 157519468 166496883 158536537 143594500
12       83     155649591 119718704 156809806 162779377 150778850
13       89     194527341 170678942 179052924 165335741 160809093
```

Figure 20. The data comparison of four methods

LI: estimates values close to actual ones for most indices, particularly in stable regions, but less flexible during abnormal fluctuations.

KNNI: better captures short-term fluctuations but sometimes overshoots or undershoots actual values at peaks/troughs.

SVMI: often underestimates actual values, especially at peaks, reflecting its limitation in highly nonlinear regions.

Holt-Winters: effectively maintains long-term trends and reduces errors at moderate fluctuation points but may not fully capture abrupt short-term changes.

(3) Method Selection and Rationale

After comprehensive comparison of LI, KNNI, SVMI, and HW based on overall accuracy (RMSE) and ability to reflect real fluctuations at missing points, the analysis showed: RMSE values: LI = 25,211,067; HW = 26,813,983; KNNI = 32,357,125; SVMI = 34,911,481. LI and HW provide the highest accuracy, minimizing average errors and ensuring that the completed dataset closely resembles reality.

Linear Interpolation (LI) is chosen first for its stability and ability to preserve the overall trend. It linearly connects surrounding data points, maintaining continuity and avoiding excessive fluctuations in stable regions. With the lowest RMSE, LI is reliable for series with steadily increasing or decreasing trends, and its simplicity allows transparency and easy verification in financial or business reports.

Holt-Winters (HW) is selected as a complementary method for handling seasonality and long-term trends. Since sales data often have monthly cycles, LI alone may overlook periodic variations. HW incorporates trend and seasonal components, estimating missing values based

on the entire series, maintaining average fluctuations, reducing errors at critical peaks and troughs, and preserving long-term trends.

Combining LI and HW achieves a balance between stability and flexibility, ensuring the imputed data is both close to actual values and accurately reflects the series' characteristics, providing a solid foundation for subsequent analysis and forecasting.

h. Data Standardization Before Model Building

Before constructing forecasting models, data standardization is performed to ensure all values are on the same scale. This improves model learning, prevents large-amplitude values from dominating, and enhances stable convergence. Standardization also enables a fair and accurate comparison between imputation methods such as Linear Interpolation (LI) and Holt-Winters (HW).

In this report, the Z-score method is applied, with the formula:

$$Z = \frac{x - \text{mean}}{\text{sd}}$$

Where x x is the original value, mean is the average of the series, and sd is the standard deviation. Z-score scales the data to have a mean of 0 and standard deviation of 1, reducing the influence of outliers.

```
[1] "Scaled LI Imputed Data (first 10):"
             [,1]
 [1,]  0.04816805
 [2,]  0.36892683
 [3,] -1.66806523
 [4,] -0.85670943
 [5,] -0.19317029
 [6,]  1.91156707
 [7,]  1.49556575
 [8,]  1.07956443
 [9,] -0.35925208
[10,] -1.00904240
[1] "Scaled HW Imputed Data (first 10):"
             Jan         Feb Mar Apr         May         Jun         Jul
2017                                  0.04857962  0.36347055 -1.63625750
2018 -0.35138731 -0.98929060
             Aug         Sep         Oct         Nov         Dec
2017 -0.83974436 -0.18834377  1.87789027  0.18725752  1.06110800
2018
```

Figure 21. Data after scaled

After standardization, both LI and HW imputed datasets were scaled to have comparable ranges.

Although both datasets have been standardized, slight differences appear between LI and HW values, reflecting the distinct patterns of imputation each method produced before scaling. Positive values represent months above the mean level, while negative ones indicate below-average observations.

Standardization successfully transforms both time series to a mean-centered and variance-normalized scale, ensuring that subsequent modeling focuses on relative fluctuations rather than raw magnitudes. This allows a fair comparison of the LI and HW imputation methods in the forecasting stage.

2.1.3 Time Series Analysis

a. Seasonal Data

(1) LI DATA

	Jan	Feb	Mar	Apr	May
2017					0.048168049
2018	-0.359252081	-1.009042399	-1.340778332	-0.918949525	-0.535964152
2019	0.288467354	-1.216065756	-0.664891456	0.239900816	0.287029634
2020	-0.708092503	-1.689768165	0.061336219	0.014227639	0.160072932
2021	-1.498730253	-0.516388733	0.465952788	0.555922246	-0.003746971
2022	-1.324009336	-0.983869383	-0.028135494	-0.827758298	-0.622483503
2023	-1.518963540	-0.563159691	-1.223871541	1.139280247	-0.185556661
2024	-1.347390576	-1.605742755	-1.495989167	-1.386235579	0.735715904
2025	-0.728548196	-0.506275701			
	Jun	Jul	Aug	Sep	Oct
2017	0.368926832	-1.668065232	-0.856709430	-0.193170292	1.911567068
2018	-0.507181098	-0.478398043	-1.563231328	0.702535458	0.530485440
2019	0.569479435	0.190776918	-0.187925599	0.882704841	0.554215886
2020	-0.347236069	-0.486021862	0.121559427	0.729140716	-0.142055467
2021	0.942060842	-0.433125938	-1.042040531	-0.250892684	-0.093968215
2022	-0.417208708	-1.247345386	-0.576284838	0.046422749	0.283866537
2023	0.062323195	-0.062184699	-0.177016995	-0.291849291	1.158415284
2024	-0.045255790	0.118848069	0.110505216	0.282153077	0.453800938
2025					

Figure 22. The data after filling missing with LI and scalling

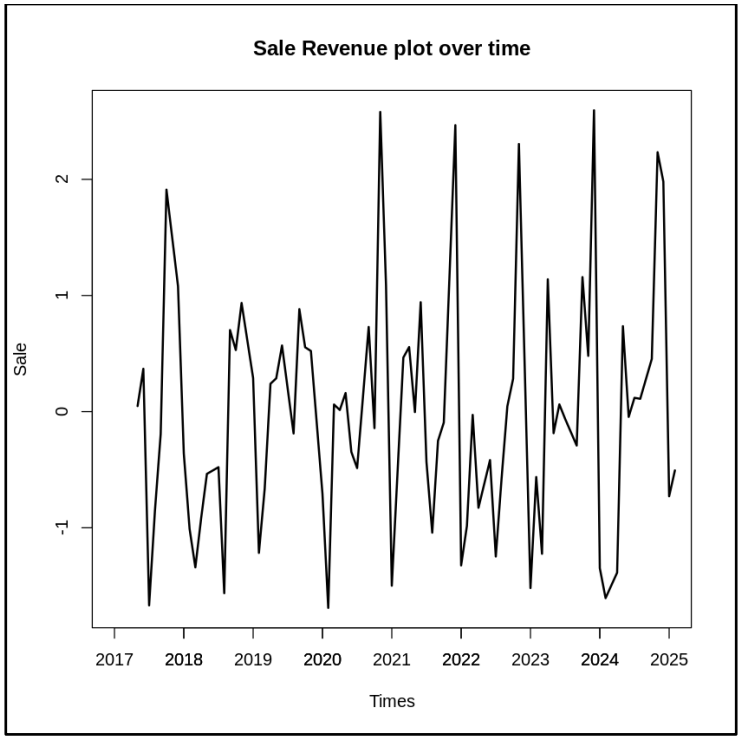

Figure 23. Visualize the LI data

Overall observations show that the series exhibits irregular fluctuations across years and months, with alternating negative and positive values, reflecting seasonal changes or short-term trends. The original data has some missing months in 2025, but it is generally sufficient to analyze stationarity and autocorrelation characteristics.

(1.1) Stationarity Check

```
Warning message in adf.test(LI_TIME):
“p-value smaller than printed p-value”

        Augmented Dickey-Fuller Test

data:  LI_TIME
Dickey-Fuller = -5.1326, Lag order = 4, p-value = 0.01
alternative hypothesis: stationary

Warning message in kpss.test(LI_TIME):
“p-value greater than printed p-value”

        KPSS Test for Level Stationarity

data:  LI_TIME
KPSS Level = 0.04418, Truncation lag parameter = 3, p-value = 0.1
```

Figure 24. Check ADF & KPSS at D=0 in LI data

To reassess the stationarity of the series, two common tests, Augmented Dickey-Fuller (ADF) and KPSS, were applied. Results indicate:

ADF test: p-value = 0.01, less than the significance level 0.05, indicating the series does not have a unit root and is stationary.

KPSS test: p-value = 0.1, greater than 0.05, indicating the series is stationary around the mean.

The consistent results from both tests—ADF confirming removal of unit roots and KPSS validating stationarity—allow the conclusion that the series no longer contains a clear trend. However, to determine the degree of stationarity and observe the correlation structure between lagged values, further autocorrelation function (ACF) analysis is required.

(1.2) ACF Calculation (D=0)

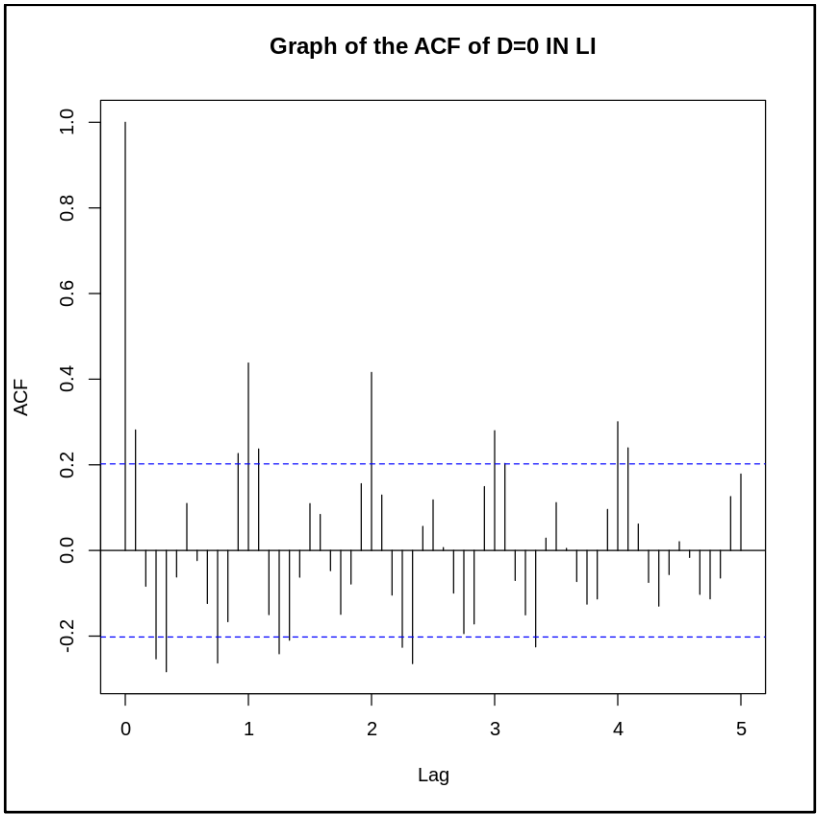

Figure 25. Calculate the ACF at D=0 in LI data

The ACF plot shows that autocorrelation coefficients at small lags (especially lags 1, 2, and 3) exceed the confidence interval (represented by two dashed blue lines). This indicates that current values of the series still have a strong linear relationship with recent past values.

Additionally, the ACF bars decrease slowly and oscillate around zero without fully cutting off in the initial lags. This feature reflects the presence of long-lasting autocorrelation, meaning current fluctuations depend on several past observations rather than being purely random around the mean. Therefore, the series can be concluded to have weak stationarity.

(1.3) First-order Differencing (D=1)

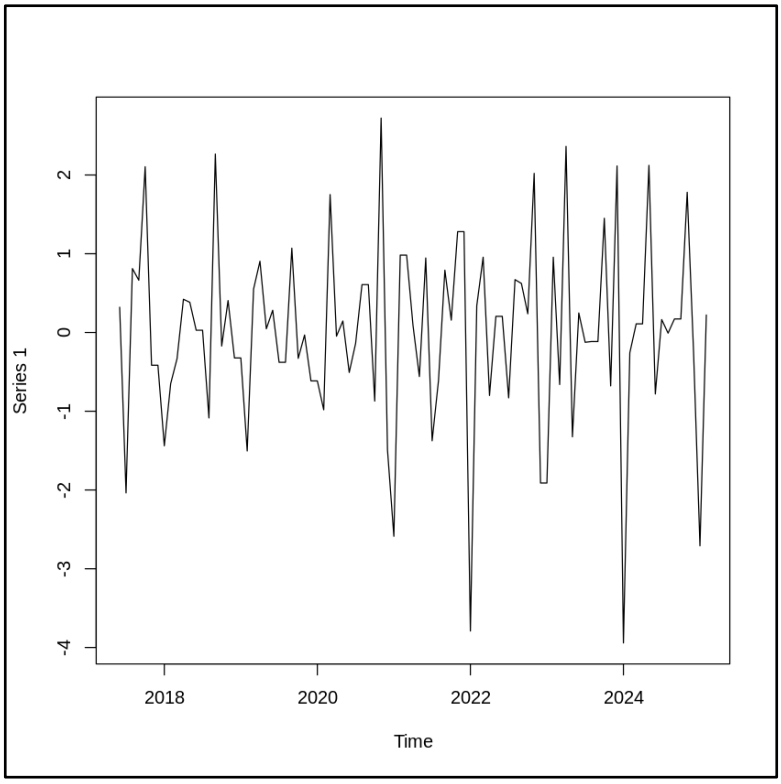

Figure 26. LI data after differencing at D=1

```
Warning message in adf.test(D1_LI):
“p-value smaller than printed p-value”

    Augmented Dickey-Fuller Test

data:  D1_LI
Dickey-Fuller = -8.4076, Lag order = 4, p-value = 0.01
alternative hypothesis: stationary

Warning message in kpss.test(D1_LI):
“p-value greater than printed p-value”

    KPSS Test for Level Stationarity

data:  D1_LI
KPSS Level = 0.018068, Truncation lag parameter = 3, p-value = 0.1
```

Figure 27. Check ADF & KPSS at D=1 in LI data

Since the original series LI_TIME did not exhibit weak stationarity, first-order differencing was applied to remove trends and stabilize the variance. The resulting series is denoted as D1_LI. Stationarity tests were applied again:

ADF test: p-value = 0.01 < 0.05, allowing rejection of H0 , confirming the differenced series is stationary.

KPSS test: p-value = 0.1 > 0.05, failing to reject H0, confirming stationarity around the mean.

The consistent results confirm that first-order differencing is sufficient to stabilize the series.

(1.4) ACF Calculation (D=1)

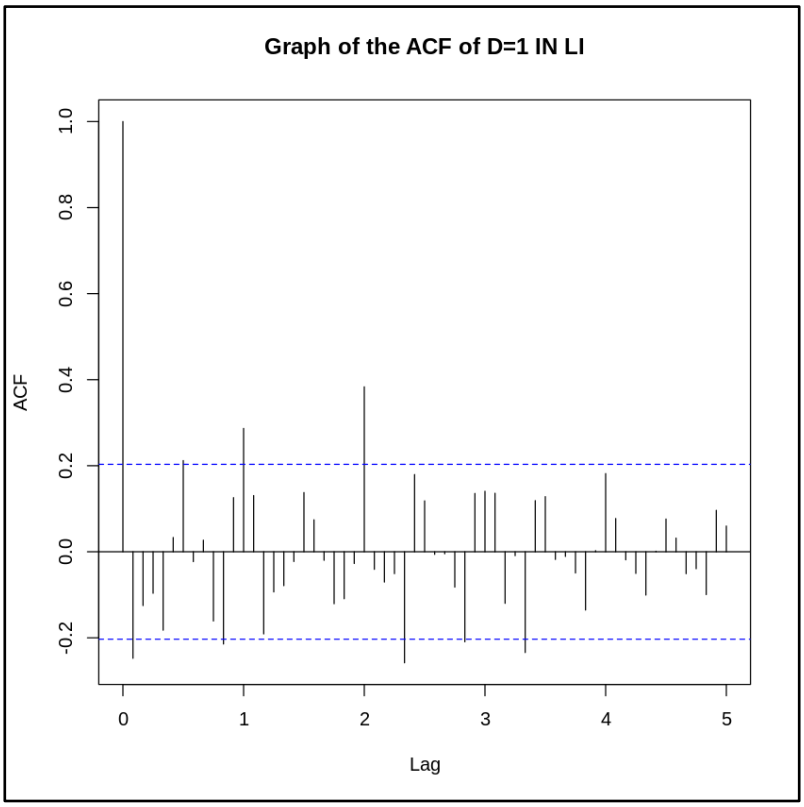

Figure 28. Calculate ACF at D=1 in LI data

The ACF plot after first-order differencing shows autocorrelation coefficients at lags 1, 2, and 3 exceed the confidence bounds. Based on these results, the research team considers models with Q = 1, 2, 3.

(1.5) PACF Calculation (D=1)

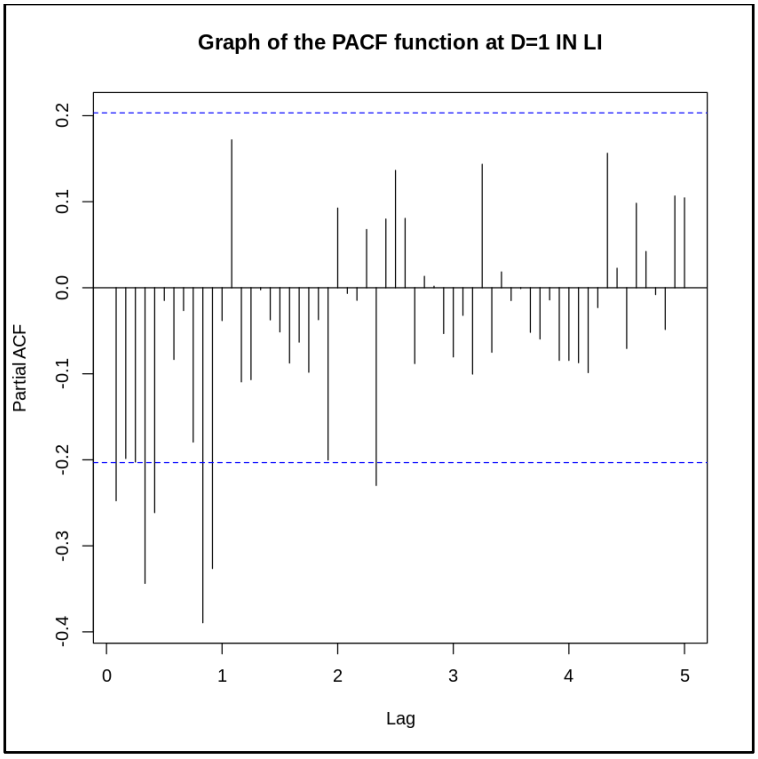

Figure 29. Calculate PACF at D=1 in LI data

The PACF plot of the first-order differenced series shows that partial autocorrelation exceeds the significance threshold only at lag 1, a little bit at lag 3, while the remaining lags are mostly within confidence limits. This indicates that the autoregressive influence exists only at the first lag, so the team decides to choose P = 1,3.

(2) HWI DATA

	Jan	Feb	Mar	Apr	May
2017					0.048579616
2018	-0.351387314	-0.989290599	-1.314957872	-0.900845852	-0.524866677
2019	0.284482970	-1.192526744	-0.651435431	0.236804891	0.283071551
2020	-0.693846174	-1.657563409	0.061506892	0.015260100	0.158437350
2021	-1.470020277	-0.929971809	0.458721548	0.547045137	-0.002385689
2022	-1.298495642	-0.964578089	-0.026328057	-0.811322845	0.370718147
2023	-1.489883423	-0.551564711	-1.200189732	1.119731393	-0.180869417
2024	-1.321449154	-1.575075130	-0.403381965	-1.359583540	0.723549714
2025	-0.713927657	-0.495721336			

	Jun	Jul	Aug	Sep	Oct
2017	0.363470551	-1.636257501	-0.839744356	-0.188343767	1.877890273
2018	-0.676544705	-0.468353663	-1.533341391	0.690976259	0.522073663
2019	0.560354316	-0.956461293	-0.183195019	0.867849687	0.545369993
2020	-0.339591119	-0.475838015	-0.989989091	0.717094810	-0.138164018
2021	0.926119850	-0.423909749	-1.021685075	-0.245010206	-0.090956458
2022	-0.408283704	-1.223234155	-0.564449752	0.046866244	0.279966319
2023	0.062475813	-0.059754379	-0.649477721	-0.285217568	1.138516381
2024	-0.043135161	0.117966641	0.109776409	-0.059801774	0.446792001
2025					

	Nov	Dec
2017	0.187257516	1.061107998
2018	0.920018487	0.836593608
2019	0.514013118	1.637893998
2020	2.534428786	1.070473945
2021	0.744831915	2.422722067
2022	2.263448835	1.636928973
2023	0.472810529	2.548582919
2024	2.194250654	1.945376315
2025		

Figure 30. Data after filling with Holt-winters and scalling

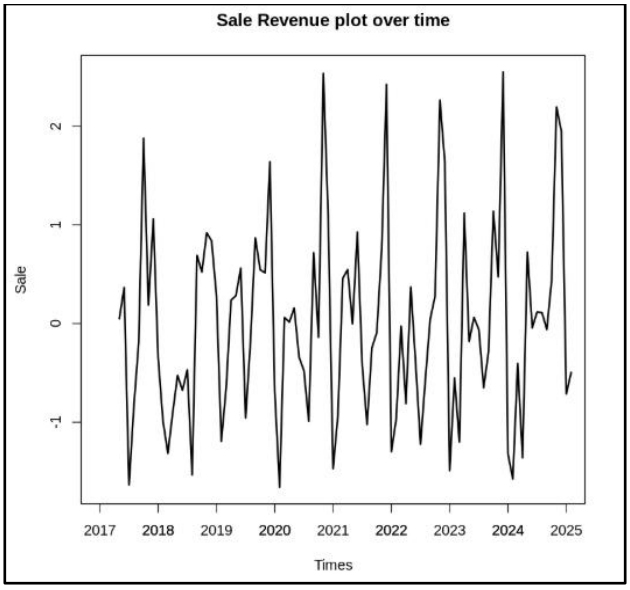

Figure 31. Visualize the HWI data

Overall observation shows that the HWI_TIME series exhibits pronounced fluctuations across years, especially during 2020–2024, with larger amplitudes compared to earlier years. Alternating negative and positive values suggest that the series may have cyclical or seasonal characteristics.

(2.1) Stationarity Check

```
Warning message in adf.test(HWI_TIME):
“p-value smaller than printed p-value”

        Augmented Dickey-Fuller Test

data:  HWI_TIME
Dickey-Fuller = -5.5439, Lag order = 4, p-value = 0.01
alternative hypothesis: stationary

Warning message in kpss.test(HWI_TIME):
“p-value greater than printed p-value”

        KPSS Test for Level Stationarity

data:  HWI_TIME
KPSS Level = 0.095408, Truncation lag parameter = 3, p-value = 0.1
```

Figure 32. Check ADF & KPSS at D=0 in HWI data

To reassess the stationarity of the series, the Augmented Dickey-Fuller (ADF) and KPSS tests were applied:

ADF test: p-value = 0.01 < 0.05, indicating the series does not have a unit root and is stationary.

KPSS test: p-value = 0.1 > 0.05, indicating the series is stable around the mean.

The consistent results from both tests—ADF confirming removal of unit roots and KPSS validating stationarity—allow concluding that the series no longer contains a clear trend. To determine the degree of stationarity and examine temporal correlations, ACF analysis is conducted next.

(2.2) ACF Calculation (D=0)

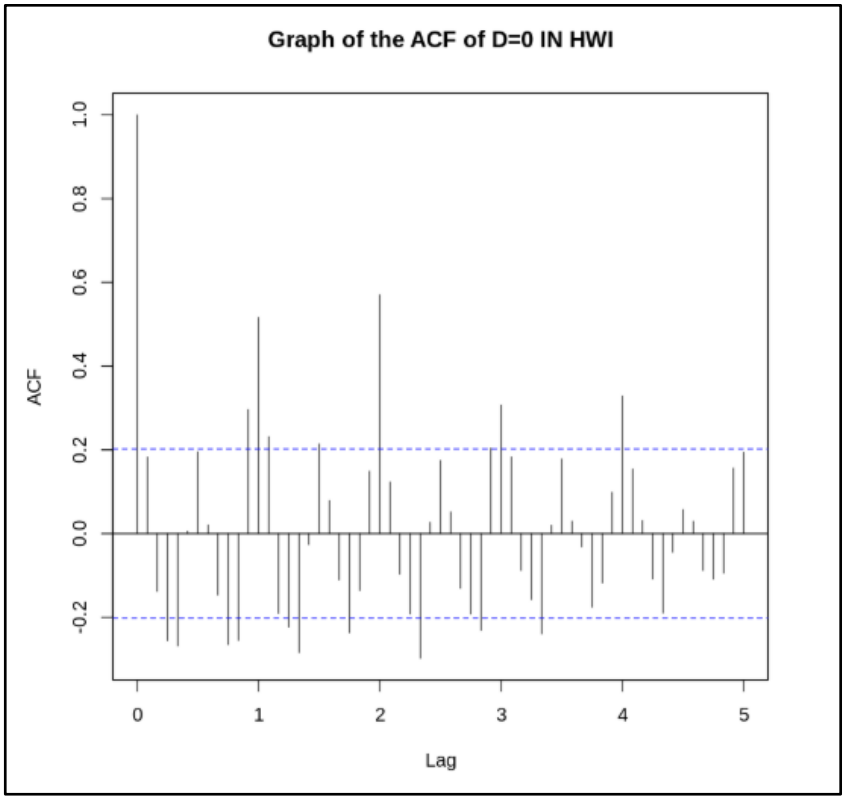

Figure 33. Calculate ACF at D=0 in HWI data

The ACF plot shows that autocorrelation coefficients at small lags (particularly lags 1, 2, 3, and 4) exceed the confidence interval (represented by two dashed blue lines), indicating that current values still have a strong linear relationship with recent past values.

Additionally, the ACF bars decrease slowly and oscillate around zero without fully cutting off in the initial lags, reflecting long-lasting autocorrelation. Hence, the series exhibits weak stationarity. The research team proceeds with first-order differencing on the Holt-Winters imputed series.

(2.3) First-order Differencing (D=1)

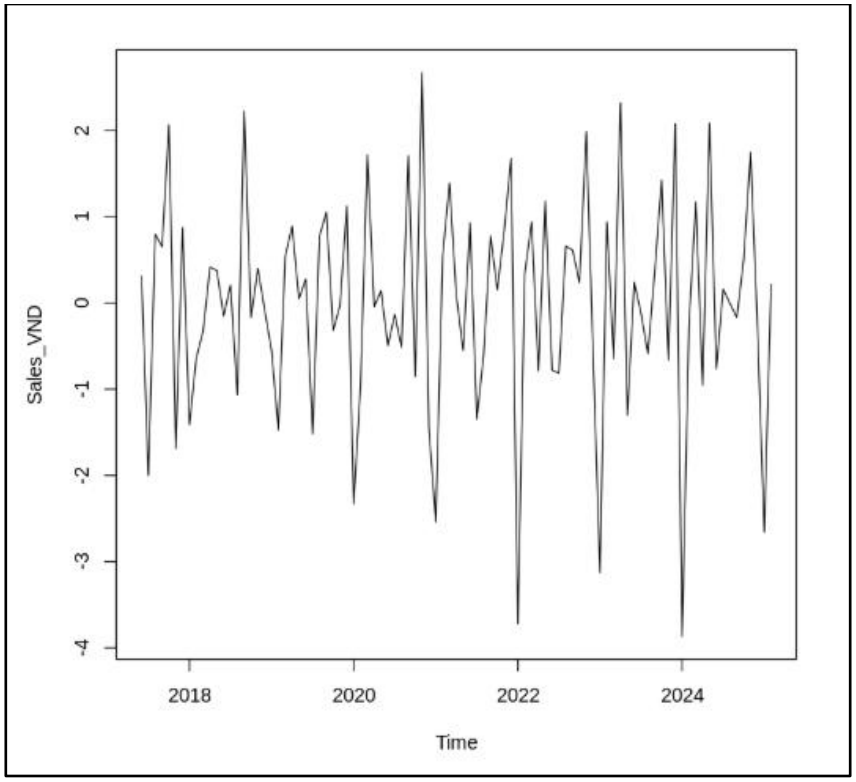

Figure 34. Visualize data after differecing at D=1 in HWI data

```
Warning message in adf.test(D1_HWI):
“p-value smaller than printed p-value”

    Augmented Dickey-Fuller Test

data:  D1_HWI
Dickey-Fuller = -9.2274, Lag order = 4, p-value = 0.01
alternative hypothesis: stationary

Warning message in kpss.test(D1_HWI):
“p-value greater than printed p-value”

    KPSS Test for Level Stationarity

data:  D1_HWI
KPSS Level = 0.018435, Truncation lag parameter = 3, p-value = 0.1
```

Figure 35. Check ADF & KPSS at D=1 in HWI data

The original HWI_TIME series shows strong fluctuations and unstable variance across periods. First-order differencing is applied to remove underlying trends and stabilize the series, yielding D1_HWI.

ADF test: p-value = 0.01 < 0.05, confirming stationarity.

KPSS test: p-value = 0.1 > 0.05, failing to reject stationarity around the mean.

Both tests consistently indicate that first-order differencing successfully stabilizes the series, making it suitable for further ACF and PACF analyses.

(2.4) ACF Calculation (D=1)

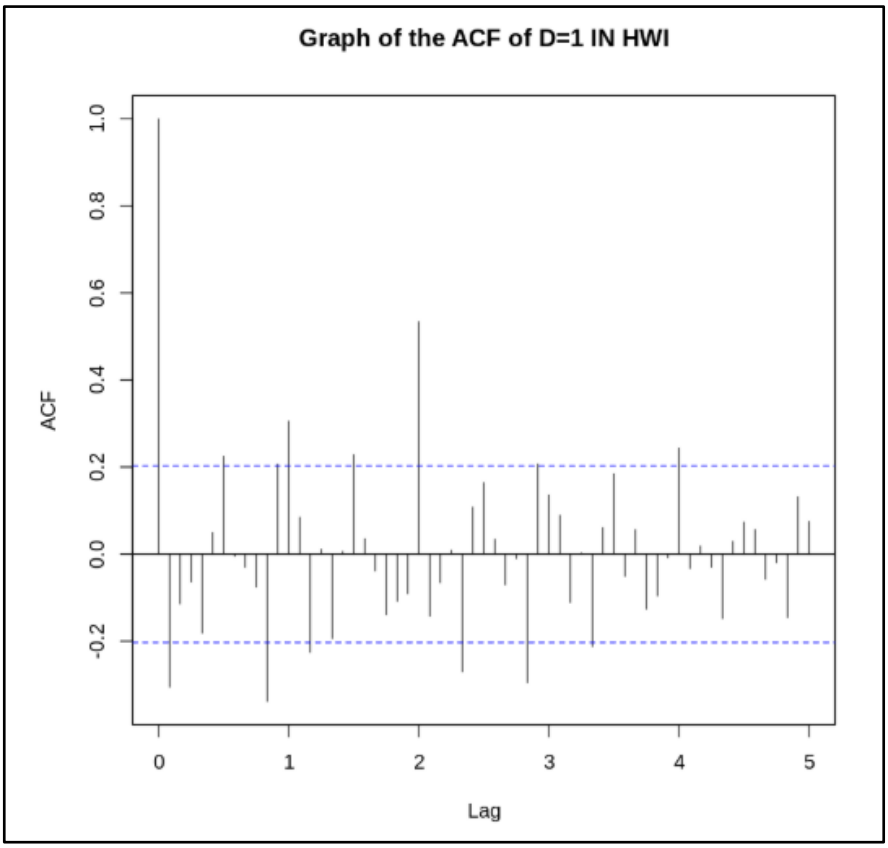

Figure 36. Calculate ACF at D=1 in HWI data

The ACF plot after first-order differencing shows significant autocorrelation at lags 1, 2, 3, and 4. However, due to data loss from differencing, lag 4 is excluded from consideration. Based on these results, models with Q = 1, 2, 3, 4 are considered.

(2.5) PACF Calculation (D=1)

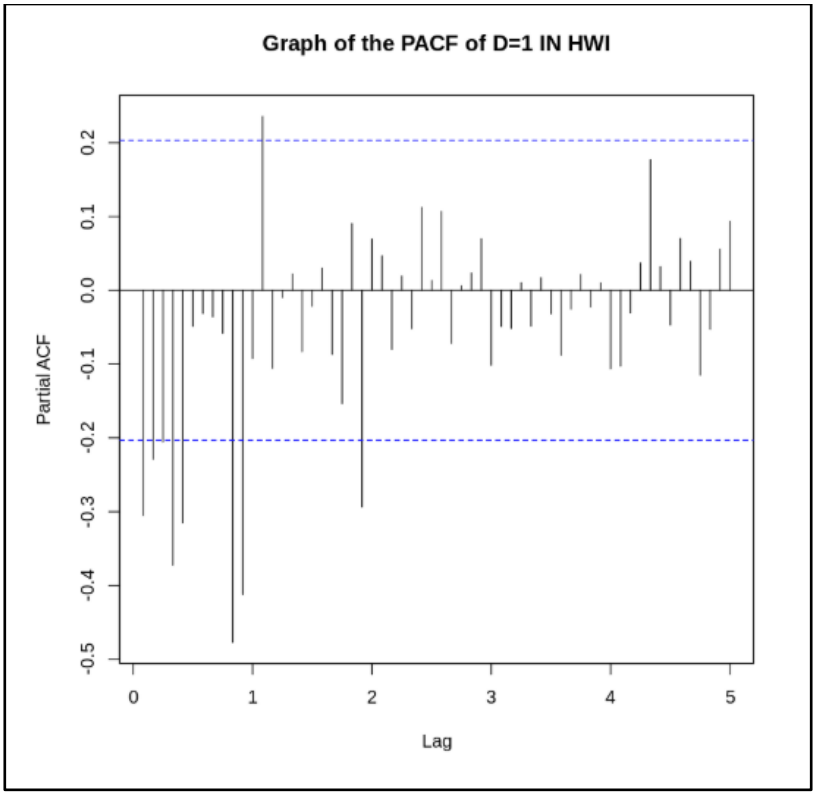

Figure 37. Calculate PACF at D=1 in HWI data

The PACF plot of the first-order differenced series shows that partial autocorrelation is significant only at lag 1, with all subsequent lags within confidence limits. This indicates that autoregressive influence exists only at the first lag, so P = 1, 2 is selected.

b. Seasonality Check and Deseasonalization

(1) LI DATA

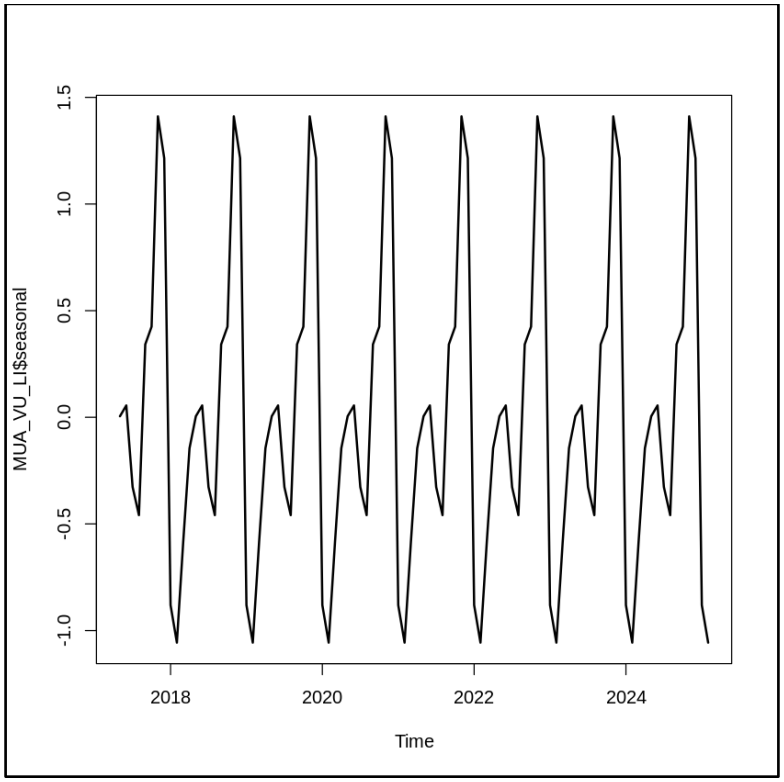

Figure 38. The seasonal of LI data

Seasonality Check: Observing the LI_DATA series (2017–2025) shows that fluctuations tend to repeat following a 12-month cycle. The increase–decrease patterns are similar across years—for example, values usually drop in mid-year and rise again toward year-end—demonstrating clear seasonal characteristics.

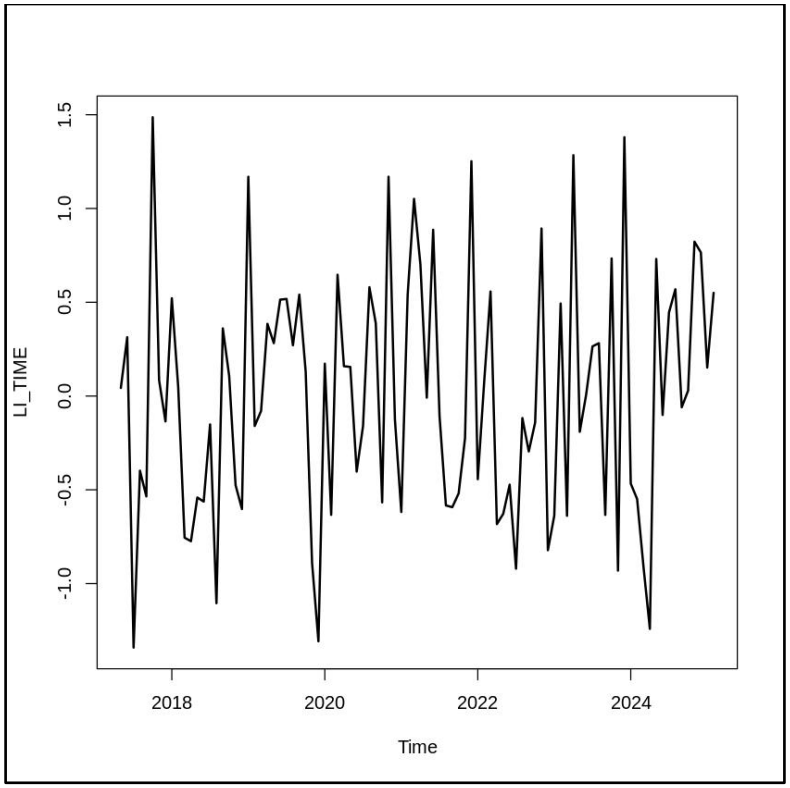

Figure 39. The data after deseasonalized

Deseasonalization: To remove seasonal effects, the series was differenced with a 12-month lag. The resulting series shows stabilized amplitudes, with no obvious repeating increase–decrease patterns. Positive and negative values now fluctuate randomly around the mean, confirming that the seasonal component has been effectively removed. After deseasonalization, LI_DATA becomes more structurally and variance-stable, ensuring stationarity for accurate SARIMA estimation and forecasting.

(1.1) Stationary check

```
Warning message in adf.test(KHUMUA_LI):
“p-value smaller than printed p-value”

        Augmented Dickey-Fuller Test

data:  KHUMUA_LI
Dickey-Fuller = -4.646, Lag order = 4, p-value = 0.01
alternative hypothesis: stationary

Warning message in kpss.test(KHUMUA_LI):
“p-value greater than printed p-value”

        KPSS Test for Level Stationarity

data:  KHUMUA_LI
KPSS Level = 0.10193, Truncation lag parameter = 3, p-value = 0.1
```

Figure 40. Check ADF & KPSS at d=0 in LI data

To reassess the stationarity of the series, two common tests, Augmented Dickey-Fuller (ADF) and KPSS, were applied.

ADF test: p-value = 0.01, smaller than the significance level (0.05), indicating the absence of a unit root. Therefore, the series is stationary.

KPSS test: p-value = 0.1, greater than 0.05, confirming that the series is stationary around its mean.

The consistent outcomes from both tests — ADF confirming no unit root and KPSS verifying level stationarity — indicate that the KHUMUA_LI series is stationary and no longer exhibits a significant trend component.

(1.2) ACF Calculation (d=0)

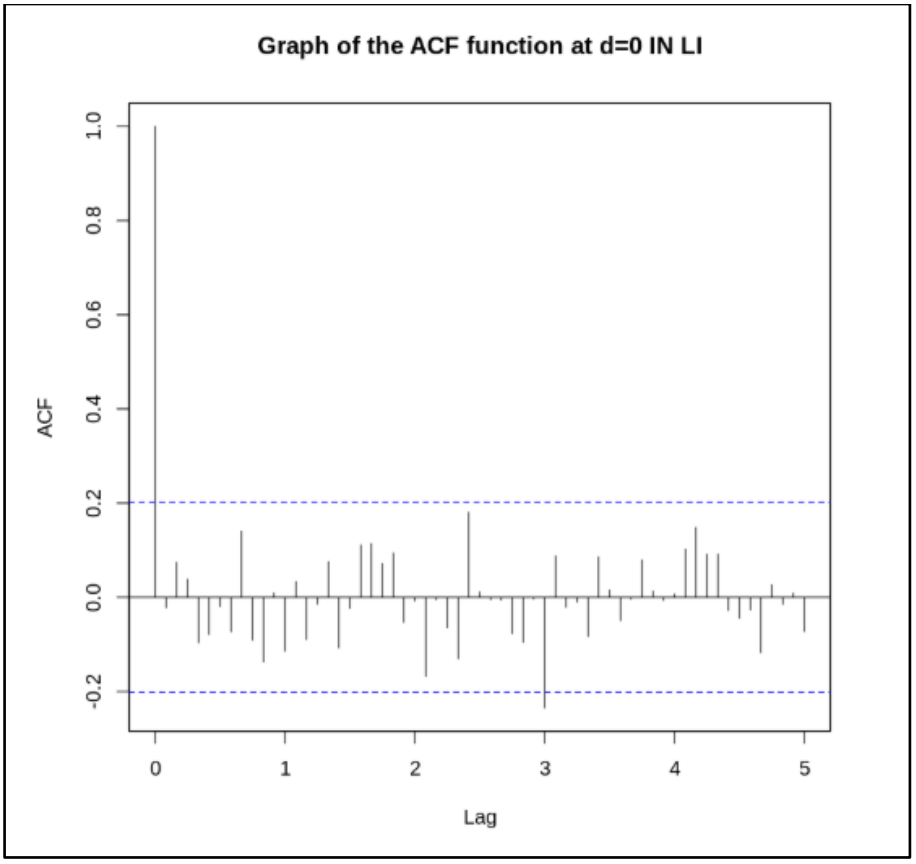

Figure 41. Calculate ACF at d=0 in LI data

The ACF plot of the deseasonalized series shows that autocorrelation coefficients at lag 3 exceed the confidence bounds. Based on this observation, the research team decides to select q=3.

(1.3) PACF Calculation (d=0)

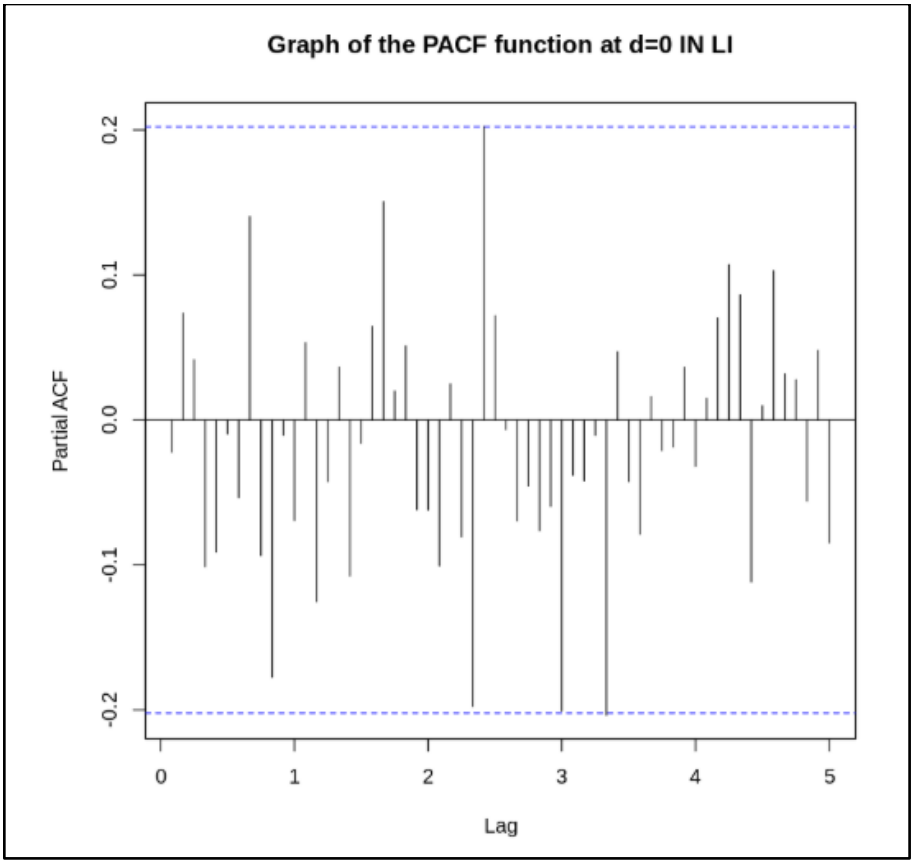

Figure 42. Calculate PACF at d=0 in LI data

The PACF plot of the same series indicates that partial autocorrelation exceeds the significance threshold at lag 4, while the remaining lags stay within the confidence limits. This suggests a strong autoregressive effect at the fourth lag, leading the team to choose p=4.

(2) HWI DATA

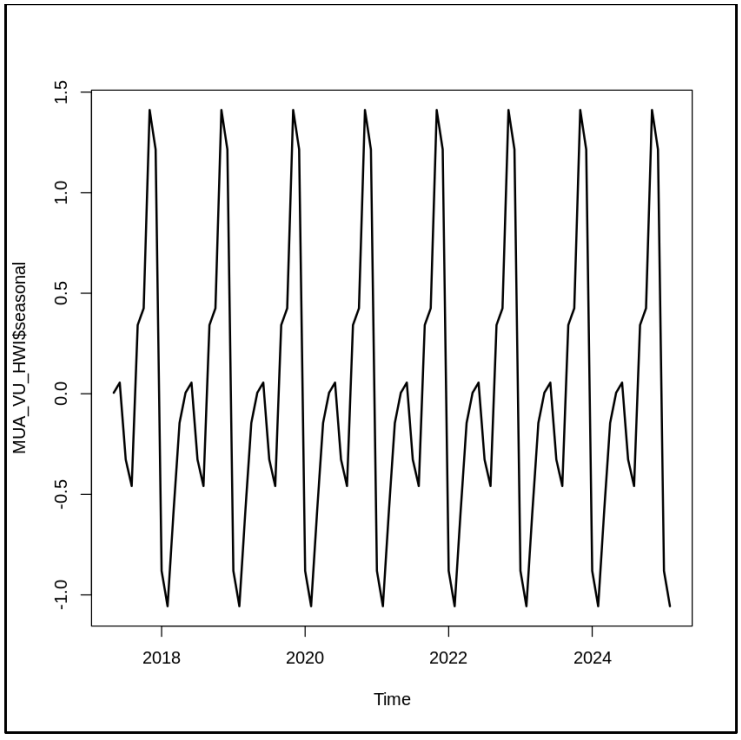

Figure 43. The seasonal of HWI data

Seasonality Check: Similarly, the original HWI_DATA series exhibits clear seasonality. Over the years, values tend to repeat in a predictable pattern: dropping in certain months and rising toward year-end. This indicates the series is influenced by cyclical factors, likely related to weather, demand, or seasonal economic conditions.

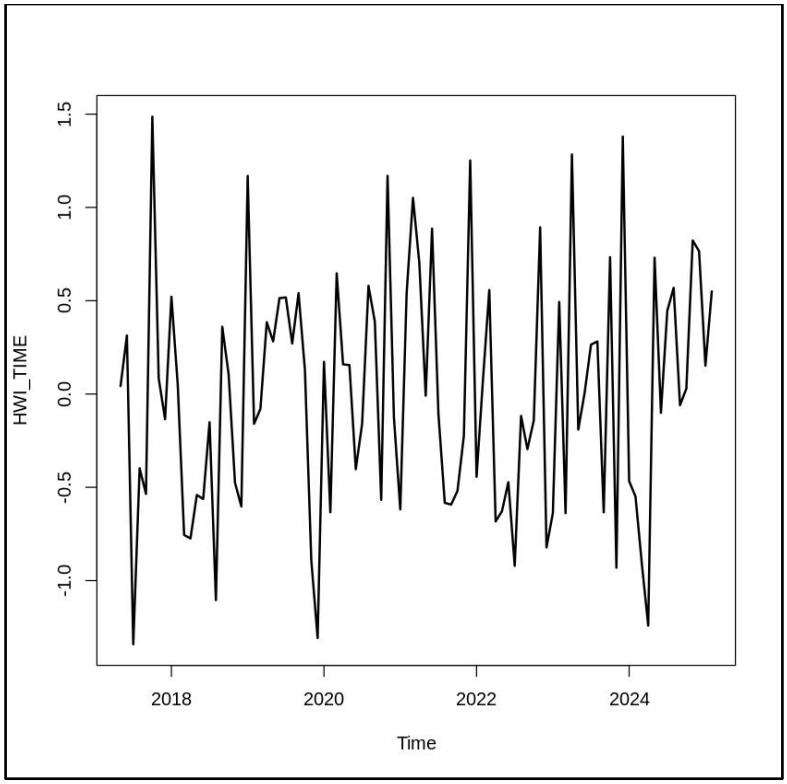

Figure 44. The HWI data after deseasonalized

Deseasonalization: After applying seasonal differencing (D=1), the HWI_DATA series shows a significant reduction in cyclical fluctuations. Amplitudes stabilize, and values no longer follow a recurring pattern, confirming that the seasonal component has been successfully removed. The deseasonalized HWI_DATA series becomes stationary and suitable for further ACF, PACF analyses, and SARIMA modeling, ensuring that the model reflects only trend and random components without annual seasonal noise.

(2.1) Stationary check

```
        Augmented Dickey-Fuller Test

data:  KHUMUA_HWI
Dickey-Fuller = -4.008, Lag order = 4, p-value = 0.01225
alternative hypothesis: stationary

Warning message in kpss.test(KHUMUA_HWI):
“p-value greater than printed p-value”

        KPSS Test for Level Stationarity

data:  KHUMUA_HWI
KPSS Level = 0.33252, Truncation lag parameter = 3, p-value = 0.1
```

Figure 45. Check ADF & KPSS at d=0 in HWI data

To reassess the stationarity of the series, two common tests, Augmented Dickey-Fuller (ADF) and KPSS, were applied. Results indicate:

ADF test: p-value = 0.01225, less than the significance level 0.05, indicating the series does not have a unit root and is stationary.

KPSS test: p-value = 0.1, greater than 0.05, indicating the series is stationary around the mean.

The consistent results from both tests—ADF confirming removal of unit roots and KPSS validating stationarity—allow the conclusion that the KHUMUA_HWI series is stationary and no longer exhibits a clear trend. However, to further understand the internal correlation pattern, autocorrelation function (ACF) and partial autocorrelation function (PACF) analyses were conducted.

(2.2) ACF Calculation (d=0)

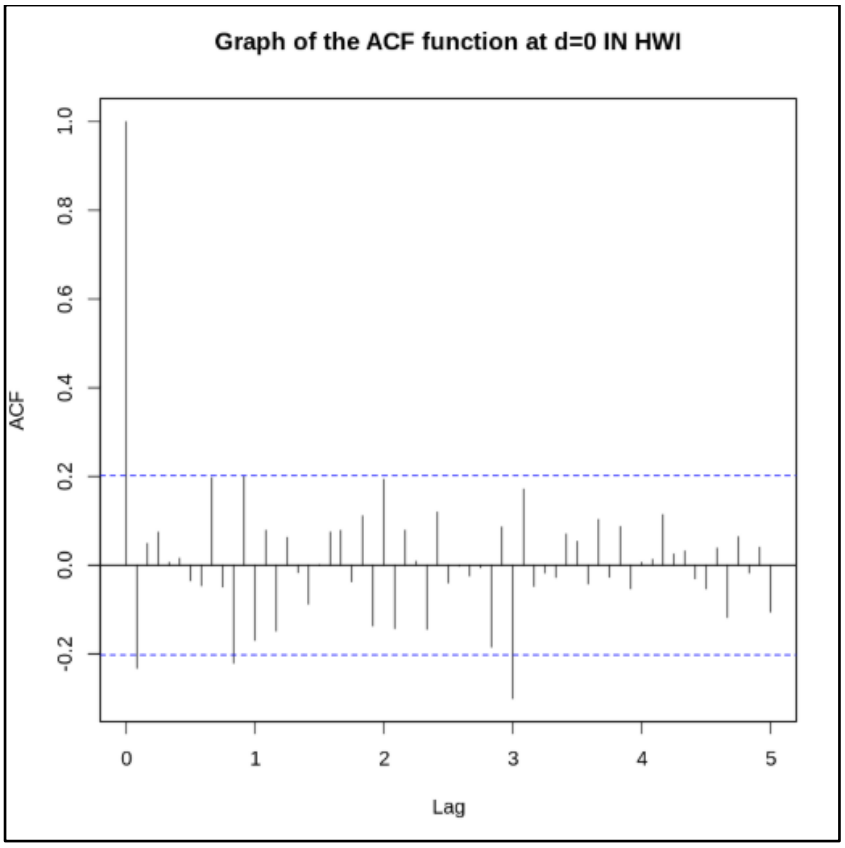

Figure 46. Calculate ACF at d=0 in HWI dat

The ACF plot of the deseasonalized series shows significant autocorrelation at lags 1 and 3, suggesting moving average components at those points. Therefore, the research team decides to choose q=1,3.

(2.3) PACF Calculation (d=0)

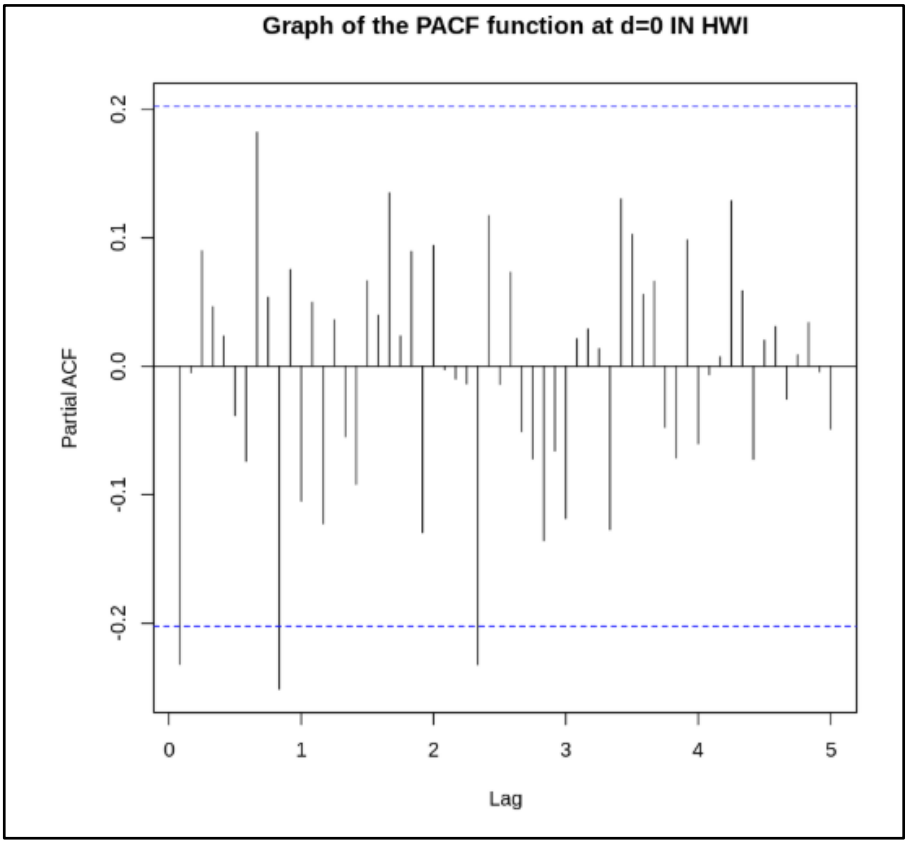

Figure 47. Calculate PACF at d=0 in HWI data

The PACF plot indicates that partial autocorrelation exceeds the significance threshold at lags 1 and 3, while the remaining lags fall within the confidence limits. This implies that autoregressive effects occur primarily at those lags, leading the team to select p=1,3.

2.1.4 SARIMA Model Construction

a. LI TIME

Model Building

After the original data were processed for missing values using the Linear Interpolation (LI) method and standardized to a Z-score scale, the process of determining the optimal SARIMA model was carried out. The analysis of the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots for both the non-seasonal and seasonal components (with a seasonal period of 12 months) provided important insights for selecting the appropriate model orders.

Based on the observed patterns from the ACF and PACF plots, a set of parameters was identified for model construction. Specifically, the SARIMA model structure is denoted as (p, d, q)(P, D, Q)[s], where s = 12 represents the seasonal period of the model:

For the non-seasonal component: the model is specified with orders p=4; d=0; q=3.

For the seasonal component: the model is specified with orders P=1,3; D=1; Q=1,2,3.

From these configurations, we constructed a total of six models as follows:

```
MH1<-Arima(LI_TIME,order = c(4,0,3),
  seasonal = list(order = c(1,1,1), period = 12),
  lambda ="auto", include.constant = FALSE)
MH2<-Arima(LI_TIME,order = c(4,0,3),
  seasonal = list(order = c(1,1,2), period = 12),
  lambda ="auto", include.constant = FALSE)
MH3<-Arima(LI_TIME,order = c(4,0,3),
  seasonal = list(order = c(1,1,3), period = 12),
  lambda ="auto", include.constant = FALSE)
MH4<-Arima(LI_TIME,order = c(4,0,3),
  seasonal = list(order = c(3,1,1), period = 12),
  lambda ="auto", include.constant = FALSE)
MH5<-Arima(LI_TIME,order = c(4,0,3),
  seasonal = list(order = c(3,1,2), period = 12),
  lambda ="auto", include.constant = FALSE)
MH6<-Arima(LI_TIME,order = c(4,0,3),
  seasonal = list(order = c(3,1,3), period = 12),
  lambda ="auto", include.constant = FALSE)
```

Figure 48. Model building of LI data

A data.frame: 6 × 2

	df	AIC
	<dbl>	<dbl>
MH1	10	201.5680
MH2	11	200.7935
MH3	12	198.9715
MH4	12	201.3156
MH5	13	203.2724
MH6	14	204.6986

Figure 49. AIC of models in LI data

A data.frame: 6 × 2

	df	BIC
	<dbl>	<dbl>
MH1	10	225.6352
MH2	11	227.2674
MH3	12	227.8521
MH4	12	230.1962
MH5	13	234.5598
MH6	14	238.3927

Figure 50. BIC of models in LI data

The selection of the optimal SARIMA model was based on a trade-off between model fit and complexity, evaluated using two criteria: the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC). The comparison of six potential models (MH1 to MH6) showed that the optimal choice depends on the evaluation criterion. Specifically, Model MH3 achieved the lowest AIC value (198.97), suggesting it offers the best balance between accuracy and simplicity for forecasting purposes.

However, according to the BIC criterion, Model MH1 was preferred, as it had the lowest BIC value (225.64). This discrepancy indicates that while MH3 fits the data better, it may be prone to overfitting when evaluated under BIC. To make a final decision, both models will undergo

Ljung–Box testing and residual diagnostics to determine which one provides the most reliable forecasting performance.

MH1 – SARIMA(4,0,3)(1,1,1)[12]

```
    Ljung-Box test

data:  Residuals from ARIMA(4,0,3)(1,1,1)[12]
Q* = 14.011, df = 11, p-value = 0.2324

Model df: 9.   Total lags used: 20
```

Figure 51. Ljung-Box test in the MH1 – LI data

For the first model, MH1, with parameters (4,0,3)(1,1,1)[12] corresponding to (p,d,q)(P,D,Q)[12], the Ljung–Box test was conducted to examine whether the residuals represent white noise and to assess the adequacy of the fitted ARIMA(4,0,3)(1,1,1)[12] model. The Ljung–Box test was applied to the residual series of the model. The test results showed a Q* statistic of 14.011 with 11 degrees of freedom (df = 11), yielding a p-value = 0.2324. Since this p-value is much greater than the 5% significance level, the null hypothesis (H_0: residuals are white noise) cannot be rejected. This indicates that there is no statistical evidence of autocorrelation in the residuals. Therefore, it can be concluded that the ARIMA(4,0,3)(1,1,1)[12] model successfully captures all predictable components from the data, and the remaining residuals behave as white noise — satisfying one of the key assumptions for a well-fitted model.

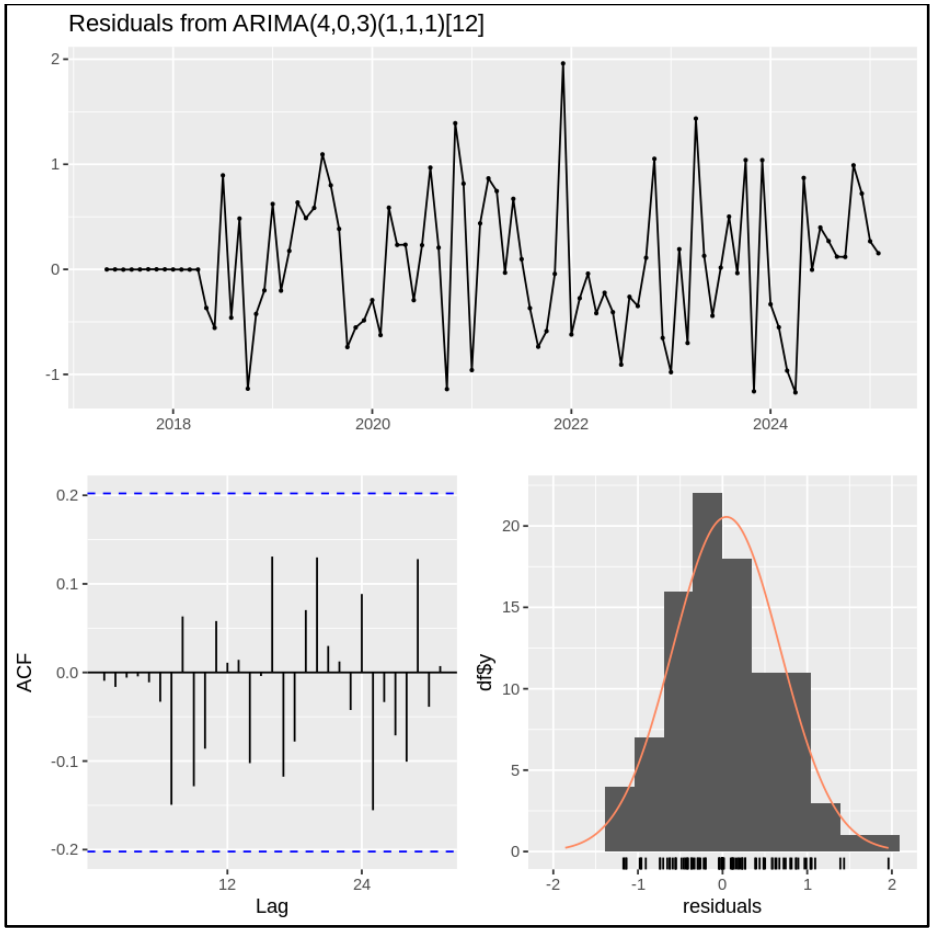

Figure 52. Check residuals in MH1

Residual analysis is a crucial step in assessing the adequacy of the ARIMA(4,0,3)(1,1,1)[12] model. Observation of the residual plot shows that the residual values are randomly dispersed around the zero line, without forming any noticeable trends or systematic patterns. This random distribution indicates that the model has effectively captured most of the systematic components in the data, and that the remaining residuals exhibit no signs of autocorrelation. This finding is consistent with the Ljung–Box test results (Q = 14.011, p-value = 0.2324), where the p-value exceeds the 5% significance level, confirming that the null hypothesis of "white noise residuals" cannot be rejected. Therefore, it can be concluded that the ARIMA(4,0,3)(1,1,1)[12] model is appropriately specified, satisfies the randomness assumption of the residuals, and can be reliably used for forecasting purposes.

In addition to randomness, testing the normality assumption of residuals is essential to ensure the reliability of model estimates and forecast intervals. Examination of the residual histogram, combined with the theoretical normal curve, reveals that the residuals of the ARIMA(4,0,3)(1,1,1)[12] model approximately follow a normal distribution. The Q–Q plot, which compares the empirical quantiles of the residuals with the theoretical quantiles of the normal distribution, further supports this observation, as most data points lie close to the reference line. Although minor deviations may appear at the tails, this is common in practice

and does not significantly affect model reliability. Hence, the normality assumption of the residuals is generally satisfied.

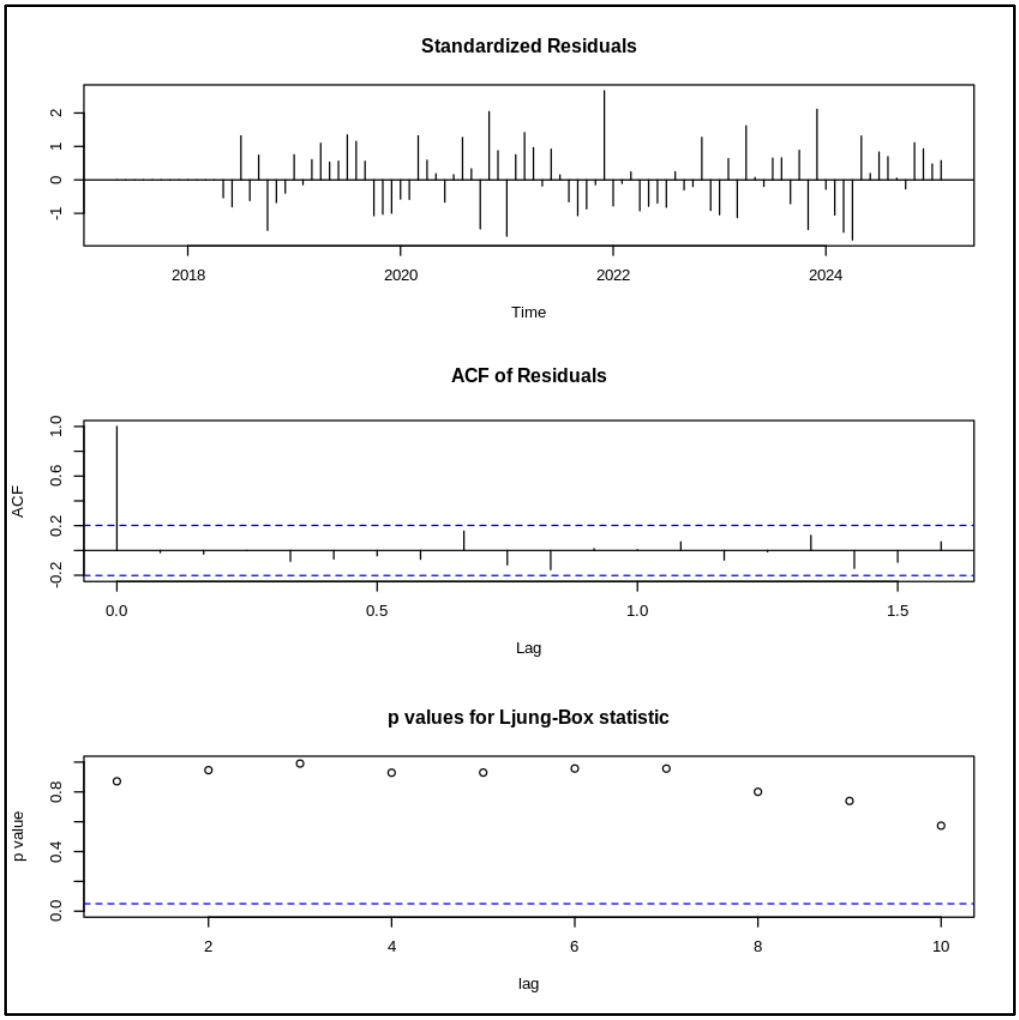

Figure 53. Tsdiag of MH1

Based on diagnostic results from the tsdiag tool for the ARIMA(4,0,3)(1,1,1)[12] model, the residuals are shown to meet key statistical assumptions. Specifically, the autocorrelation function (ACF) of the residuals indicates that all correlation coefficients fall within the confidence bounds, implying the absence of significant autocorrelation across lags. This outcome is consistent with the Ljung–Box test (Q = 14.011, p-value = 0.2324), where a p-value greater than 5% confirms that the residuals behave as white noise.

Furthermore, the residuals are approximately normally distributed, as shown by the density plot combined with the theoretical normal curve and the Q–Q plot, where most data points cluster near the diagonal line despite minor tail deviations — a common and acceptable phenomenon in real-world analyses. Overall, the diagnostics from tsdiag confirm that the ARIMA(4,0,3)(1,1,1)[12] model is well-specified, with residuals satisfying the assumptions of randomness, non-autocorrelation, and near-normal distribution, thereby ensuring the reliability of the forecasts generated by the model.

MH3 - ARIMA(4,0,3)(1,1,3)[12]

```
    Ljung-Box test

data:  Residuals from ARIMA(4,0,3)(1,1,3)[12]
Q* = 12.038, df = 9, p-value = 0.2112

Model df: 11.   Total lags used: 20
```

Figure 54. Ljung-Box test of MH3

For the second model, MH3, with parameters (4,0,3)(1,1,3)[12] corresponding to (p,d,q)(P,D,Q)[12], the Ljung–Box test was conducted to verify whether the residuals behave as white noise and to assess the adequacy of the ARIMA(4,0,3)(1,1,3)[12] model. The test was performed on the residual series of the model.

The results showed a Q* statistic of 12.038 with 9 degrees of freedom (df = 9), yielding a p-value = 0.2112. Since this p-value is well above the 5% significance level, the null hypothesis (H_0: residuals are white noise) cannot be rejected. This indicates no statistical evidence of autocorrelation in the residuals. Therefore, it can be concluded that the ARIMA(4,0,3)(1,1,3)[12] model has captured most of the predictable structure in the data, and the remaining residuals exhibit white-noise characteristics—satisfying one of the key assumptions of the model. However, to determine which model performs better in forecasting, further residual diagnostics were conducted.

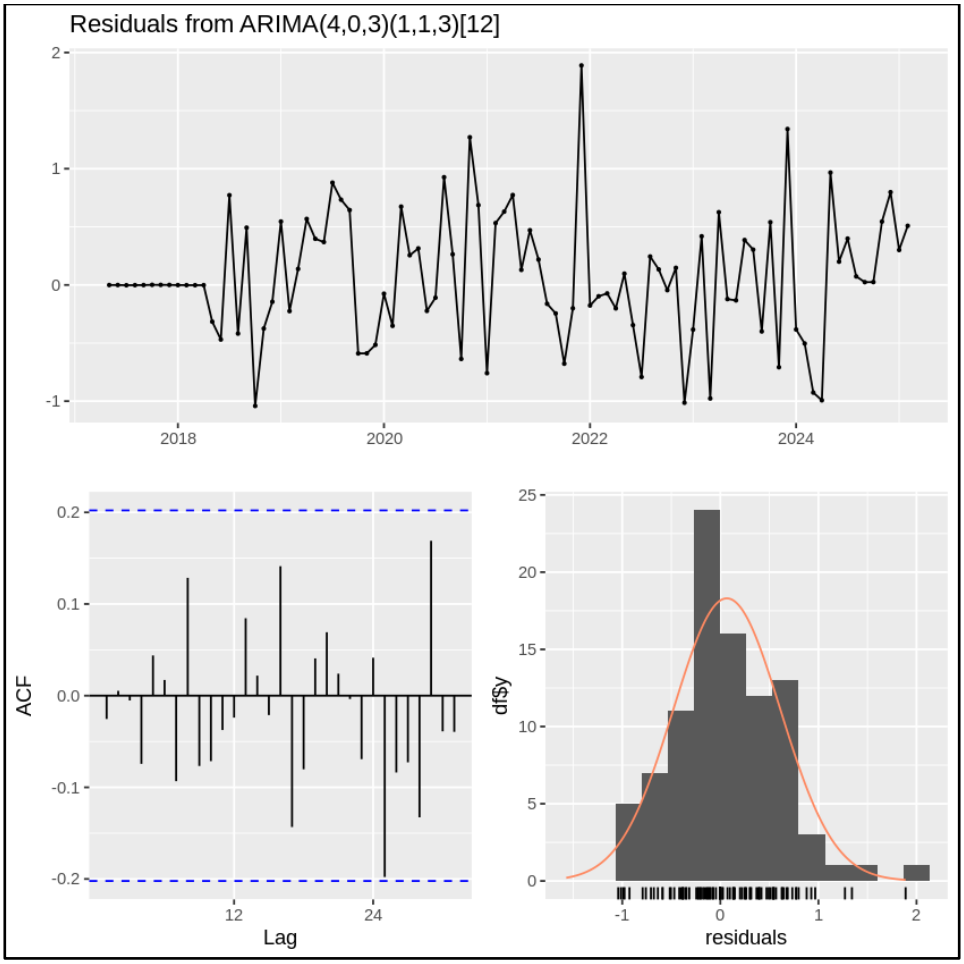

Figure 55. Check residuals of MH3

The residuals of the ARIMA(4,0,3)(1,1,3)[12] model exhibit a noticeably poorer approximation to normality compared to the previous model. Specifically, the histogram of residuals shows an asymmetric shape with extreme values that the density curve fails to capture accurately, although most residuals are still concentrated near zero. Compared to the previous model, the residual distribution here has longer tails and contains more outliers. These characteristics indicate that this model leaves more unmodeled "signals" within the residuals. Consequently, based on residual analysis criteria, the ARIMA(4,0,3)(1,1,3)[12] model demonstrates poorer goodness-of-fit and is not recommended for forecasting purposes.

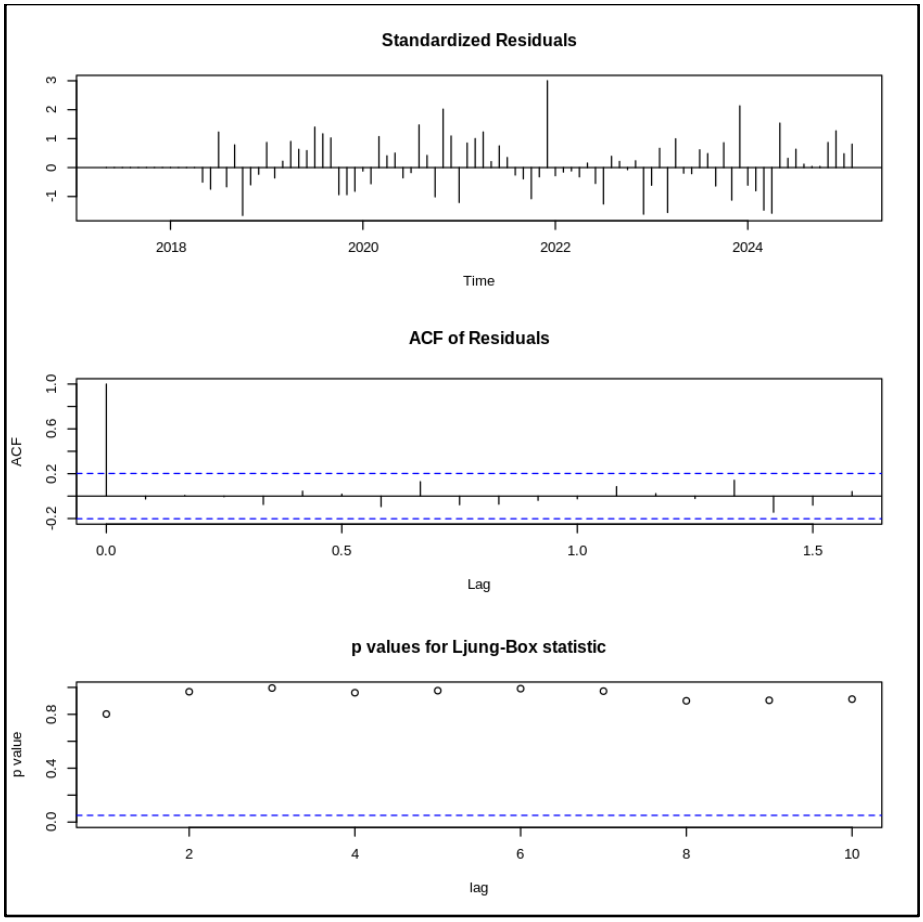

Figure 56. Tsdiag of MH3

According to the diagnostic results from the tsdiag tool, the residuals of the ARIMA(4,0,3)(1,1,3)[12] model generally satisfy the white-noise assumption, as all autocorrelation coefficients in the ACF plot fall within the confidence bounds and the Ljung–Box test yields p-values > 0.05 across all lags. However, compared with the previous model, this one presents clear limitations: the residuals deviate further from normality, contain more outliers, and exhibit clusters of varying variance—indicating remaining unmodeled patterns. Therefore, although it meets the basic white-noise requirement, the ARIMA(4,0,3)(1,1,3)[12] model shows weaker overall adequacy and is less suitable for forecasting than the preceding model.

Forecasting Results

As mentioned earlier, we decided to select Model 1 with parameters (4,0,3)(1,1,1)[12] to forecast six months corresponding to the six actual months we already have, in order to compare forecast errors and determine the model that best fits our dataset.

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Mar 2025	-0.68358842	-1.605992	0.3824297	-2.073464	0.8962556
Apr 2025	-0.39765397	-1.346012	0.6708697	-1.820584	1.1688883
May 2025	-0.33279397	-1.294790	0.7392090	-1.774195	1.2376953
Jun 2025	-0.03299584	-1.049915	1.0050036	-1.540263	1.4968613
Jul 2025	-0.18953553	-1.179875	0.8792111	-1.667689	1.3771925
Aug 2025	-0.11330497	-1.124225	0.9525631	-1.617908	1.4520279

Figure 57. Forecast of MH1

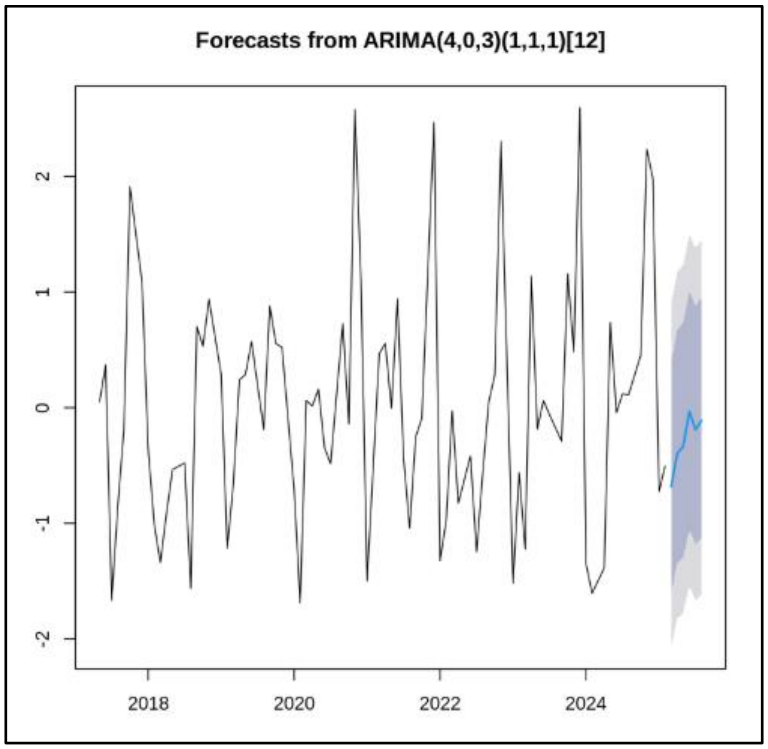

Figure 58. Visualize the MH1 forecast

The ARIMA(4,0,3)(1,1,1)[12] model was applied to generate forecasts for the six-month period from March to August 2025. It is important to note that all forecasted results are presented on the normalized (scaled) data scale. The point forecasts on this scale indicate a stable trend at a relatively low level, with all predicted values ranging from -0.684 to -0.033. Specifically, the forecast for March 2025 is the lowest (-0.684) and shows a gradual increase to -0.113 by August 2025.

The 80% and 95% confidence intervals were also calculated and remain within the normalized scale. A notable feature is that all these confidence intervals encompass the value 0, reflecting a considerable degree of uncertainty from the model. This implies that, on the current scale, the actual future values could potentially fall in the positive range.

After completing forecasting and preliminary evaluation based on the SARIMA model built from the dataset with missing values handled using Linear Interpolation (LI), the next step in our comparison process is to replicate the same procedure on the second dataset. Specifically,

we will construct and estimate a SARIMA model for the dataset in which missing values were treated using the Holt-Winters method. This step aims to assess the consistency of the forecasting model and, more importantly, to compare the forecasting performance between the two preprocessing methods. Through this, we seek to determine which method provides more reliable input data for the SARIMA model.

b. HWI TIME

Model Building

After the missing values were handled using the Holt-Winters method and the data were standardized with Z-score normalization, we proceeded to identify the optimal SARIMA model through ACF/PACF analysis. Based on this, a grid search was performed with the following sets of potential parameters:

For the non-seasonal component: p = [1, 3], d = 0, q = [1, 3].

For the seasonal component (period = 12): P = [1, 2], D = 1, Q = [1, 2, 3, 4].

A total of 32 models were constructed and estimated; however, during this process, 12 models were eliminated for failing to meet technical standards. The rejected models primarily had overly complex parameter configurations, leading to three main issues: first, the estimation process did not converge because the optimization algorithm failed to find a stable solution; second, some estimated parameters fell outside the stability or invertibility region, which undermined the practical interpretability of the model; and third, the Fisher information matrix could not be properly determined due to the large number of parameters, reducing the reliability of the estimates. As a result, 20 feasible models were retained for evaluation based on the AIC and BIC criteria in the final model selection step, ensuring both stability and reliability for forecasting.

```
HW1<-Arima(HWI_TIME,order = c(1,0,1),
  seasonal = list(order = c(1,1,1), period = 12),
  lambda ="auto", include.constant = FALSE)
HW2<-Arima(HWI_TIME,order = c(1,0,1),
  seasonal = list(order = c(1,1,2), period = 12),
  lambda ="auto", include.constant = FALSE)
HW3<-Arima(HWI_TIME,order = c(1,0,1),
  seasonal = list(order = c(1,1,3), period = 12),
  lambda ="auto", include.constant = FALSE)
HW4<-Arima(HWI_TIME,order = c(1,0,1),
  seasonal = list(order = c(1,1,4), period = 12),
  lambda ="auto", include.constant = FALSE)
HW5<-Arima(HWI_TIME,order = c(1,0,1),
  seasonal = list(order = c(2,1,1), period = 12),
  lambda ="auto", include.constant = FALSE)
HW6<-Arima(HWI_TIME,order = c(1,0,1),
  seasonal = list(order = c(2,1,2), period = 12),
  lambda ="auto", include.constant = FALSE)
HW7<-Arima(HWI_TIME,order = c(1,0,1),
  seasonal = list(order = c(2,1,3), period = 12),
  lambda ="auto", include.constant = FALSE)
HW8<-Arima(HWI_TIME,order = c(1,0,1),
  seasonal = list(order = c(2,1,4), period = 12),
  lambda ="auto", include.constant = FALSE)
HW9<-Arima(HWI_TIME,order = c(1,0,3),
  seasonal = list(order = c(1,1,1), period = 12),
  lambda ="auto", include.constant = FALSE)
```

Figure 59. Model building of HWI data

```
HW10<-Arima(HWI_TIME,order = c(1,0,3),
  seasonal = list(order = c(1,1,2), period = 12),
  lambda ="auto", include.constant = FALSE)
HW11<-Arima(HWI_TIME,order = c(1,0,3),
  seasonal = list(order = c(1,1,3), period = 12),
  lambda ="auto", include.constant = FALSE)
HW12<-Arima(HWI_TIME,order = c(1,0,3),
  seasonal = list(order = c(1,1,4), period = 12),
  lambda ="auto", include.constant = FALSE)
#HW13<-Arima(HWI_TIME,order = c(1,0,3),
#  seasonal = list(order = c(2,1,1), period = 12),
#  lambda ="auto", include.constant = FALSE)
#HW14<-Arima(HWI_TIME,order = c(1,0,3),
#  seasonal = list(order = c(2,1,2), period = 12),
#  lambda ="auto", include.constant = FALSE)
#HW15<-Arima(HWI_TIME,order = c(1,0,3),
#  seasonal = list(order = c(2,1,3), period = 12),
#  lambda ="auto", include.constant = FALSE)
#HW16<-Arima(HWI_TIME,order = c(1,0,3),
#  seasonal = list(order = c(2,1,4), period = 12),
#  lambda ="auto", include.constant = FALSE)
HW17<-Arima(HWI_TIME,order = c(3,0,1),
  seasonal = list(order = c(1,1,1), period = 12),
  lambda ="auto", include.constant = FALSE)
HW18<-Arima(HWI_TIME,order = c(3,0,1),
  seasonal = list(order = c(1,1,2), period = 12),
  lambda ="auto", include.constant = FALSE)
```

Figure 60. Model building of HWI data

```
HW19<-Arima(HWI_TIME,order = c(3,0,1),
  seasonal = list(order = c(1,1,3), period = 12),
  lambda ="auto", include.constant = FALSE)
HW20<-Arima(HWI_TIME,order = c(3,0,1),
  seasonal = list(order = c(1,1,4), period = 12),
  lambda ="auto", include.constant = FALSE)
#HW21<-Arima(HWI_TIME,order = c(3,0,1),
#  seasonal = list(order = c(2,1,1), period = 12),
#  lambda ="auto", include.constant = FALSE)
#HW22<-Arima(HWI_TIME,order = c(3,0,1),
#  seasonal = list(order = c(2,1,2), period = 12),
#  lambda ="auto", include.constant = FALSE)
#HW23<-Arima(HWI_TIME,order = c(3,0,1),
#  seasonal = list(order = c(2,1,3), period = 12),
#  lambda ="auto", include.constant = FALSE)
#HW24<-Arima(HWI_TIME,order = c(3,0,1),
#  seasonal = list(order = c(2,1,4), period = 12),
#  lambda ="auto", include.constant = FALSE)
HW25<-Arima(HWI_TIME,order = c(3,0,3),
  seasonal = list(order = c(1,1,1), period = 12),
  lambda ="auto", include.constant = FALSE)
```

Figure 61. Model building of HWI data

```
HW26<-Arima(HWI_TIME,order = c(3,0,3),
  seasonal = list(order = c(1,1,2), period = 12),
  lambda ="auto", include.constant = FALSE)
HW27<-Arima(HWI_TIME,order = c(3,0,3),
  seasonal = list(order = c(1,1,3), period = 12),
  lambda ="auto", include.constant = FALSE)
HW28<-Arima(HWI_TIME,order = c(3,0,3),
  seasonal = list(order = c(1,1,4), period = 12),
  lambda ="auto", include.constant = FALSE)
#HW29<-Arima(HWI_TIME,order = c(3,0,3),
#  seasonal = list(order = c(2,1,1), period = 12),
#  lambda ="auto", include.constant = FALSE)
#HW30<-Arima(HWI_TIME,order = c(3,0,3),
#  seasonal = list(order = c(2,1,2), period = 12),
#  lambda ="auto", include.constant = FALSE)
#HW31<-Arima(HWI_TIME,order = c(3,0,3),
#  seasonal = list(order = c(2,1,3), period = 12),
#  lambda ="auto", include.constant = FALSE)
#HW32<-Arima(HWI_TIME,order = c(3,0,3),
#  seasonal = list(order = c(2,1,4), period = 12),
#  lambda ="auto", include.constant = FALSE)
```

Figure 62. Model building of HWI data

After estimating 20 feasible models, the comparison results based on two information criteria, the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC), were summarized. The analysis revealed a clear consensus in selecting the optimal model. Specifically, model HW1 achieved both the lowest AIC value (170.33) and the lowest BIC value (182.36) among all evaluated models. The fact that HW1 simultaneously minimizes both criteria indicates that it offers an ideal balance between model complexity and data fit, while also demonstrating strong generalization capability for new data. The

agreement between AIC and BIC in identifying the same model (HW1) strongly reinforces the robustness of this selection. Therefore, HW1 was concluded to be the optimal SARIMA model for the dataset processed using the Holt-Winters method and was subsequently used for residual diagnostics and forecasting.

A data.frame: 20 × 2

	df	AIC
	<dbl>	<dbl>
HW1	5	170.3287
HW2	6	171.5360
HW3	7	171.0658
HW4	8	172.7788
HW5	6	172.1463
HW6	7	171.9909
HW7	8	173.0185
HW8	9	174.3972
HW9	7	173.8208
HW10	8	174.8631
HW11	9	173.2805
HW12	10	174.6266
HW17	7	173.7520
HW18	8	174.9037
HW19	9	174.5669

Figure 63. AIC of HWI models

HW20	10	175.9248
HW25	9	174.4158
HW26	10	175.1079
HW27	11	172.8686
HW28	12	175.9321

Figure 64. AIC of HWI models

A data.frame: 20 × 2

	df	BIC
	<dbl>	<dbl>
HW1	5	182.3623
HW2	6	185.9763
HW3	7	187.9129
HW4	8	192.0326
HW5	6	186.5866
HW6	7	188.8380
HW7	8	192.2723
HW8	9	196.0576
HW9	7	190.6679
HW10	8	194.1169
HW11	9	194.9410
HW12	10	198.6938
HW17	7	190.5990
HW18	8	194.1574
HW19	9	196.2274

Figure 65. BIC of HWI models

HW20	10	199.9920
HW25	9	196.0763
HW26	10	199.1751
HW27	11	199.3425
HW28	12	204.8128

Figure 66. BIC of HWI models

HW1 - SARIMA(4,0,3)(1,1,1)[12]

Based on comprehensive residual diagnostics, the SARIMA(1,0,1)(1,1,1)[12] (HW1) model was found to be well-aligned with the data structure. The Ljung-Box test indicated no evidence of autocorrelation in the residuals, with a p-value of 0.096, exceeding the 5% significance threshold and supporting the null hypothesis that the residuals are white noise. This finding is further supported by the autocorrelation function (ACF) plot, where most autocorrelation coefficients at different lags fall within the confidence bounds, confirming no systematic dependence among residuals.

Regarding the residual distribution, the histogram with an overlaid density curve showed that the residuals are approximately normally distributed, with only minor deviations. Additionally, the residual time plot displayed a random pattern fluctuating around the zero line without any visible trend or structure. However, it is worth noting that the Ljung-Box p-value, while above the significance threshold, is relatively close to 0.05, suggesting the presence of weak residual correlations. This implies that although the HW1 model fits well, there remains potential for further refinement. Overall, given the current diagnostic evidence, the HW1 model is considered appropriate and reliable for forecasting purposes.

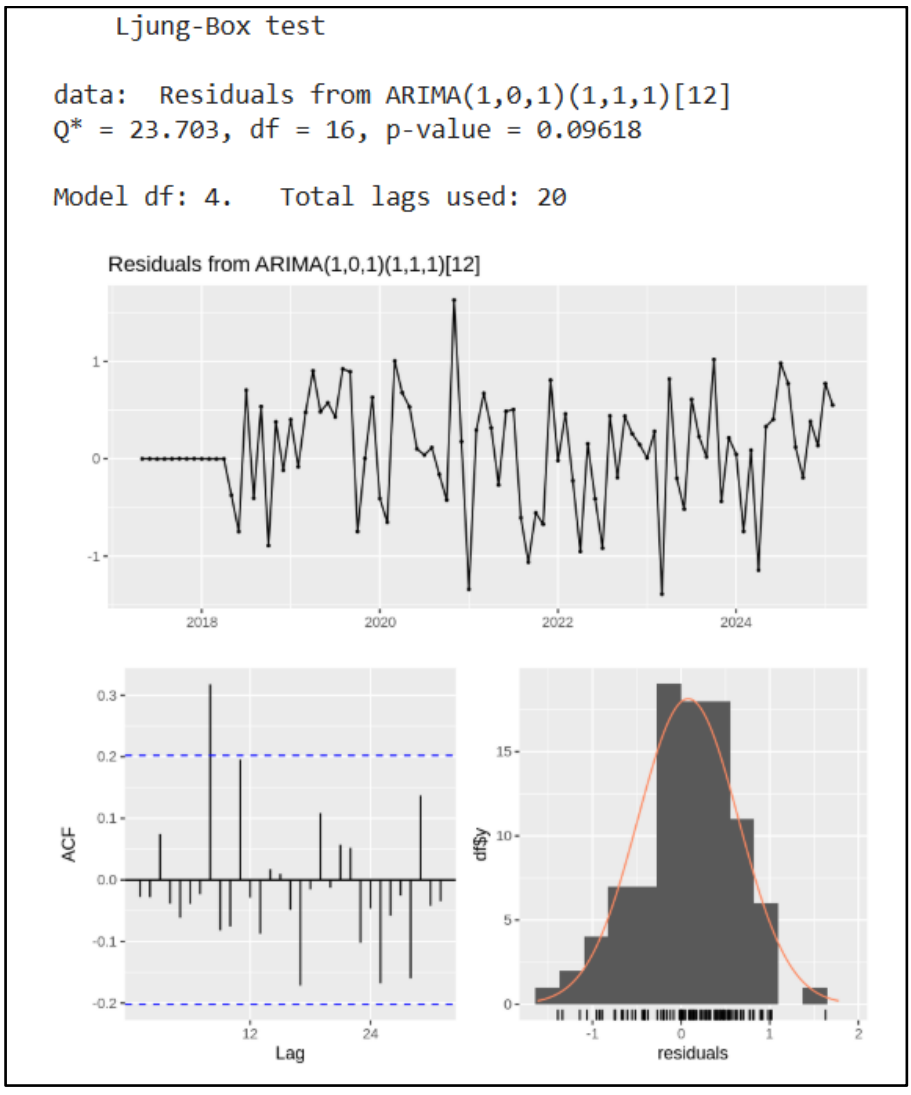

Figure 67. Check residuals of HW1

However, it is still necessary to perform a double-check using the tsdiag function to determine whether the model satisfies the statistical requirements for forecasting.

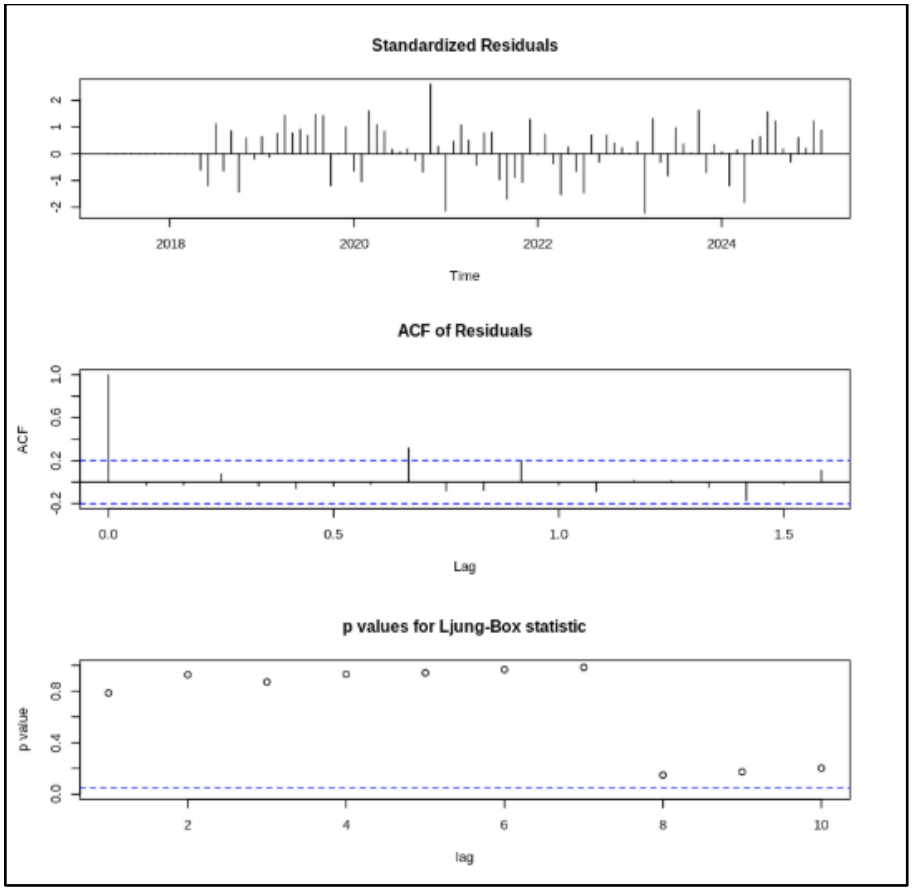

Figure 68. Tsdiag of HW1

To comprehensively validate the reliability of the HW1 model, we conducted a dual residual diagnostic using the tsdiag function. The analysis results revealed a high level of consistency across evaluation criteria. Specifically, the standardized residual plot demonstrated random fluctuations around zero, with most residuals remaining within the confidence bounds, reinforcing the assumptions of white noise and homoscedasticity. Notably, the residual autocorrelation function (ACF) further confirmed this finding, as nearly all autocorrelation coefficients fell within the confidence limits—indicating that the model effectively captured the autocorrelation structure of the original data.

However, the sequence of p-values from the Ljung-Box tests at different lags revealed a point worth attention. While most lags from 1 to 7 produced p-values exceeding the 0.05 significance threshold, a noticeable drop occurred at lags 8, 9, and 10, where the p-values approached the threshold (as indicated by the blue reference line). Although the overall Ljung-Box p-value (0.096) remains acceptable, this localized decline suggests that a weak autocorrelation structure may still persist at the higher lags, which the model has not fully captured. Nevertheless, from an overall perspective, the evidence from tsdiag confirms that the HW1 model satisfactorily meets the key statistical assumptions and demonstrates strong reliability for forecasting purposes, while still leaving minor room for further refinement in future improvements.

Forecasting Results

Based on the forecasting outcomes from model HW1 for the period from March to August 2025 using the normalized dataset, several notable characteristics regarding trend and seasonality can be observed. In terms of trend, the forecasted values fluctuate around the mean level without forming a clear and consistent upward or downward pattern. Specifically, after a significant decline in March, the forecasted values recover to the positive range from April to July before dropping back into the negative range in August. This indicates that the model predicts a relatively stable short-term state, with localized fluctuations around the baseline.

Regarding seasonality, the forecast clearly captures cyclical variations, as reflected by the substantial differences between adjacent months. The sharp transition from a deep negative value in March to positive values in April, and again from positive in July to negative in August, demonstrates that the model successfully identifies recurring seasonal patterns within the data — consistent with the seasonal component [12] [12] incorporated into the model. Although these forecasted values remain on the standardized scale, the structure of the forecast provides a clear view of the post-processed time series, where seasonal effects dominate over long-term trends.

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Mar 2025	-1.1116069	-1.9008167	-0.2963236	-2.312846	0.1710821
Apr 2025	0.3755009	-0.4969677	1.1965852	-0.932717	1.6204281
May 2025	0.1286334	-0.7350348	0.9643371	-1.165832	1.3913868
Jun 2025	0.1429611	-0.7216478	0.9775819	-1.152696	1.4044422
Jul 2025	0.2156262	-0.6528172	1.0453521	-1.085200	1.4712616
Aug 2025	-0.3650596	-1.1868616	0.5075293	-1.610948	0.9431447

Figure 69. Forecast of HW1

From the visualized chart combining historical and forecasted data, the SARIMA(1,0,1)(1,1,1)[12] (HW1) model built on data preprocessed with the Holt-Winters method shows superior performance in capturing and modeling seasonal dynamics. The forecast line (in blue) not only continues but also accurately replicates the characteristic 12-month cyclical pattern previously observed in the historical data. Specifically, the model successfully forecasts the early-year decline (March) followed by a mid-year recovery phase, fully aligning with the recurring seasonal patterns established during the years 2018, 2020, 2021, and 2023.

This forecasting ability directly stems from the Holt-Winters preprocessing, which effectively preserved the intrinsic seasonal structure of the data—providing a strong foundation for the SARIMA model with its seasonal component (1,1,1)[12] to operate efficiently. The seasonal differencing term (D=1) helps stabilize the series, while the seasonal autoregressive (P=1) and moving average (Q=1) components enable the model to capture

cross-year correlations effectively. As a result, the forecast line maintains its characteristic "wave-like" pattern instead of flattening out, confirming the model's capacity to forecast cyclic fluctuations. Additionally, the forecast confidence intervals widen gradually over time in a reasonable manner, accurately reflecting the increasing uncertainty associated with longer-term forecasts and demonstrating the model's reliable estimation of forecast error variability.

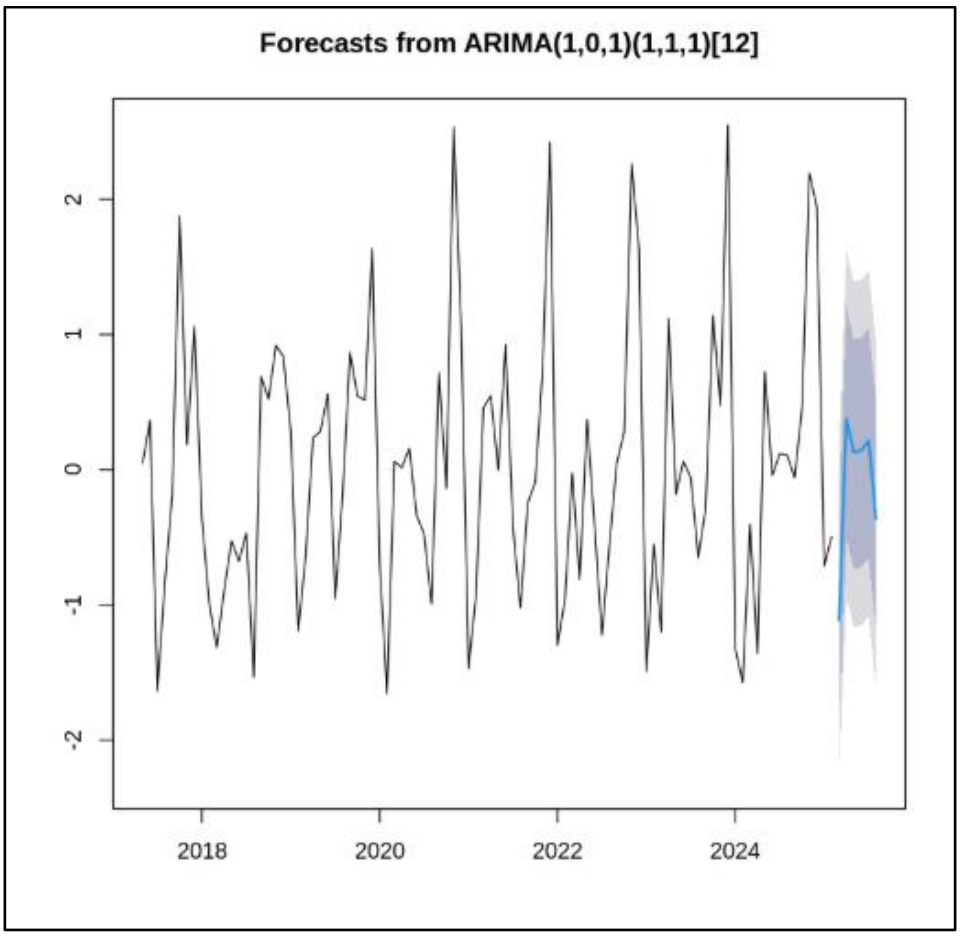

Figure 70. Visualize of HW1

The next crucial step in model evaluation is to reverse the normalization (unscaling) of all forecasted point values and confidence intervals to restore them to their original measurement units. Only after this transformation can the forecasted values be meaningfully compared with the actual observed data. The main objective of generating these short-term forecasts is to support an objective performance evaluation. By comparing the unscaled forecasted values with their corresponding actual observations, key error metrics such as MAE, RMSE, and MAPE can be accurately computed. The results of this comparison will serve as a critical basis for identifying the most optimal forecasting model, which can then be confidently applied for long-term forecasting.

2.2 Results

Reinstate initial value:

```
# Unscale the forecasted values from MH3 (LI_TIME)
# Need the original mean and standard deviation from when LI_TIME was scaled
mean_li = attr(LI_scaled, 'scaled:center')
sd_li = attr(LI_scaled, 'scaled:scale')
forecast_li_original_scale = pre_MH1$mean * sd_li + mean_li

# Unscale the forecasted values from HW1 (HWI_TIME)
# Need the original mean and standard deviation from when HWI_TIME was scaled
mean_hwi = attr(holt_winters_imputed_scaled, 'scaled:center')
sd_hwi = attr(holt_winters_imputed_scaled, 'scaled:scale')
forecast_hwi_original_scale = pre_HW1$mean * sd_hwi + mean_hwi

# Print the unscaled forecasts
print("Unscaled Forecasts (Linear Interpolation):")
print(forecast_li_original_scale)
print("Unscaled Forecasts (Holt-Winters Based Imputation):")
print(forecast_hwi_original_scale)
```

```
[1] "Unscaled Forecasts (Linear Interpolation):"
           Mar       Apr       May       Jun       Jul       Aug
2025 143001507 151196176 153055016 161647008 157160697 159345408
[1] "Unscaled Forecasts (Holt-Winters Based Imputation):"
           Mar       Apr       May       Jun       Jul       Aug
2025 130103418 173517020 166310141 166728412 168849748 151897607
```

Figure 71. Unscale the forecast

The code snippet is used to unscale the forecasted values from two different models - Linear Interpolation (LI_TIME) and Holt-Winters Based Imputation (HWI_TIME) - back to the original scale of the ceramic revenue data. The team performed the transformation to return the forecasting results to the original units (e.g., million VND) so they can be interpreted, analyzed, and compared with actual revenue figures.

The unscaling process is based on the inverse formula of z-score standardization:

$$\text{Original Value} = (\text{Scaled Value} \times \text{SD}) + \text{Mean}$$

Where:

Mean and SD (Standard Deviation) are the parameters that were initially used during standardization with the scale() function.

These two values are re-extracted from the scaled data object using the attr() command:

```
attr(dataset_scaled, 'scaled:center')  # Mean
attr(dataset_scaled, 'scaled:scale')   # SD
```

Figure 72. The algorithms

The team performs the unscaling according to the following steps:

Retrieve the Mean and Standard Deviation (SD) of the standardized data from the two datasets:

LI_scaled (corresponding to the Linear Interpolation model).

holt_winters_imputed_scaled (corresponding to the Holt-Winters model).

Apply the unscaling formula to each model:

Multiply the forecasted value (which is in standardized form) by the corresponding Standard Deviation.

Add the corresponding Mean value to convert it back to the original (true) value.

Store the results in two new variables:

forecast_li_original_scale for the Linear Interpolation model.

forecast_hwi_original_scale for the Holt-Winters model.

Print the unscaled forecast results to the screen to compare the two methods.

Both models returned unscaled forecast values, which are expressed in the original unit of measurement. The Linear Interpolation method yielded results with greater volatility between months, as linear interpolation simply connects adjacent data points. In contrast, the Holt-Winters Based Imputation produced a smoother sequence of values, which better reflects the data's inherent seasonality and trend—a hallmark feature of the Holt-Winters model. Crucially, all values are at a reasonable level and can be used for further analysis (e.g., comparison with actual data or plotting the forecast).

Compare forecast results:

```r
# Create a data frame to compare actual and forecast values
comparison_values_df <- data.frame(
  Datetime = seq(as.Date("2025-03-01"), by = "month", length.out = length(actual_data$Sales_VND)), # Use a date sequence starting from March 2025
  Actual_Values = as.numeric(actual_data$Sales_VND), # Convert time series to numeric vector
  Forecast_LI = as.numeric(forecast_li_original_scale), # Convert time series to numeric vector
  Forecast_HWI = as.numeric(forecast_hwi_original_scale) # Convert time series to numeric vector
)

# Print the comparison table
print("Comparison of Actual and Forecast Values (Starting March 2025):")
print(comparison_values_df)
```

```
[1] "Comparison of Actual and Forecast Values (Starting March 2025):"
    Datetime Actual_Values Forecast_LI Forecast_HWI
1 2025-03-01     145399492   143001507    130103418
2 2025-04-01     157758923   151196176    173517020
3 2025-05-01     176006666   153055016    166310141
4 2025-06-01     182082611   161647008    166728412
5 2025-07-01     205509243   157160697    168849748
6 2025-08-01     236514347   159345408    151897607
```

Figure 73. Comparison of actual and forecast values

The team implemented this code snippet with the aim of comparing the actual ceramic revenue values with the forecasted values. This comparison serves to assess the accuracy and reliability of the previously built forecasting models.The analysis is conducted for the period from March 2025 to August 2025, which falls within the forecasting validation phase after the models were trained using ceramic revenue data spanning from May 2017 to February 2025. Selecting this period helps to test the models' ability to accurately reflect near-future revenue trends.The primary method used in this section is a direct comparison between forecasted and actual values. The team created a data table encompassing the time series, actual revenue, and two columns for the forecasted values from the two distinct models: LI (Linear Interpolation) and HWI (Holt-Winters Based Imputation). The forecasted data was converted to a numerical format to ensure accurate comparison and analysis. This section does not delve into the model building process but focuses on evaluating the effectiveness and adherence to reality of the forecasting results.The implementation steps were carried out concisely. First, the team created the time series starting from March 2025, corresponding to the required comparison period. Subsequently, the actual revenue data and the forecasted values from the two models (Forecast_LI and Forecast_HWI) were extracted and converted to the same numerical format. Finally, the team generated a summary table containing these values and printed the result for easy observation, analysis, and direct visual comparison between the actual and forecasted figures.The comparison table results indicate that during the initial months of the forecast period (from March to May 2025), the forecasted values from both models adhered quite closely to the actual revenue, demonstrating that the models successfully learned the stable growth trend of ceramic revenue. However, starting from June 2025, a slight increase in the discrepancy can be observed. Specifically, the HWI model tended to under-forecast compared to the actual figures in some months, while the LI model maintained a relatively stable fit. Thus, it can be concluded that both models demonstrate relatively accurate short-term forecasting ability, but the LI model appears more suitable and reliable for forecasting the subsequent period.

```
[1] "Comparison of Forecast Error Metrics:"
                                  Method     RMSE      MAE     MAPE
1          Linear Interpolation Forecast 39340054 29644245 14.37108
2 Holt-Winters Based Imputation Forecast 39403834 29563522 14.67757
```

Figure 74. The error metrics of forecast and actual values

The error calculation results indicate that both models have quite similar levels of error; however, the Linear Interpolation Forecast model yields slightly more accurate results compared to the Holt-Winters Based Imputation Forecast model. Specifically, Linear Interpolation has an RMSE (Root Mean Square Error) of 39,340,054, an MAE (Mean Absolute Error) of 29,644,245, and a MAPE (Mean Absolute Percentage Error) of 14.37%. In contrast, Holt-Winters has an RMSE of 39,403,834, an MAE of 29,563,522, and a MAPE of 14.68%.

This suggests that the Linear Interpolation model tracks the actual data slightly better, with a smaller average error across all three evaluation metrics. However, the difference between the two models is not substantial, indicating that both models possess relatively stable and reliable forecasting capabilities. In this context, Linear Interpolation is deemed the more suitable model to use for forecasting ceramic revenue in the subsequent period.

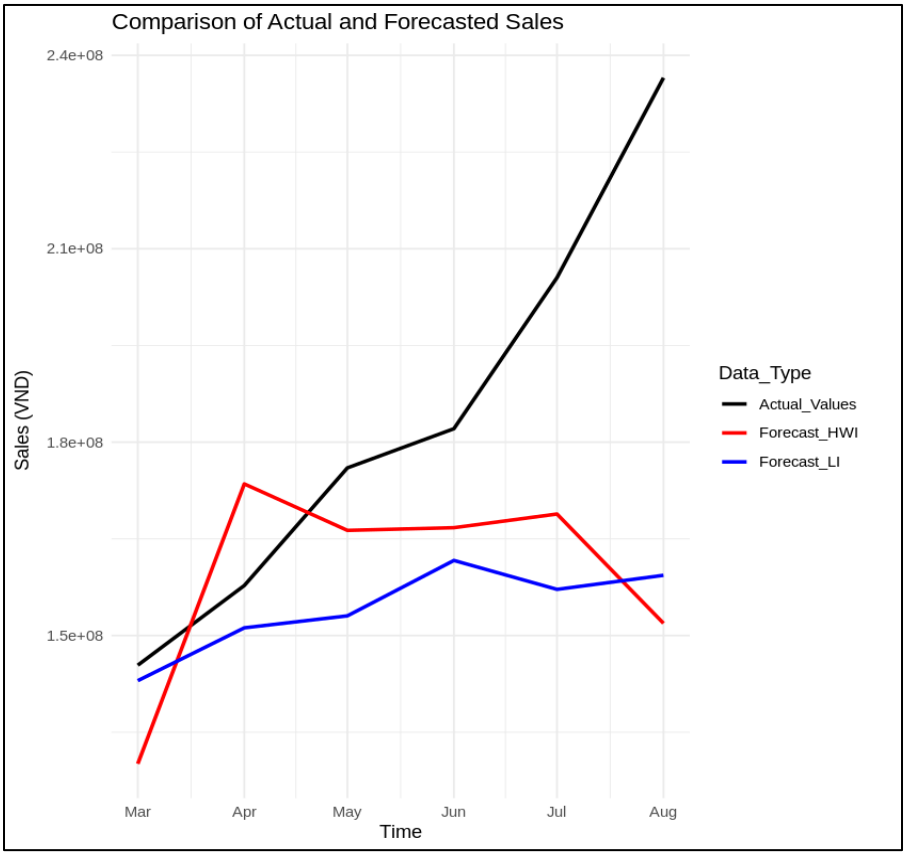

Figure 75. Visualize the forecast with actual

The chart above illustrates a comparison between the actual revenue (black line) and the forecasted values from two models: Linear Interpolation (blue line) and Holt-Winters Based Imputation (red line) for the period from March 2025 to August 2025.

Observations from the chart indicate that the actual revenue increased sharply and continuously month-over-month, especially from May to August 2025. Meanwhile, both forecasting models show a tendency to be lower than the actual figures, reflecting a more conservative forecasting approach.

The Linear Interpolation model (blue) yields results that are lower than the actual revenue but maintains a stable upward trend. In contrast, the Holt-Winters model (red) exhibits slight fluctuations and shows a greater deviation at certain points in time.

When compared with the quantitative error metrics (RMSE, MAE, MAPE), the results show that the Linear Interpolation model has slightly smaller error values than Holt-Winters (RMSE = 39,340,054; MAE = 29,644,245; MAPE = 14.37%). Holt-Winters, on the other hand, has an RMSE of 39,403,834; an MAE of 29,563,522; and a MAPE of 14.68%. This confirms that Linear Interpolation possesses higher numerical accuracy.

Synthesizing both the visual results and the error metrics, it can be concluded that Linear Interpolation is the more suitable model for forecasting ceramic revenue in the 2025 period. This model not only achieves better accuracy but also demonstrates stability and closely adheres to the actual trend, providing reliable forecast results for future business decisions.

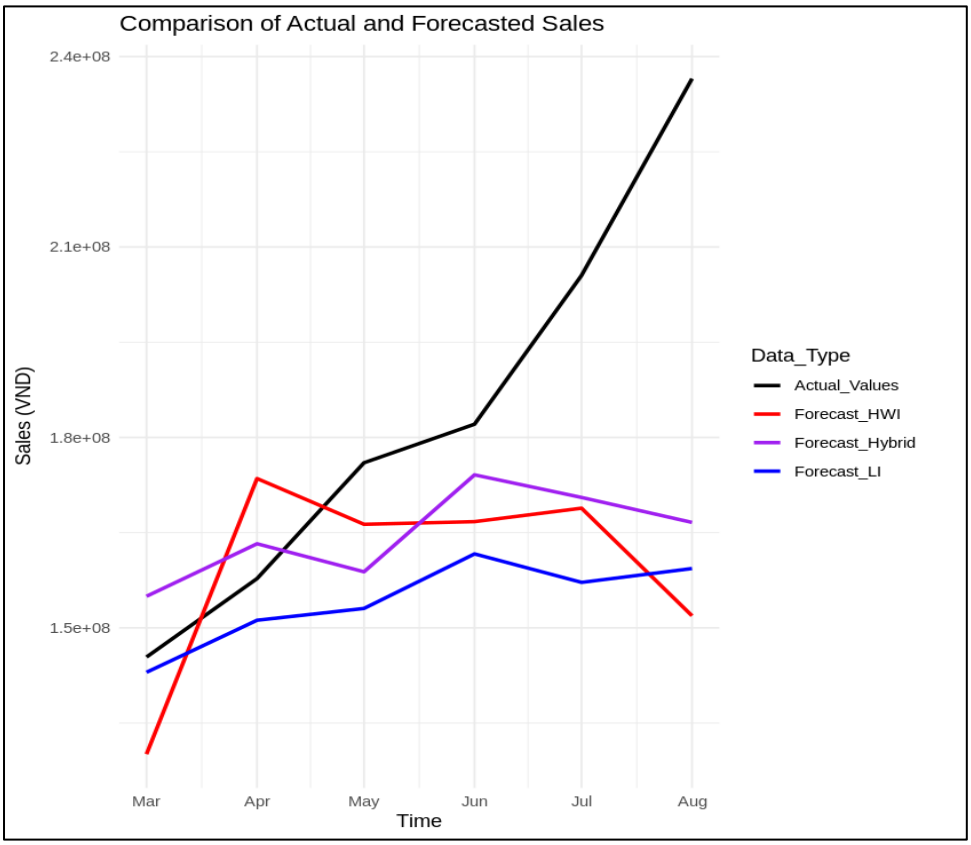

Figure 76. Compare the forecast with actual (include hybrid)

The chart above illustrates a comparison between the actual revenue (black line) and the forecasted values from three models: Linear Interpolation (LI – blue), Holt-Winters Imputation (HWI – red), and Hybrid (purple) for the period from March 2025 to August 2025.Observations from the chart reveal that actual revenue increased sharply and continuously during this period, particularly from May onwards. Both the LI and HWI models tend to forecast significantly lower than the actual figures, demonstrating their limited ability to react to the rapid rate of revenue increase.Conversely, the Hybrid model (purple line) yields results that adhere more closely to the actual revenue line, which is particularly evident between May and July. During this period, the model successfully reflects the stable upward trend and reduces the deviation from the true values.In terms of stability and trend capturing ability, the Hybrid model performs superiorly to the other two models. Its forecast line not only tracks closer to the actual data but also minimizes fluctuations and errors, effectively reflecting the growth characteristics of the ceramic revenue data.Conclusion:The Hybrid model demonstrates the best forecasting efficiency among the three models compared. By combining the strengths of SARIMA (capturing trend and seasonality) and XGBoost (effective non-linear learning and residual handling), the Hybrid model delivers results that are closer to the actual values, proving it to be the optimal choice for forecasting ceramic revenue in the 2025 period and subsequent cycles.

Forecast next 12 months:

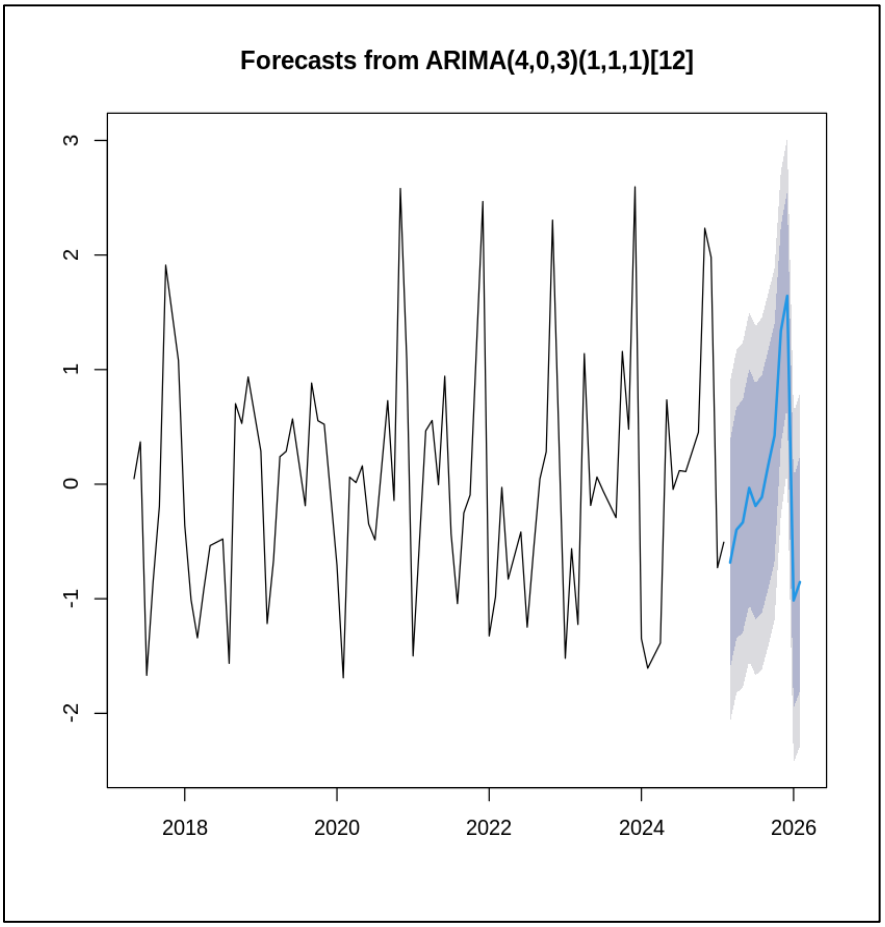

Figure 77. The HWI forecast for 12 months

The chart above illustrates the 12-month forecast results of the SARIMA(4,0,3)(1,1,1)[12] model for ceramic revenue.The black line represents historical data from 2017 to early 2025, while the blue line shows the forecasted values for the 2025–2026 period, accompanied by a grey band indicating the model's confidence interval.Observing the chart reveals that the SARIMA(4,0,3)(1,1,1)[12] model forecasts a clear upward trend in the initial months of the 2025–2026 period, demonstrating its ability to reflect the seasonal cycles and characteristic fluctuations of revenue from previous years. The forecast line gradually increases before showing signs of a slight adjustment towards the final months, reflecting a gradual stabilization of the market.The confidence interval (grey band) slightly widens towards the end, indicating an increase in uncertainty for longer-term forecasts, but it remains within a reasonable range. This proves that the model maintains reliability for short-term forecasting.

The SARIMA(4,0,3)(1,1,1)[12] model demonstrates a good capability to simulate the trend and seasonality of ceramic revenue, with a stable increase forecasted for 2025–2026. This result suggests the model is suitable for short-term forecasting objectives and can be used to support planning for production, inventory, and product distribution in the coming year.

XGBOOST:

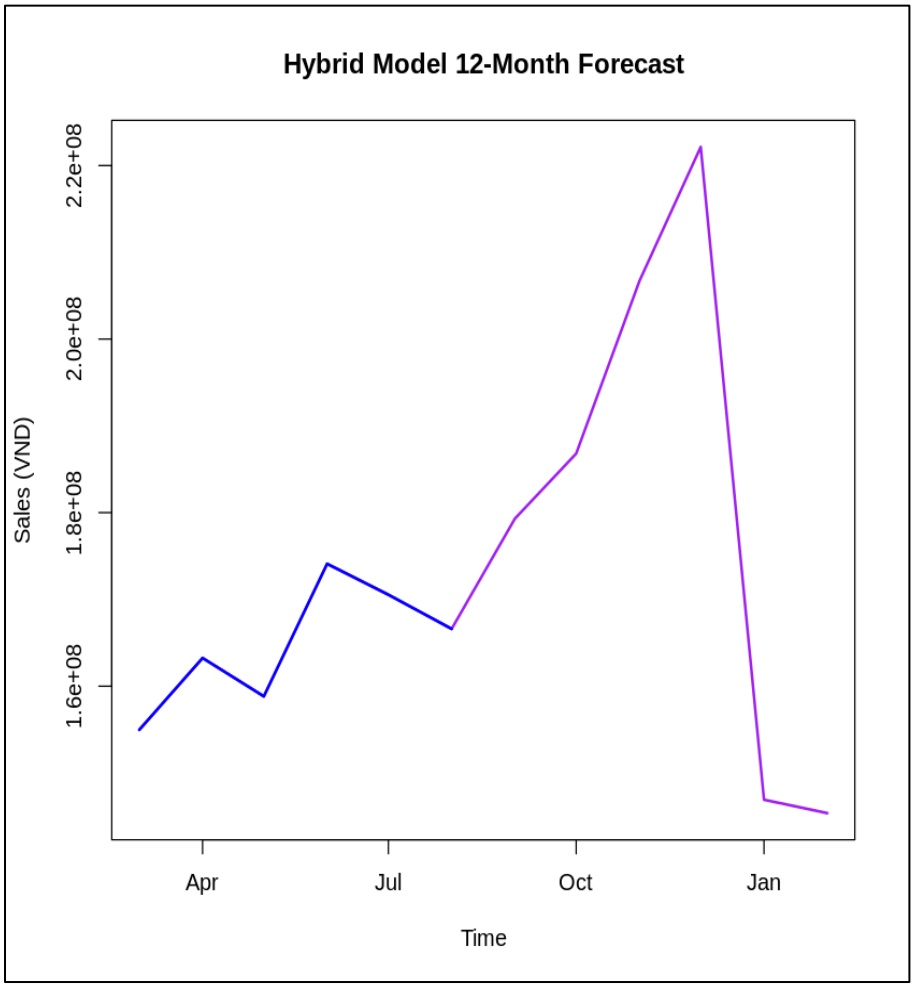

Figure 78. The last forecasting: hybrid of HW1 + XGboost

The chart above illustrates the 12-month forecast of ceramic sales (from March 2025 to February 2026) generated by a Hybrid Model, which combines SARIMA (a linear statistical model) and XGBoost (a nonlinear machine learning model). The input data were standardized during training and later unscaled back to the actual unit (VND), ensuring that the values on the vertical axis accurately represent the predicted sales figures.

From the observed trend, ceramic sales are projected to rise steadily from March to December 2025, reflecting a positive recovery and consistent growth in the ceramics market.

During March to June 2025, sales increase gradually, showing stable consumption following the early-year period.

From July to November 2025, sales accelerate more sharply, indicating the market is entering a peak production and consumption season, often linked to export activities and increased demand for decoration and gifting during year-end holidays.

The highest forecasted sales occur in December 2025, reaching approximately 220 million VND, consistent with the seasonal pattern of the handicraft industry, which typically experiences a strong performance in the fourth quarter of each year.

However, after this peak period, the model predicts a significant drop in January–February 2026, with sales falling to around 150 million VND. This decline likely reflects the post-peak seasonal effect, as consumer demand and export orders usually decrease following the Lunar New Year period.

From a technical perspective, the Hybrid (SARIMA–XGBoost) model demonstrates strong performance by effectively combining:

SARIMA, which captures long-term trends and stable seasonal patterns, and

XGBoost, which handles short-term fluctuations and nonlinear residual corrections, improving overall forecast accuracy.

As a result, the model successfully captures both seasonal variations and real-world fluctuations in sales, particularly during transitional market phases.

The Hybrid (SARIMA–XGBoost) model delivers accurate and realistic forecasting results, showing a steady upward trend during the first three quarters of 2025 and a peak in December. It is therefore a reliable tool for short-term sales forecasting, enabling businesses to plan production, inventory, and marketing strategies proactively, and to optimize resources during the high-demand year-end season.

Chapter 3
DISCUSSION AND CONCLUSION

Discussion

The research results demonstrate that different imputation and forecasting approaches produce varying levels of accuracy in predicting ceramic sales revenue. Among them, the Hybrid SARIMA–XGBoost model achieved the best overall performance, outperforming both traditional statistical and standalone machine learning methods.

The Holt–Winters Based Imputation method proved effective for handling missing values, as it accurately reconstructed the seasonal and trend components inherent in the ceramic sales data. This approach ensured the continuity and reliability of the time series before forecasting.

The SARIMA model captured the autoregressive and seasonal structures of the data well, producing a stable short-term forecast. However, its linear nature limited its responsiveness to sudden market fluctuations or nonlinear interactions in sales dynamics. The XGBoost model, on the other hand, excelled in modeling complex nonlinear relationships and residual variations that SARIMA could not fully explain.

By combining the strengths of both models, the Hybrid SARIMA–XGBoost approach successfully enhanced predictive performance. It maintained SARIMA's ability to model seasonality and trend while leveraging XGBoost's flexibility to capture short-term irregularities. The hybrid model's forecast closely aligned with actual data, particularly during the rapid growth period from mid to late 2025, confirming its superior adaptability and accuracy.

Overall, the 12-month forecast (from March 2025 to February 2026) indicates a steady upward trend in ceramic sales, peaking in December 2025 before slightly declining in early 2026. This outcome reflects the strong seasonality of the industry — high sales during festive and export seasons followed by post-peak slowdowns — and provides a solid foundation for strategic business planning.

Given the reliability and interpretability of the Hybrid SARIMA–XGBoost model, its forecast results offer practical guidance for business decisions. The projected upward trend and seasonal fluctuations highlight opportunities for growth and the need for efficient resource management. These insights serve as the basis for developing strategic plans that enable the company to capitalize on high-demand periods, optimize operations, and maintain stability during low-season months.

Summary of the forecast period:

The forecast results indicate that the revenue of the ceramics industry will increase significantly from the second quarter to the fourth quarter of 2025, peaking at the end of the year, and then slightly declining in the first quarter of 2026. This trend clearly reflects the seasonal nature of the industry — sales usually rise during holidays and year-end periods, then drop afterward. From a business perspective, this period represents both a golden opportunity to expand market share and a challenge in maintaining stable resources and financial balance.

Production – Supply Strategy:

The production and supply strategy is established with the dual objective of ensuring sufficient output and maintaining flexibility against market demand fluctuations. On the production front, the company will increase capacity by 20–30% during the peak period from June to November 2025 to prepare for the revenue peak in December. To ensure continuity, the supply strategy focuses on early inventory stocking of key raw materials (such as clay, glaze, and packaging) before Q3 to prevent price increases or shortages late in the year. Operationally, the company will optimize the production schedule: applying the Just-In-Time (JIT) method in Q1 and Q2 2025 to reduce inventory costs, then switching to Build-to-Stock in Q3 and Q4 to be ready for the high season. For effective implementation, the company needs to utilize ERP or MRP (Material Requirement Planning) systems to track inventory and coordinate production based on forecasted data. Concurrently, the company will diversify its suppliers, prioritizing domestic sources to reduce shipping costs, and sign long-term framework agreements with key suppliers to ensure a stable supply of materials throughout the 6-month peak period.

Human Resources – Operations Strategy:

The Human Resources and Operations strategy is designed to maintain high productivity during the peak demand period while simultaneously minimizing labor costs when demand cools down. On the personnel side, the company will implement seasonal recruitment or short-term contract extensions for the peak period from August to December 2025. To ensure product quality, the company prioritizes pre-peak technical training, with a special focus on specialized techniques such as firing, shaping, packaging, and quality control. Investing in in-depth technical training for shapers and kiln operators is essential to reduce the product defect rate. Furthermore, the company will apply a monthly performance bonus policy to incentivize and motivate employees to work productively during the high season. For implementation, the company will collaborate with vocational schools or local training centers for recruitment and labor training. Concurrently, the automation of certain repetitive tasks (such as packaging, sorting, and classification) will be carried out to reduce dependence on seasonal labor. Finally, building an internal motivation program is critical to encouraging skill enhancement and boosting morale.

Marketing – Sales Strategy:

The marketing and sales strategy is focused on leveraging the projected growth phase to expand market share and strengthen the brand. Specifically, the company will increase the marketing budget by 40% in Q3 and Q4 2025, concentrating mainly on end-of-year and Lunar New Year campaigns. Heavy investment in promotional programs and Tet gift combos will be executed to target the surging demand for decoration and gifting during the year-end holiday season. Additionally, the company will launch a new collection or limited edition product lines in October–December to create a shopping buzz. Regarding distribution channels, the strategy involves expanding online sales (e-commerce) alongside direct exports, specifically targeting Asian markets that celebrate the Lunar New Year. The company will intensify communications via social media (Facebook, TikTok, Instagram) and e-commerce platforms (Shopee, Lazada, Etsy), while simultaneously increasing discounts for agents and wholesale partners during the strong growth period. Furthermore, organizing post-Tet promotional programs (January–February 2026) will help clear inventory and maintain stable cash flow.

Financial – Risk Strategy:

The Financial and Risk strategy is established with the core objectives of ensuring stable cash flow and mitigating risks that may arise following the peak growth period. Specifically, the company will increase its cash reserves by 10 –15% before December 2025 to maintain liquidity when revenue typically declines at the beginning of 2026. To minimize exposure, the company needs to diversify its customer portfolio, avoiding excessive reliance on a single market or partner group. Additionally, the company will seek to stagger payments with suppliers or logistics partners to alleviate short-term financial pressure. For external risks, export risk insurance will be applied, along with proactive measures to hedge against fluctuations in raw material prices and exchange rates. Operationally, the company will work with its partner bank to establish necessary short-term credit limits during the peak season. Concurrently, the company will create quarterly budget plans tied closely to actual revenue forecasts, and periodically monitor the forecasting effectiveness of the Hybrid model to adjust financial plans flexibly.

Overall conclusion

This study focuses on forecasting ceramic sales revenue based on time series data from May 2017 to August 2025, with the goal of developing an accurate and practical forecasting model to support business strategy planning. After thorough data preprocessing, analysis, and model evaluation, the Hybrid SARIMA–XGBoost model was identified as the optimal approach, producing the lowest error rates and the most realistic representation of revenue fluctuations.

During the preprocessing phase, the Holt–Winters Based Imputation method was applied to handle missing data, effectively reconstructing the underlying trend and seasonality — two key characteristics of the ceramic industry. Subsequently, the SARIMA model was employed to capture linear and seasonal components, while XGBoost was used to learn the residuals,

modeling the nonlinear and complex interactions that SARIMA could not fully explain. This integration resulted in a flexible and robust hybrid system that significantly improved forecasting accuracy.

The 12-month forecast (from September 2025 to August 2026) reveals a strong upward trend in ceramic sales revenue from Q2 to Q4 of 2025, peaking at the end of the year, followed by a slight decline during Q1 of 2026. The model successfully captures the seasonal characteristics of the ceramic industry — with sales surging during festive and export periods — while maintaining a high level of stability and short-term forecasting reliability.

From a business perspective, these forecasting results serve as a valuable foundation for strategic planning. During the revenue growth phase, companies can expand production capacity, strengthen market presence, enhance marketing activities, and optimize supply chains to capitalize on rising demand. Conversely, in the post-peak period, firms should focus on cost management, inventory balancing, and operational efficiency to maintain long-term sustainability.

Overall, this research demonstrates the effectiveness of the Hybrid SARIMA–XGBoost model in forecasting time series data with strong seasonal patterns. The model not only delivers high predictive accuracy for ceramic sales revenue but also provides meaningful insights for strategic decision-making, production and financial planning, and sustainable business development in an increasingly competitive market environment.

References

DataCamp. (n.d.). *Handling missing data with imputations in R: Donor-based imputation* [Online course module]. Retrieved from https://campus.datacamp.com/courses/handling-missing-data-with-imputations-in-r/donor-based-imputation?ex=9

datanerddhanya. (n.d.). *Data analysis using SVM classifications*. RPubs. Retrieved from https://rpubs.com/datanerddhanya/1300125

Hyndman, R. J., & Athanasopoulos, G. (2018). Holt-Winters' seasonal method. In *Forecasting: Principles and Practice* (2nd ed.). OTexts. Retrieved from https://otexts.com/fpp2/holt-winters.html

Wickham, H., Çetinkaya-Rundel, M., & Grolemund, G. (2023). Missing values. In *R for Data Science* (2nd ed.). O'Reilly. Retrieved from https://r4ds.hadley.nz/missing-values.html