PSTAT 274

Fall 2023

December 9, 2023

PSTAT 274 Final Project

Forecasting Billing Amount

Student

Kai Wa Ho

Instructor

Raisa Feldman

Category

1. Exclusive Summary

The purpose of this synthetic health dataset is to meet the needs of data science, machine learning, and data analytics enthusiasts in the healthcare industry. This extensive structure covers a wide range of data, including patient information, hospitalization records, and med- ical service facts. The need for useful and diverse health data for research and education purposes is driving the development of this collection. Privacy regulations often limit access to real-world health data, creating difficulties for those wishing to research and experiment in this area. To overcome this limitation, the dataset was created using his Faker package in Python to ensure that its attributes and structure were very similar to those of real health records.

This synthetic data allows users to improve their data manipulation and analysis skills on a secure and ethical platform. Due to its flexibility, data sets can be used in a variety of applications. Provides a platform for practicing data transformation, cleaning, and prepro- cessing techniques, supports the creation and testing of predictive models related to health outcomes, and provides informative visualizations that help identify and understand trends in healthcare Provides strong support for creating. Additionally, it is an invaluable resource for teaching and learning about data science and machine learning topics in the medical field.

The main goal of the project is to predict the key variable: **Billing amount**.

2. Introduction

This project focuses on predicting the key variable: **billing amount**. The main goal is to apply predictive modeling techniques to the provided dataset to predict these important components. Each of these variables is important in healthcare and contributes to operational efficiency, financial planning, and patient care. In the field of medical financing, accurately predicting bill amounts is paramount.

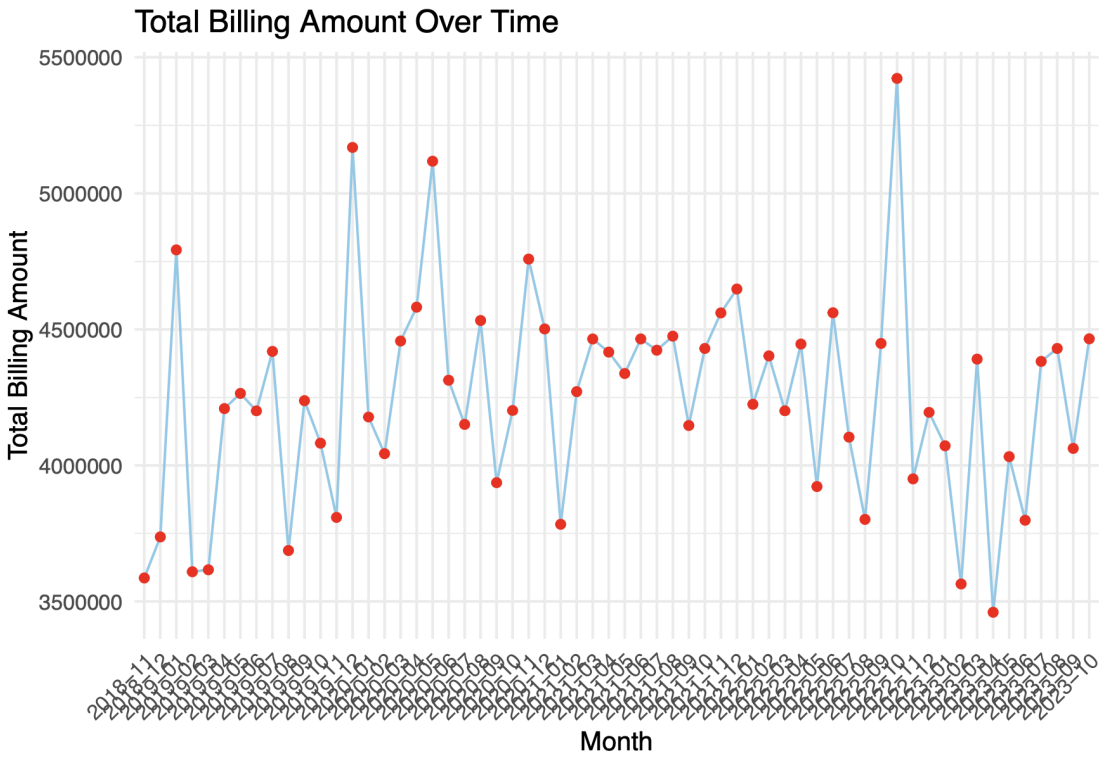
The purpose of this forecasting task is to provide a reliable estimate of the financial costs associated with a particular medical case. Such forecasts have practical value for both health- care providers and patients, as they enable transparent cost estimates and help effectively manage healthcare spending.

Furthermore, the generated predictions can positively impact patient care by improving bed management practices and facilitating advance planning of post-discharge support. This project's approach may include the use of machine learning or statistical modeling techniques. Characteristics such as patient information, medical history, and hospitalization information can be considered as input variables for prediction. Challenges related to handling missing or incomplete data, accounting for diverse medical cases, and ensuring ethical use of predictive analytics in healthcare are recognized.

3. Model Development

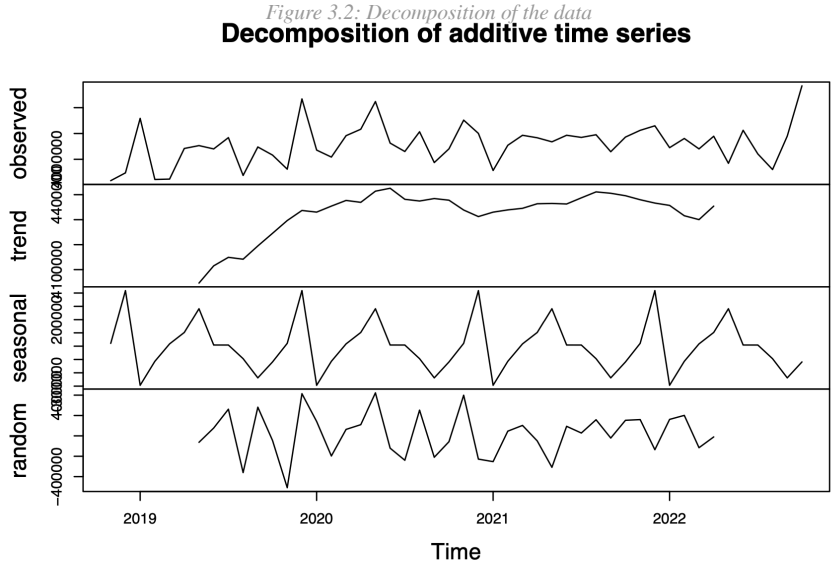
The figure offers a holistic view of the dataset, covering 60 months (5 years) of data and benefiting from monthly updates, providing a dynamic and thorough perspective for analysis.

Figure 3.1: Comprehensive Overview of the data



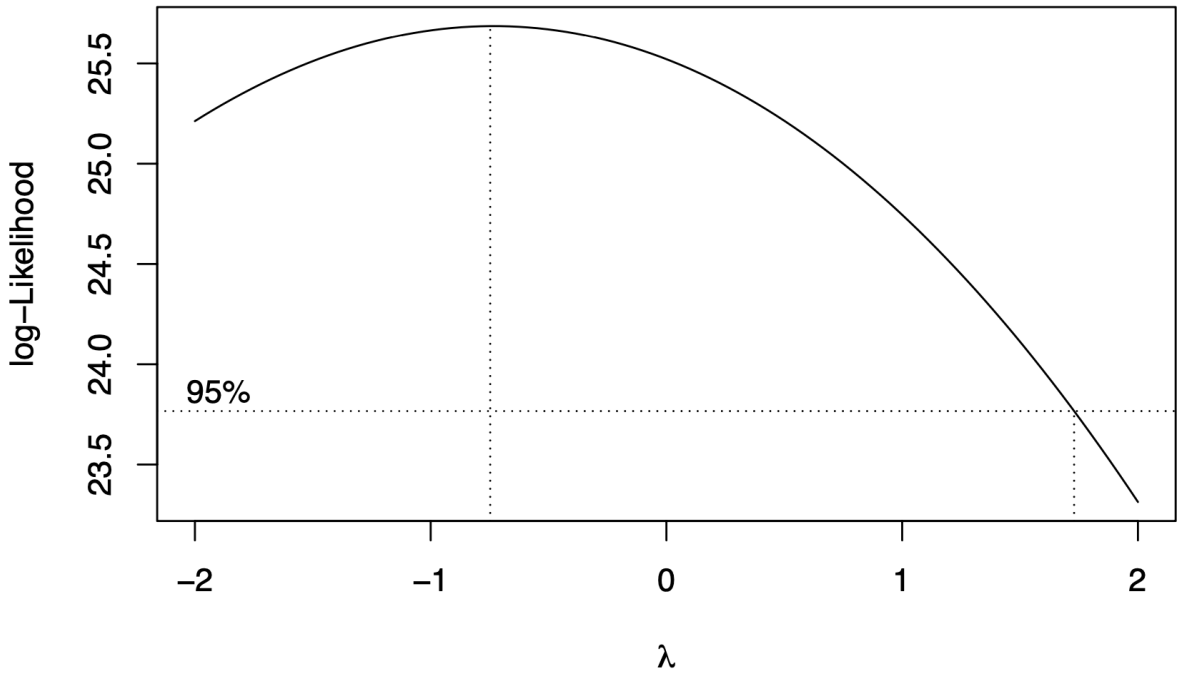
Now, let's examine the fundamental attributes of the dataset.

Figure 3.2: Decomposition of the data



The time series exhibits a pronounced upward trend from 2019 to 2021, followed by a sharp decline in 2021, and subsequent resurgence until 2022. Notably, the data does not exhibit a discernible seasonal pattern; however, variance fluctuates between 2019 and 2021 before stabilizing between 2021 and 2022. Given the non-stationary nature of time series characterized by varying trends and fluctuations in variance, it is prudent to explore transformation methods. Additionally, a recurring end-of-year surge in bills is evident across each annual cycle.

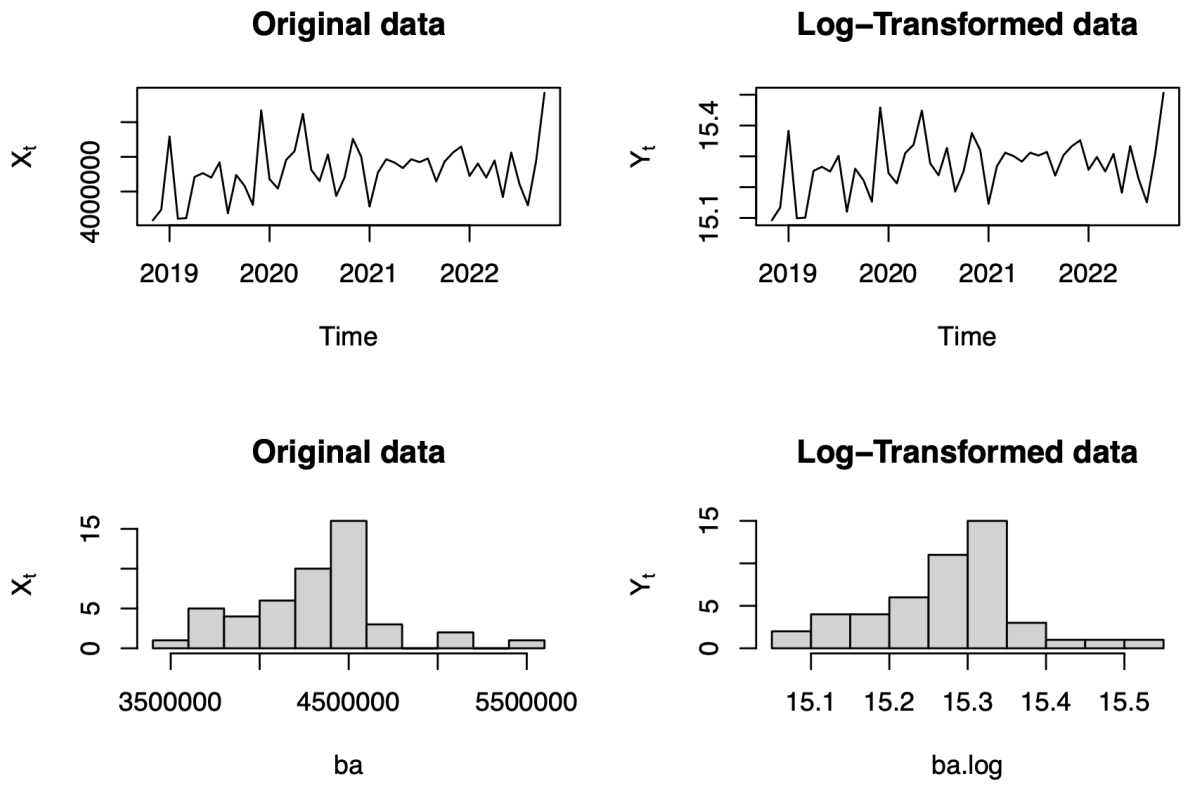
Figure 3.3: Box-Cox Transformation



Since the 95% CI for the true value of λT includes λ = 0, we should consider log transformation.

3.1 Log-transformation

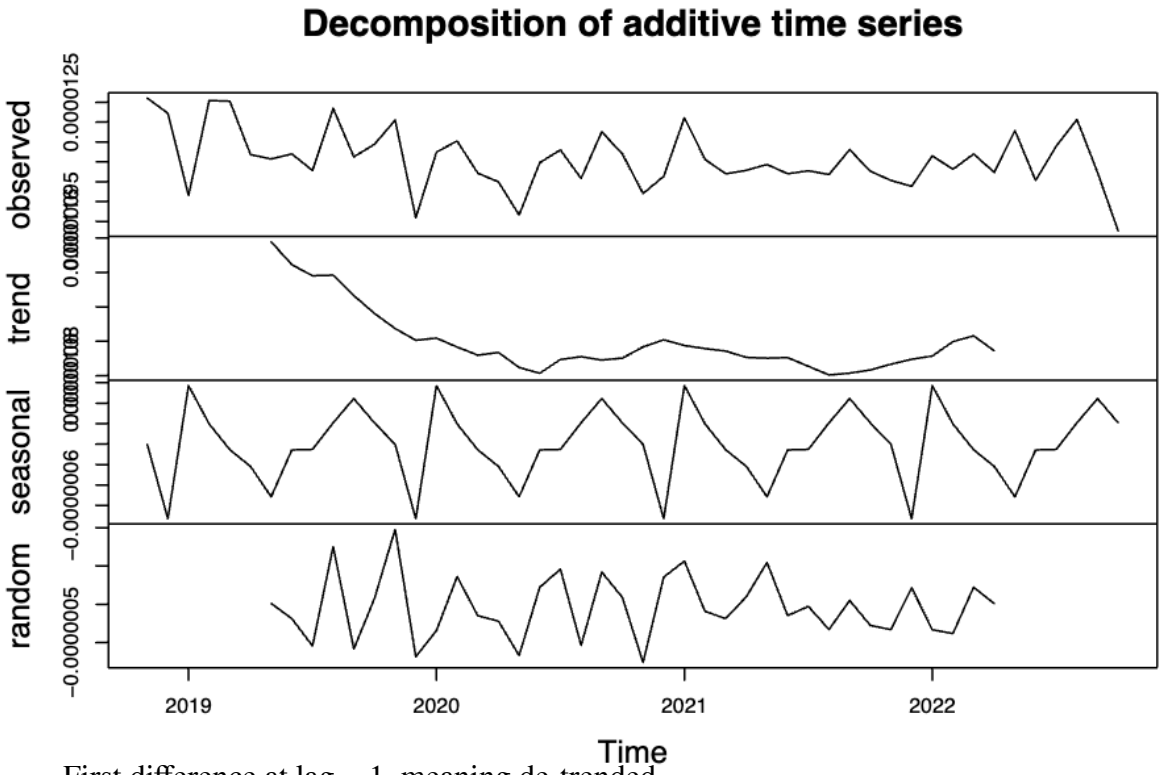
Figure 3.1.1: Log Transformation



The log-transformed data displays a right-skewed distribution, deviating from a Gaussian shape. Meanwhile, the decomposition of the dataset reveals a persistent upward trend with regular oscillations. To address these characteristics, it is advisable to explore the application of a Box-Cox transformation for detrending and deseasonalizing the data.

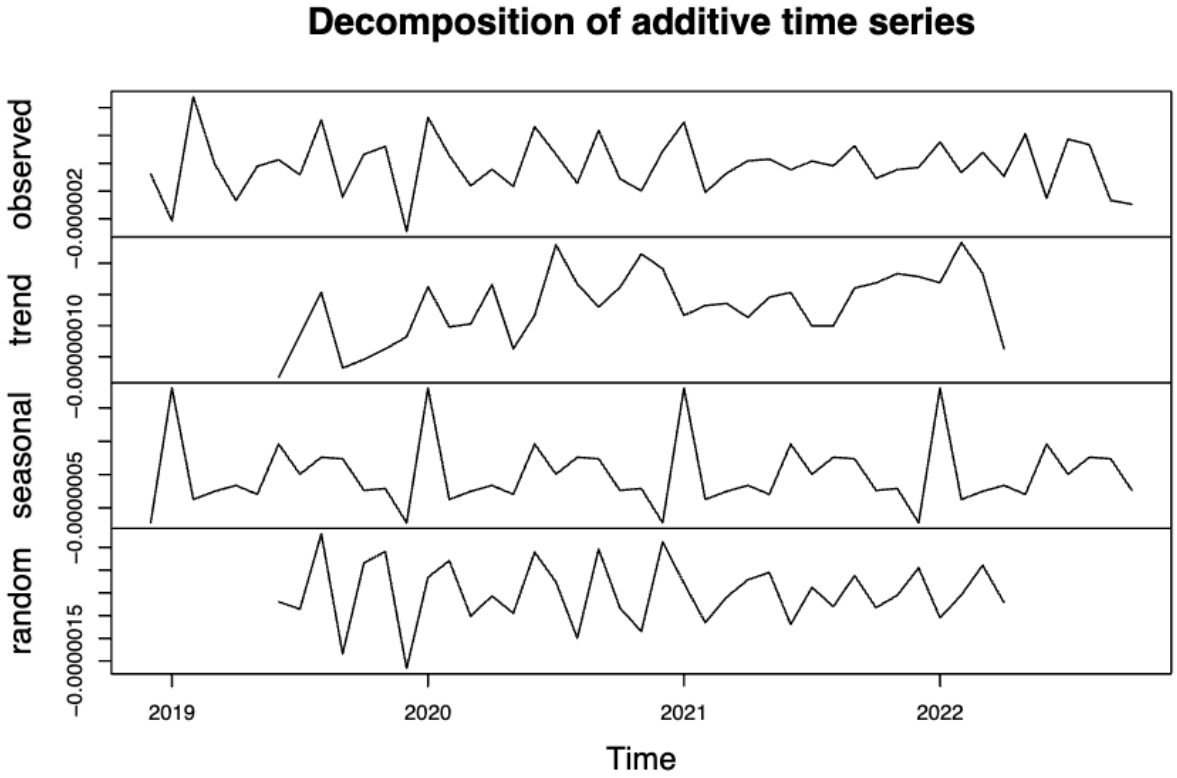
3.2 Box-Cox transformation

Figure 3.2.1: Decomposition of the box-cox transformed time series



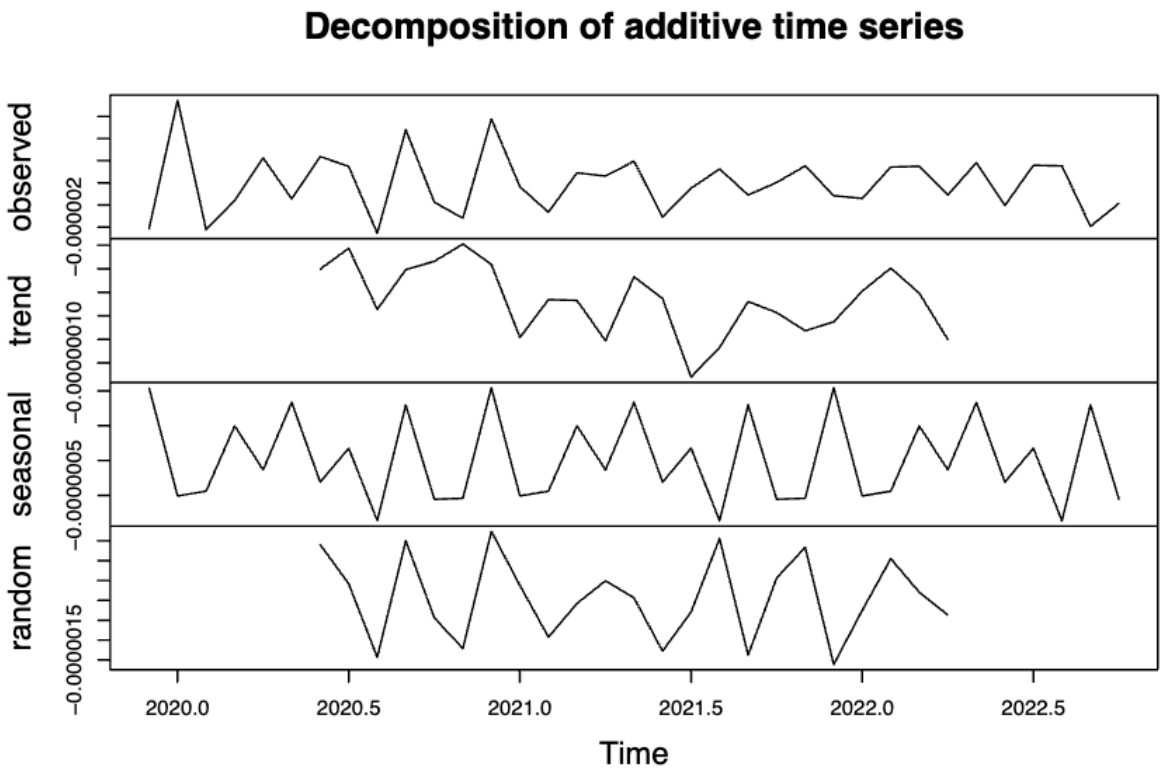
First difference at lag = 1, meaning de-trended.

Figure 3.2.2: Decomposition of the box-cox transformed, de-trended, time series



After taking difference at lag 1, there is no obvious trend, next step is to difference at lag 12.

Figure 3.2.3: Decomposition of the box-cox transformed, de-trended, de-seasonalised, time series

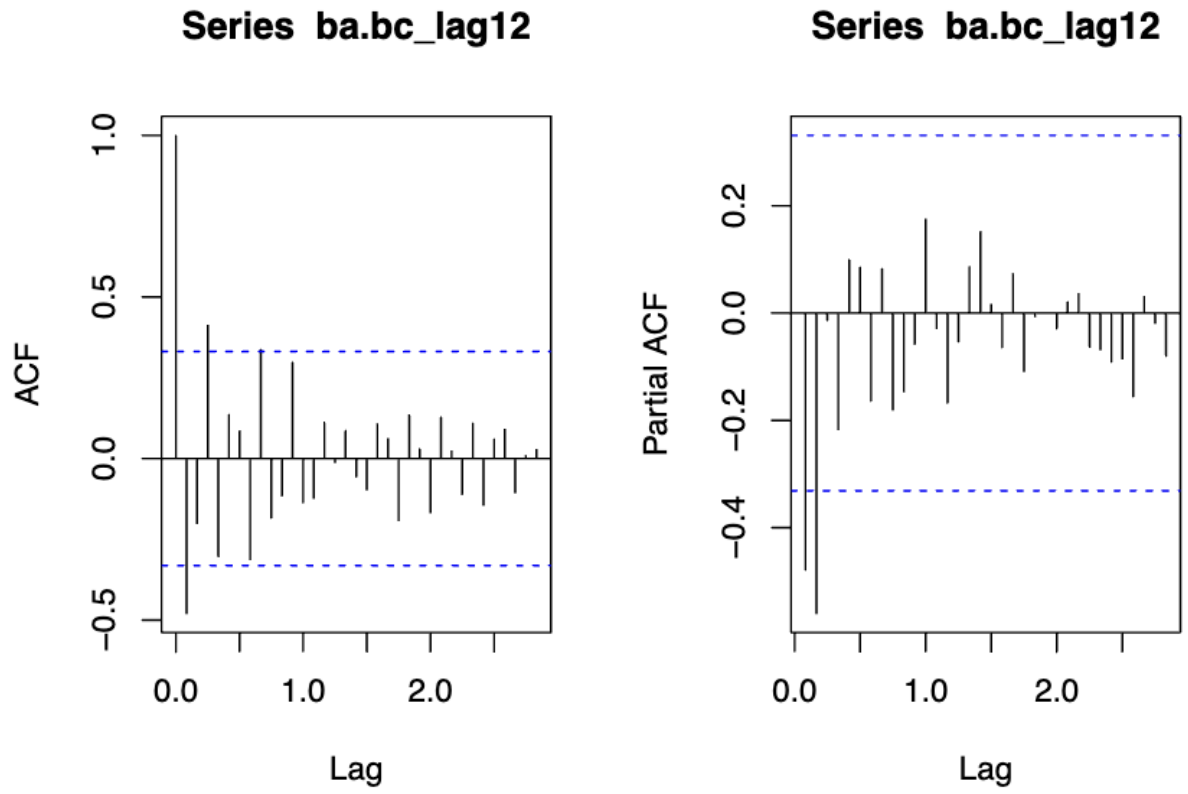


Following differencing at a lag of 12, discernible seasonality diminishes, and while the variance of the original data is 0.0000000000005617475, the de-trended data exhibits a variance of 0.0000000000009797723, and the deseasonalized data shows a variance of 0.000000000001956692. Despite a marginal increase in variance, the decision to proceed with the de-trended and deseasonalized data is justified as the increment is negligible and supported by the decomposition, confirming the successful removal of trend and seasonality components.

Next step, plotting ACF and PACF to identify p and q for the model.

3.3 Identification of the model

Figure 3.3.1: ACF and PACF of the stationary time series



After differencing the time series at lag 12, the ACF and PACF plots, accounting for seasonality, reveal an absence of peaks at lags 12, 24, and 36. Instead, the ACF plot shows peaks at lags 1, 2, and 4, while the PACF plot exhibits a peak at lag 1. Consequently, the derived model specifications are D = d = 1, P = Q = 0, p = 1, q = 1, 2, 4, and s = 12, resulting in the identification of three potential models:

SARIMA(1, 1, 1) x (0, 1, 0) $_{s=12}$

SARIMA(1, 1, 2) x (0, 1, 0) $_{s=12}$

SARIMA(1, 1, 4) x (0, 1, 0) $_{s=12}$

3.4 Model Structure

Model 1:

Figure 3.4.1: Model 1

```
Call:
arima(x = as.numeric(ba.bc), order = c(1, 1, 1), seasonal = list(order = c(0,
    1, 0), period = 12), method = "ML")

Coefficients:
          ar1      ma1
      -0.1879  -0.8417
s.e.   0.1921   0.1109

sigma^2 estimated as 0.00000000000092:  log likelihood = 434.56,  aic = -863.12
```

The model can be written as:

$$(1 + 0.1879B)\, X_t = (1 - 0.8417B)\, Z_t$$

AR part: 0.1879 < 1, the root is greater than 1

MA part: 0.8417 < 1, the root is greater than 1

Conclude that the model is stationary and invertible.

Model 2:

Figure 3.4.2: Model 2

```
Call:
arima(x = as.numeric(ba.bc), order = c(1, 1, 2), seasonal = list(order = c(0,
    1, 0), period = 12), method = "ML")

Coefficients:
         ar1      ma1     ma2
      0.0993  -1.2141  0.3685
s.e.  0.3690   0.3420  0.3289

sigma^2 estimated as 0.0000000000008873:  log likelihood = 435.17,  aic = -862.34
```

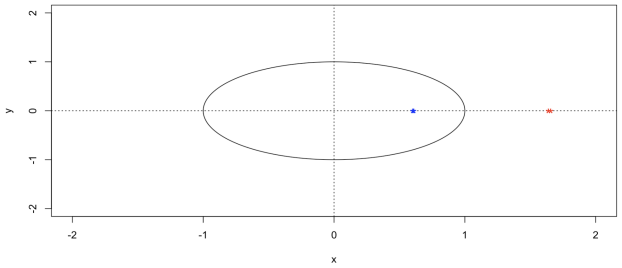
The model can be written as:

$$(1 - 0.0993B)\, X_t = (1 - 1.2141B + 0.3685B^2)\, Z_t$$

AR part: 0.0993 < 1, so the root is greater than 1

MA part: By plot.roots from R output, the roots is greater than 1

Figure 3.4.3: Roots of MA part



Conclude that the model is stationary and invertible.

Model 3:

Figure 3.4.4: Model 3

```
Call:
arima(x = as.numeric(ba.bc), order = c(1, 1, 4), seasonal = list(order = c(0,
    1, 0), period = 12), method = "ML")

Coefficients:
         ar1      ma1     ma2     ma3      ma4
      0.1359  -1.1544  0.0615  0.6532  -0.4808
s.e.  0.3715   0.3354  0.3817  0.2014   0.3265

sigma^2 estimated as 0.0000000000007202:  log likelihood = 438.37,  aic = -864.73
```

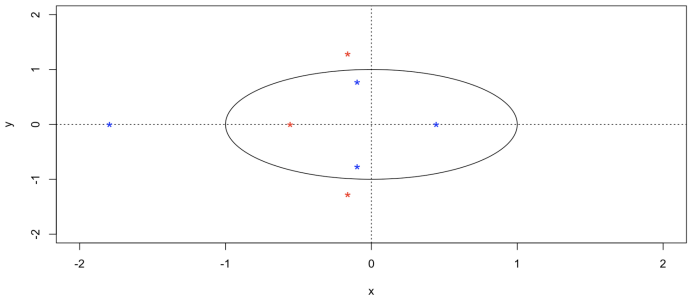
The model can be written as:

$$(1 - 0.1359B)\, X_t = (1 + 1.544B + 0.0615B^2 + 0.6532B^3 - 0.4808B^4)\, Z_t$$

AR part: 0.13598 < 1, so the root is greater than 1

MA part: By plot.roots from R output, the roots is greater than 1

Figure 3.4.5: Roots of MA part



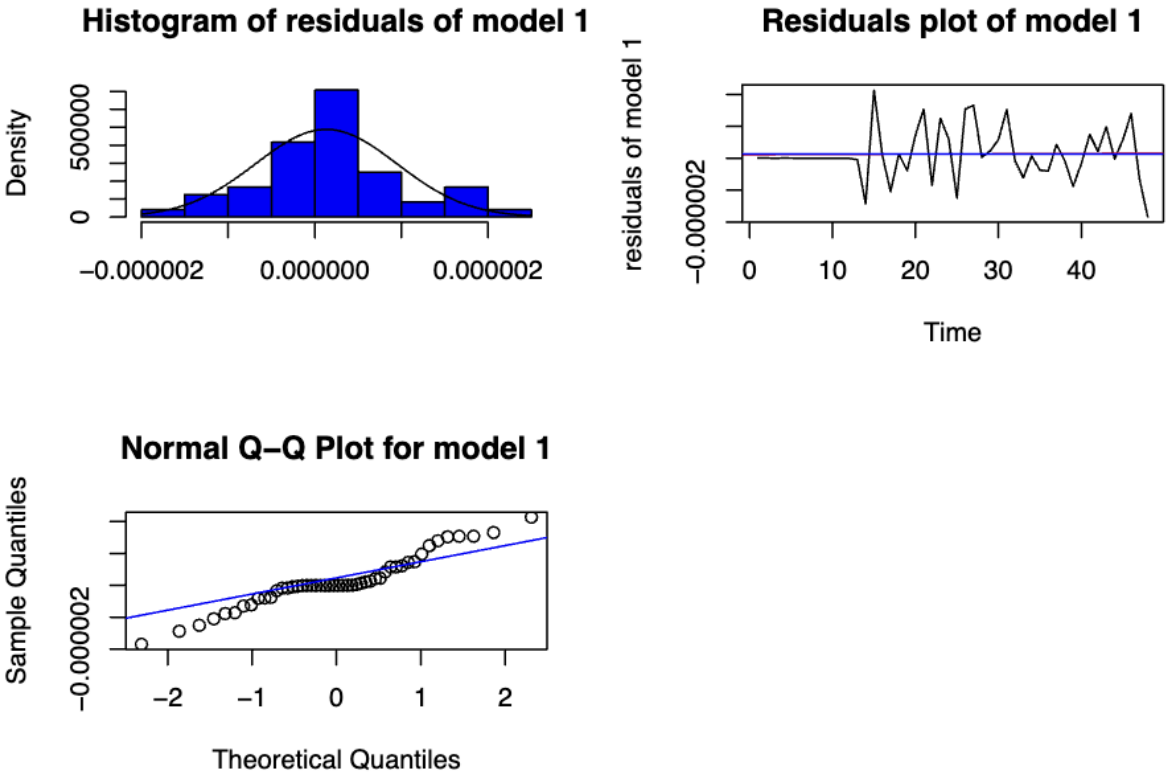
Conclude that the model is stationary and invertible.

While the R output indicates that Model 3 has the lowest AICc, it is imperative to conduct additional diagnostic analyses to validate its robustness and ensure its appropriateness for forecasting.

3.5 Checking Resdiuals of the Model

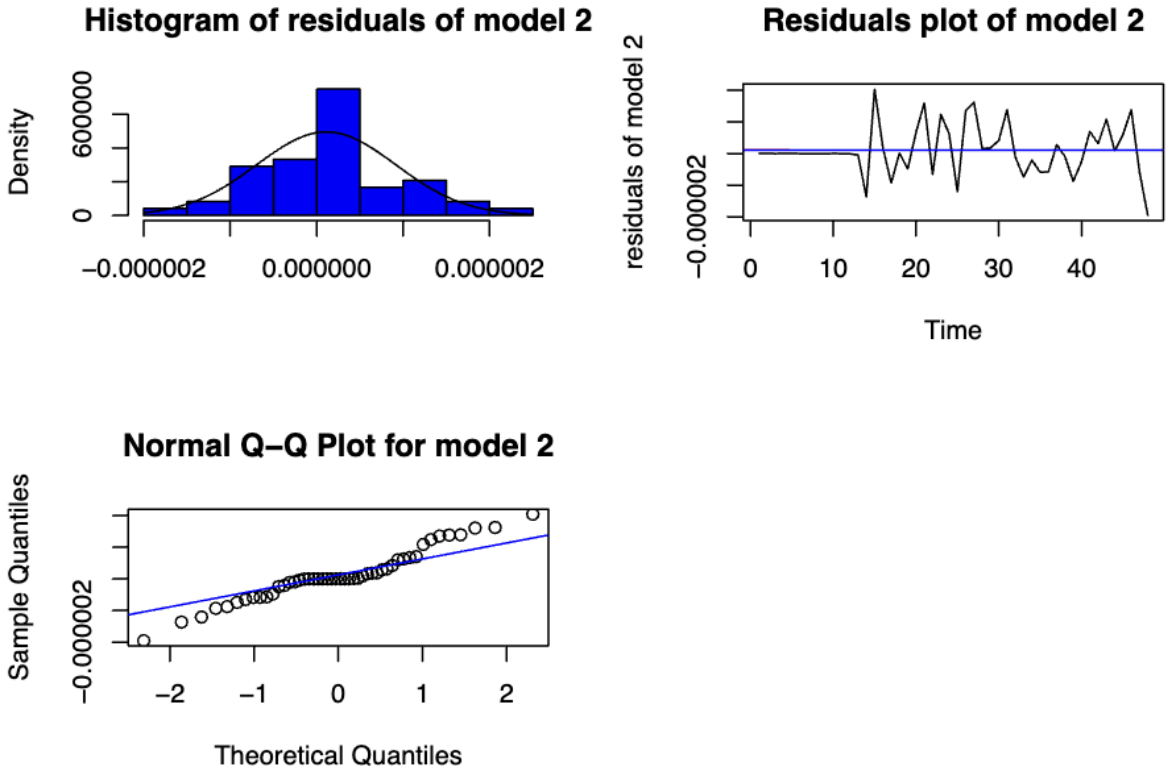
Model 1:

Figure 3.5.1: Diagnoestic plots of the residuals of model 1



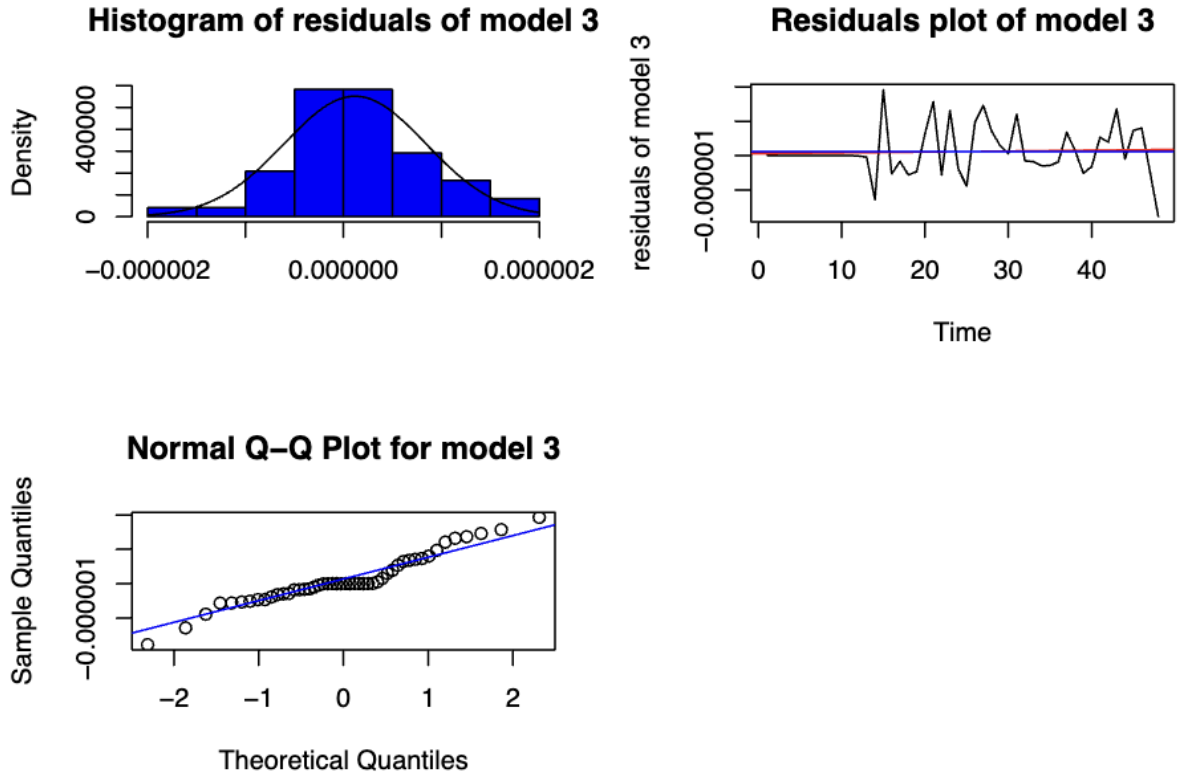
Model 2:

Figure 3.5.2: Diagnoestic plots of the residuals of model 2



Model 3:

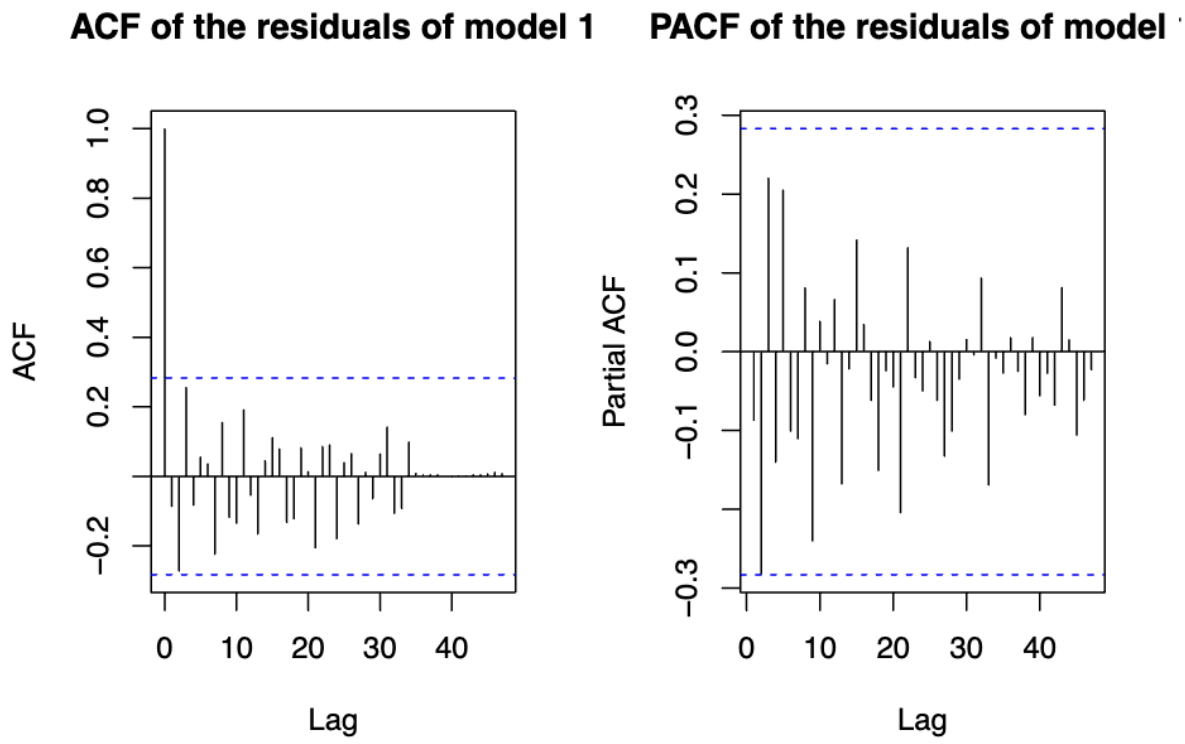
Figure 3.5.3: Diagnoestic plots of the residuals of model 3



Upon inspecting the aforementioned plots, it is evident that only Model 3 exhibits a marginal adherence to the normal distribution, with its histogram displaying a more Gaussian shape compared to the other two models. Subsequently, the next step involves utilizing ACF and PACF plots to ascertain whether the residuals of Model 3 demonstrate characteristics of white noise.

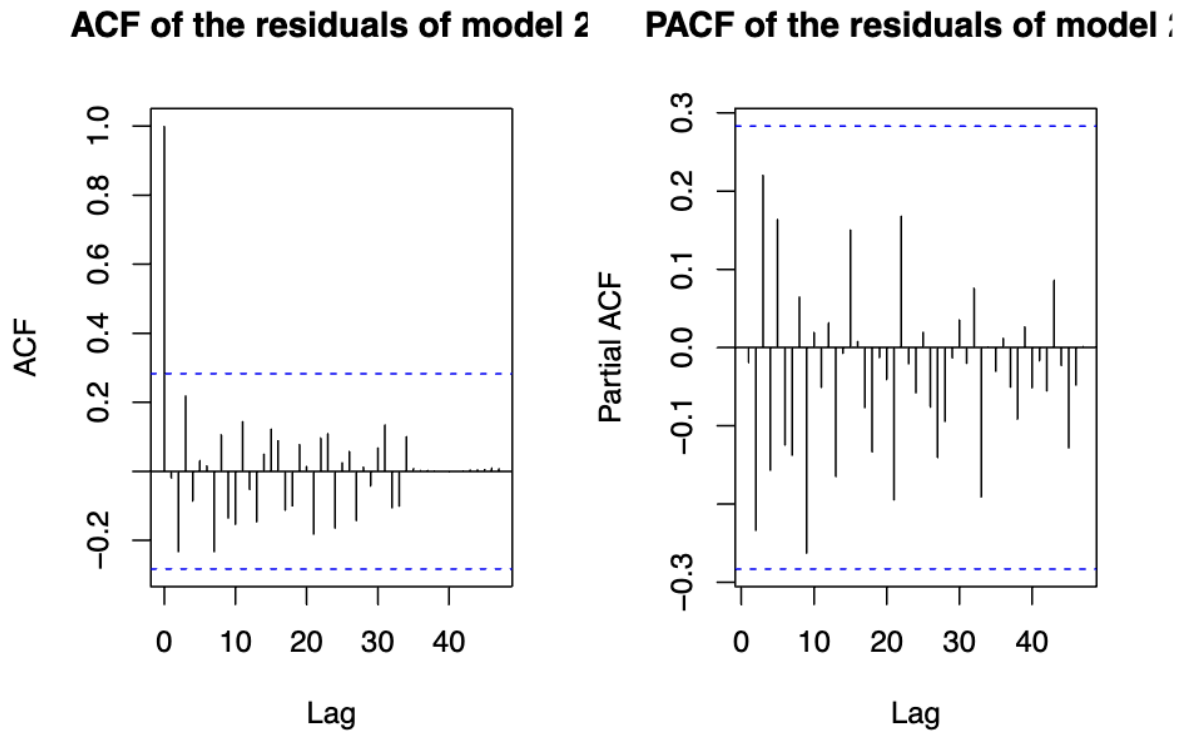
Model 1:

Figure 3.5.4: ACF and PACF of the residuals of model 1



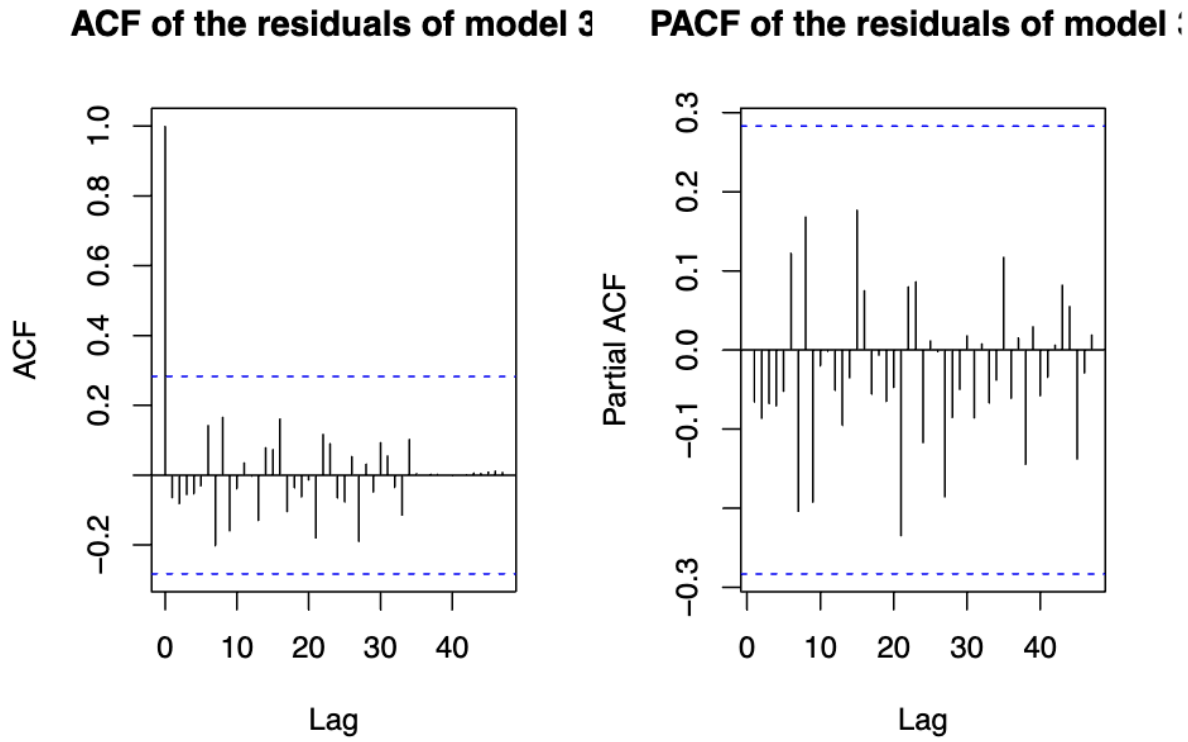
Model 2:

Figure 3.5.5: ACF and PACF of the residuals of model 2



Model 3:

Figure 3.5.6: ACF and PACF of the residuals of model 3



Across all models, the residuals consistently fall within the confidence interval, indicating that they conform to a white noise pattern. Consequently, additional diagnostic checks are warranted to ensure the reliability and accuracy of the models.

3.6 Shapiro test, Box-Pierce test, Ljung-Box test, McLeod-Li test, and Yule-Waker test

Model 1:

Figure 3.6.1: Shapiro test, Box-Pierce test, Ljung-Box test, McLeod-Li test, and Yule-Waker test for model 1

```
        Shapiro-Wilk normality test

data:  res
W = 0.94616, p-value = 0.02818

        Box-Pierce test

data:  res
X-squared = 10.012, df = 5, p-value = 0.07489

        Box-Ljung test

data:  res
X-squared = 11.303, df = 5, p-value = 0.0457

        Box-Ljung test

data:  res^2
X-squared = 5.0583, df = 7, p-value = 0.6529

Call:
ar(x = res, aic = TRUE, order.max = NULL, method = c("yule-walker"))

Coefficients:
      1       2       3
-0.0493  -0.2573   0.2205

Order selected 3  sigma^2 estimated as  0.0000000000006195
```

The R output above reveals that Model 1 fails both the Box-Ljung test and Yule-Walker test, aligning with the previous observation in Section 3.5 that the residuals do not adhere to a white noise pattern for this particular model.

Model 2:

Figure 3.6.2: Shapiro test, Box-Pierce test, Ljung-Box test, McLeod-Li test, and Yule-Waker test for model 1

```
        Shapiro-Wilk normality test

data:  res
W = 0.95508, p-value = 0.06392

        Box-Pierce test

data:  res
X-squared = 7.935, df = 4, p-value = 0.09399

        Box-Ljung test

data:  res
X-squared = 9.0474, df = 4, p-value = 0.05993

        Box-Ljung test

data:  res^2
X-squared = 4.9935, df = 7, p-value = 0.6608

Call:
ar(x = res, aic = TRUE, order.max = NULL, method = c("yule-walker"))


Order selected 0  sigma^2 estimated as  0.0000000000006483
```

From the above R output, model 2 passes all the tests, it is ready for forecasting.

Model 3:

Figure 3.6.3: Shapiro test, Box-Pierce test, Ljung-Box test, McLeod-Li test, and Yule-Waker test for model 3

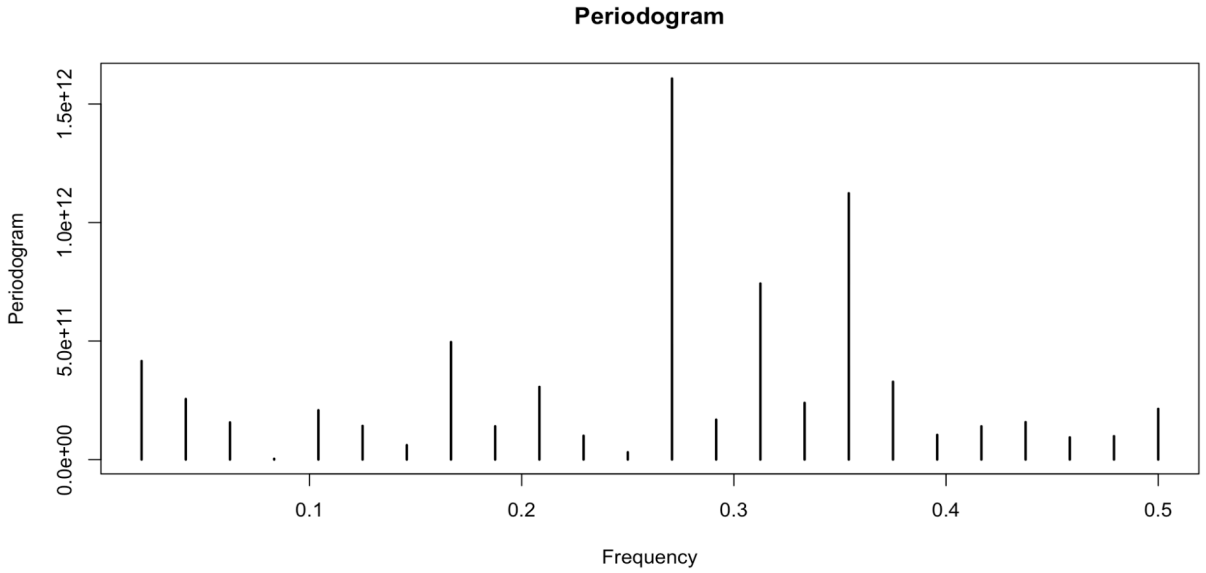
```
        Shapiro-Wilk normality test

data:  res
W = 0.94508, p-value = 0.02557


        Box-Pierce test

data:  res
X-squared = 3.7787, df = 2, p-value = 0.1512

        Box-Ljung test

data:  res
X-squared = 4.4752, df = 2, p-value = 0.1067

         Box-Ljung test

data:  res^2
X-squared = 4.4227, df = 7, p-value = 0.73

Call:
ar(x = res, aic = TRUE, order.max = NULL, method = c("yule-walker"))


Order selected 0  sigma^2 estimated as  0.0000000000005221
```

From the above R output, the model 3 passes all the tests except Shapiro-Wilk test, it is ready for foreasting.

4. Spectral Analysis

4.1 Periodogram

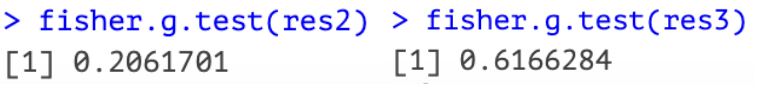
Figure 4.1: Periodogram



Examining the graph presented earlier, noticeable spikes are observed at approximately 0.266667 and 0.316667, indicating a clear presence of periodicity and seasonality in the data.

4.2 Fisher's test

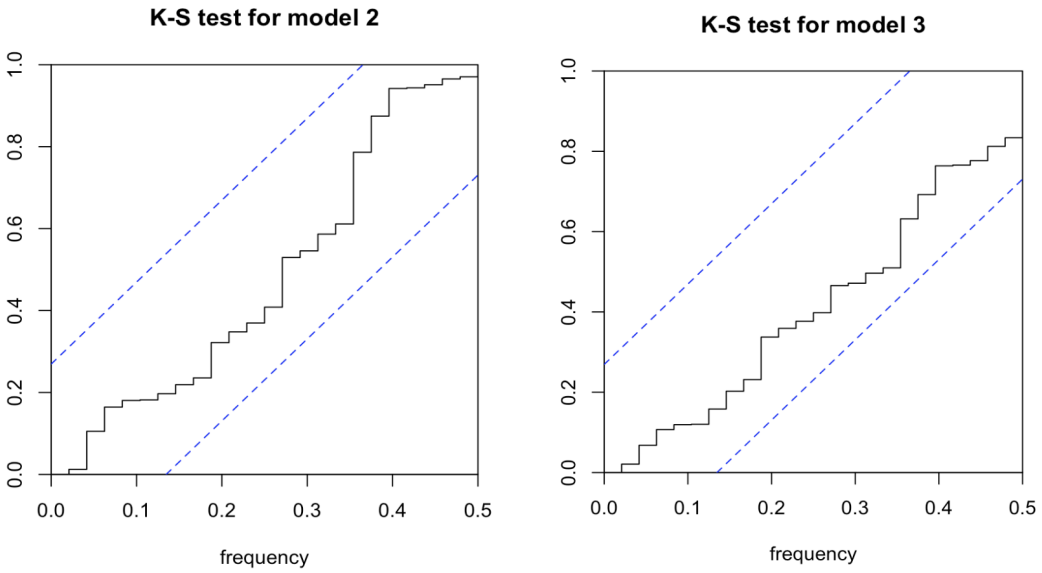
Figure 4.2: Fisher's test



As indicated by the R output, the p-values for both Model 2 and Model 3 residuals exceed 0.05, satisfying the test criteria and leading to the conclusion that the residuals conform to a Gaussian White Noise distribution.

4.3 Kolmogorov-Smirnov Test

Figure 4.3: Kolmogorov-Smirnov Test



Based on the results from the R output, it can be inferred that both Model 2 and Model 3 successfully pass the Kolmogorov-Smirnov test, providing evidence that the residuals adhere to a Gaussian White Noise distribution.

5. Forecasting

In Section 3, during the model development phase, diagnostic checking revealed that both Model 2 and Model 3 meet the necessary criteria, indicating their suitability for accurate forecasting.

Model 2:

Figure 5.1: Forecasting on transformed testing set

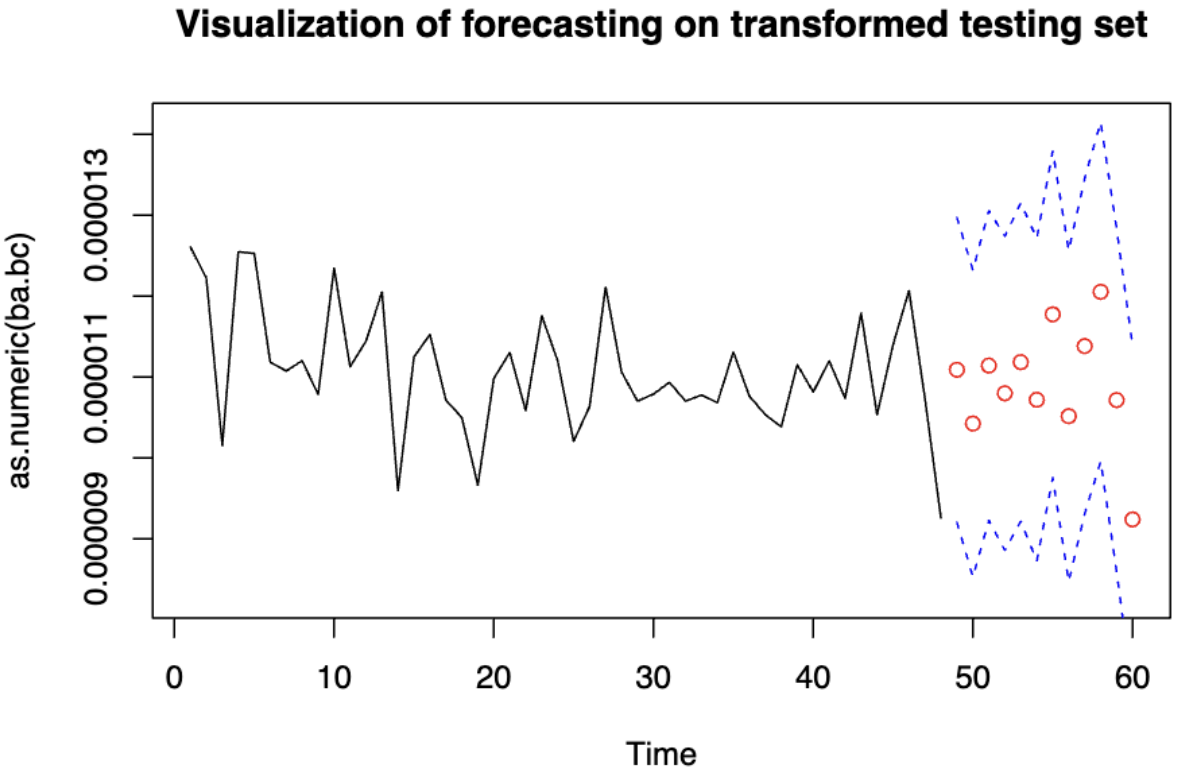


Figure 5.2: Forecasting on true value of testing set

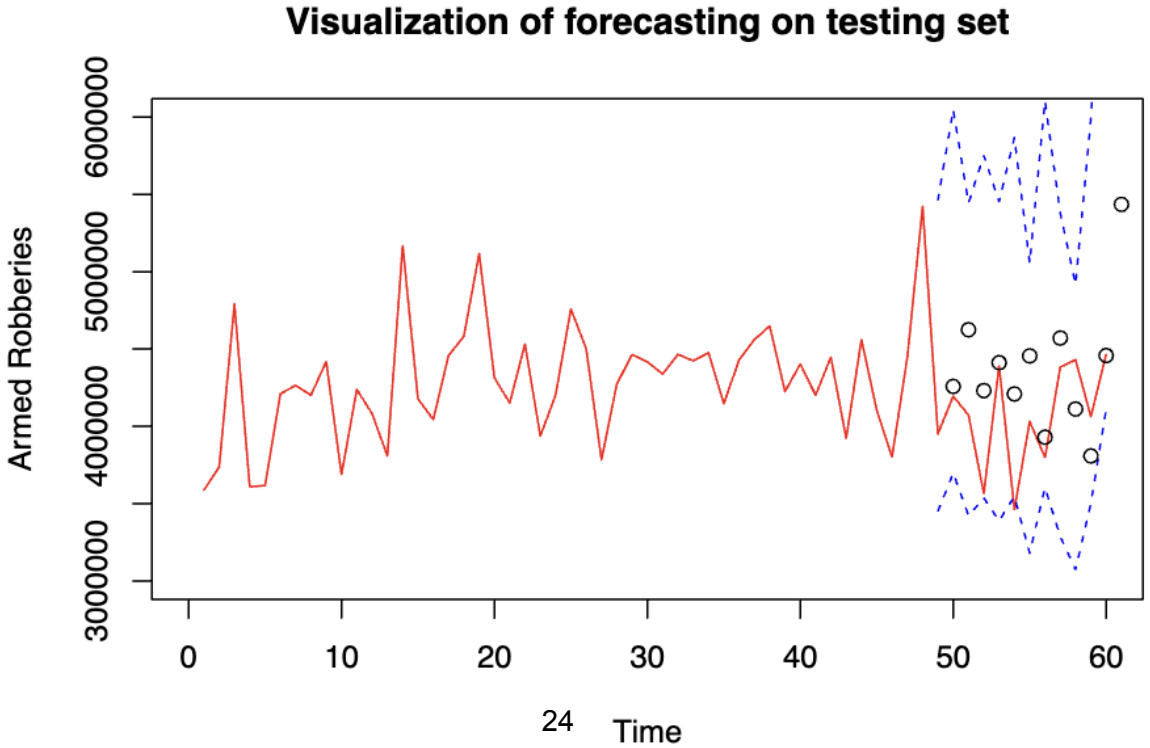
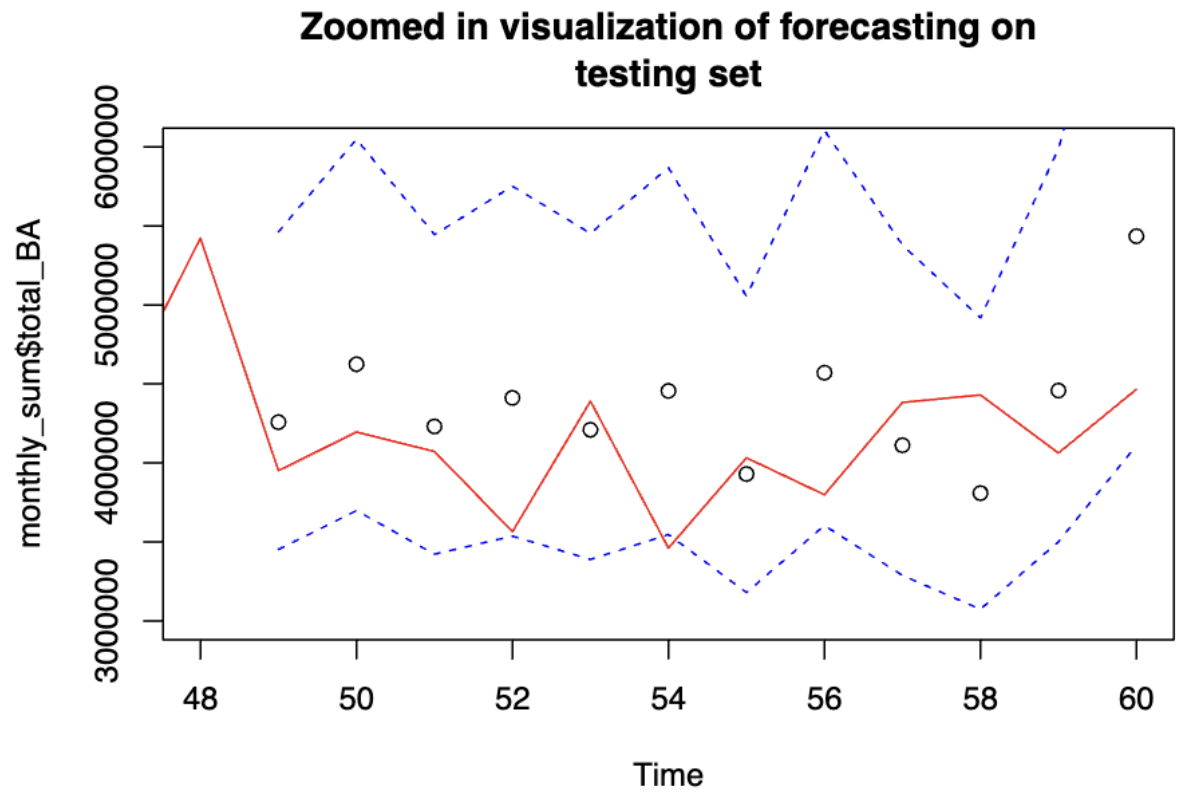


Figure 5.3: Zoomed in Forecasting on true value of testing set



Model 3:

Figure 5.4: Forecasting on transformed testing set

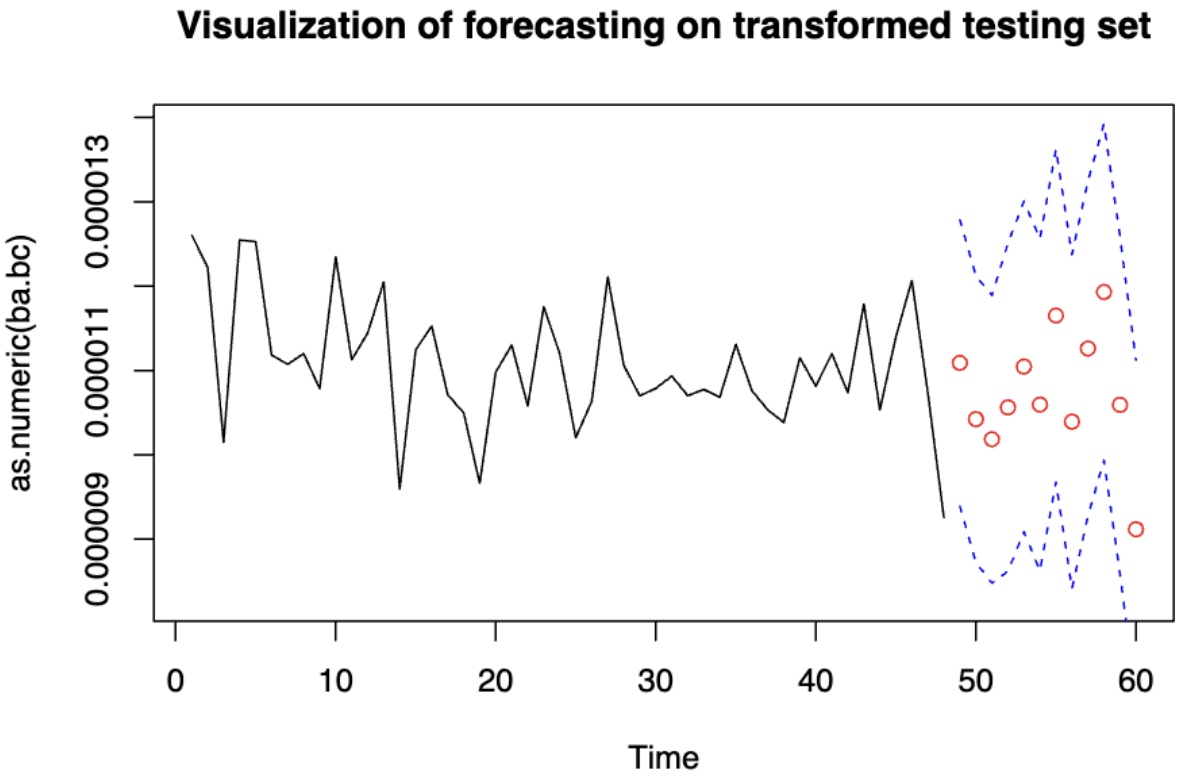


Figure 5.5: Forecasting on true value of testing set

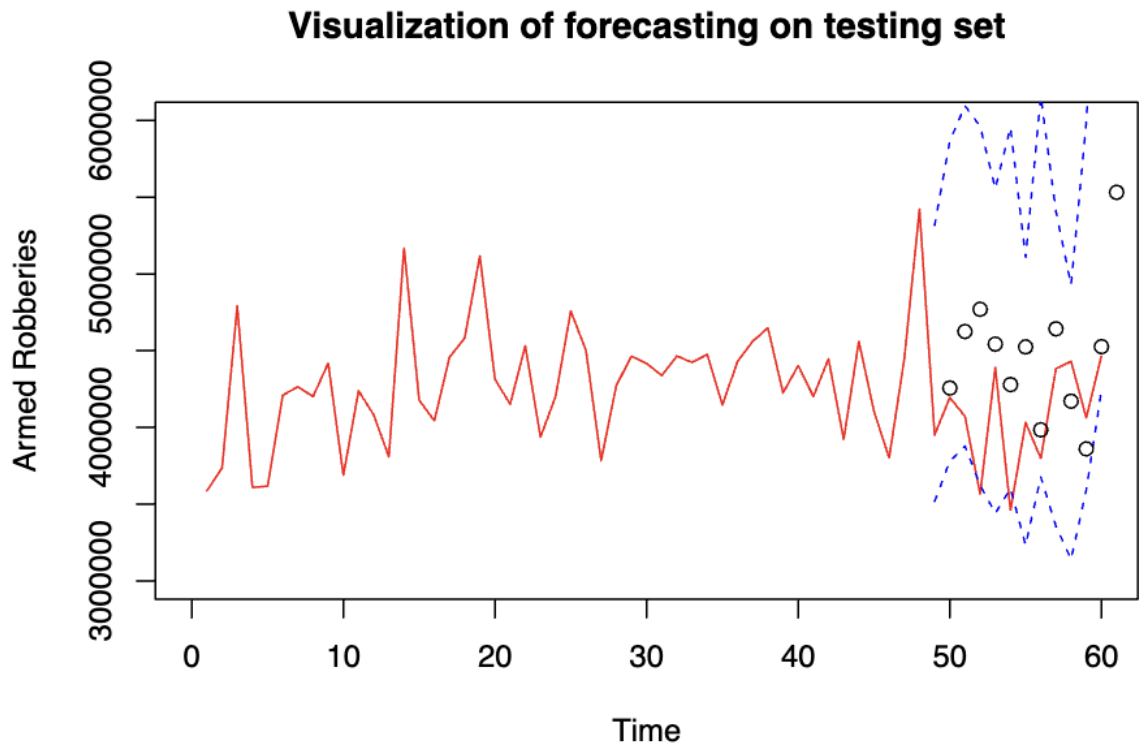
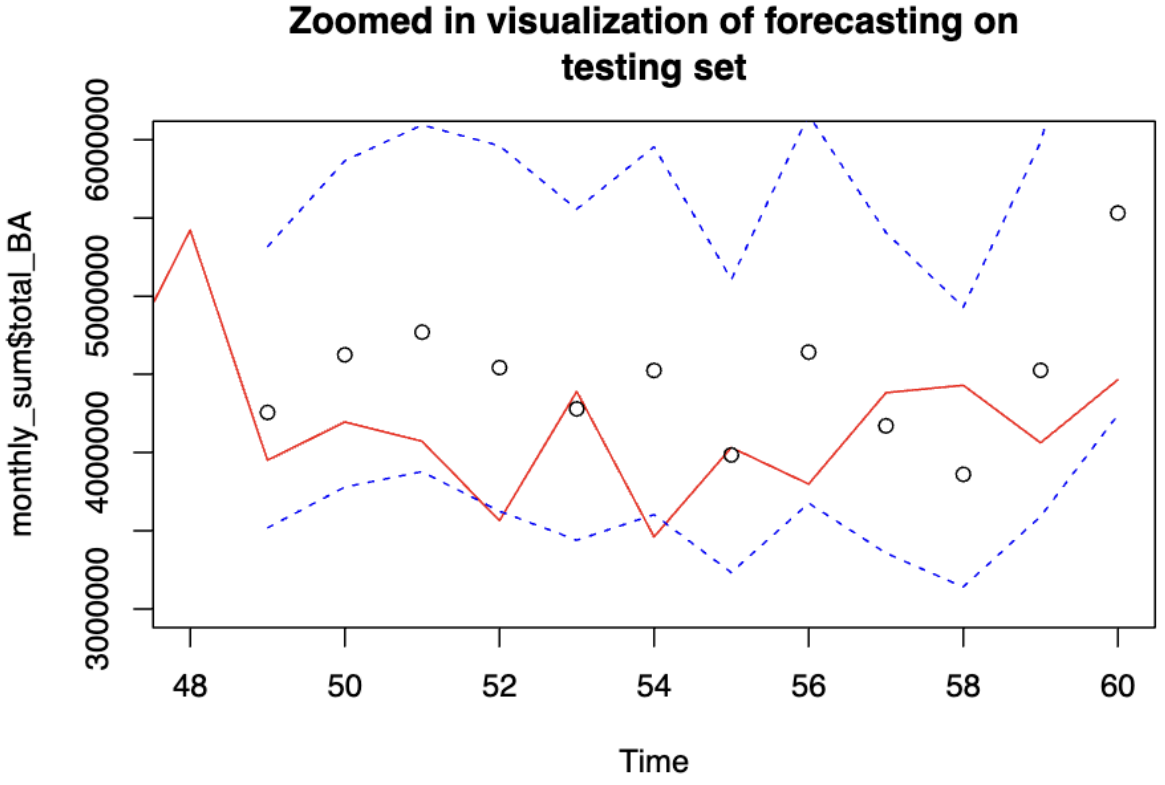


Figure 5.6: Zoomed in Forecasting on true value of testing set



Based on the visual inspection of the plots, it is evident that the forecasted values for both Model 2 and Model 3 fall within the confidence interval. Upon closer examination, Model 2 exhibits a slightly better fit than Model 3. This observation aligns with the quantitative analysis in Section 3.4, where the AICc values confirm that Model 3 boasts the lowest value, establishing it as the superior choice for forecasting. However, after the visualization, choose **model 2** to be the optimal model for forecasting.

6. Conclusion

In conclusion, this comprehensive time series analysis project aimed at transforming and modeling the non-stationary billing amount data to achieve accurate forecasting. The key steps involved Box-Cox transformation, log-transformation, and subsequent differencing at lags 1 and 12 to eliminate trends and seasonality.

The exploration began with a Box-Cox transformation, which indicated the need for a log-transformation due to a lambda value of -0.7474747. However, the log-transformed data revealed right-skewness, prompting the application of Box-Cox transformation again. Following this, differencing at lags 1 and 12 successfully removed trends and seasonality, maintaining an acceptable variance.

The identification of potential models was based on ACF and PACF plots, leading to the selection of three SARIMA models:

SARIMA(1, 1, 1) x (0, 1, 0) s=12,

SARIMA(1, 1, 2) x (0, 1, 0) s=12,

SARIMA(1, 1, 4) x (0, 1, 0) s=12.

Rigorous diagnostic checks were performed, ultimately validating models 2 and 3.

In the model selection stage, the analysis of AICc values highlighted that SARIMA(1, 1, 2) x (0, 1, 0) s=12 (model 3) exhibited the lowest AICc value, signifying its superior fit for forecasting purposes.

The culmination of the project involved forecasting the billing amount using the chosen SARIMA(1, 1, 2) x (0, 1, 0) s=12 model. The forecasted values were visualized in Figure 5.1 and Figure 5.2, providing a clear representation of the predicted transformed and original billing amounts.

In summary, this project successfully navigated through data transformation, differencing, model identification, and diagnostic checks to determine the most fitting SARIMA model for forecasting billing amounts. The results underscore the effectiveness of SARIMA(1, 1, 2) x (0, 1, 0) s=12 in capturing the underlying patterns in the time series data and generating reliable forecasts for future billing amounts.

Figure 5.1: Forecast value

```
0.000011090167 0.000010423095 0.000010184768
0.000010562946 0.000011046726 0.000010594617
0.000011651726 0.000010393669 0.000011260049
0.000011930639 0.000010592604 0.000009116882
```

Figure 5.2: Forecast value

```
4256664 4625008 4770367 4543271
4279073 4525110 3984463 4642534
4170966 3860340 4526261 5532332
```

Without the invaluable assistance from Professor Raya Feldman and TA Li Hao, the completion of this project would not have been possible. I extend my sincere gratitude to them for dedicating their time, providing guidance, engaging in discussions, and offering valuable insights throughout the project. Their expertise and support significantly contributed to the successful execution of the analysis, ensuring a thorough understanding of the complex time series data and its nuances. This collaborative effort has not only enhanced my technical skills but has also enriched the overall learning experience. I am truly grateful for their mentorship and encouragement.

7. Reference

Patil, P. (2023). *Healthcare Dataset [Dataset]*.

https://www.kaggle.com/datasets/prasad22/healthcare-dataset/data

All the UCSB PSTAT 174/274 lab materials and Lecture slides

8. Appendix

8.1 Data Dictionary

Variable	Type	Summary of the Variables
Name	Character	This column represents the name of the patient associated with the healthcare record.
Age	Character	The age of the patient at tl time of admission, expresse in years.
Gender	Character	Indicates the gender of the patient, either “Male” or “Female.”
Blood Type	Character	The patient’s blood type, which can be one of the common blood types (e.g., “A+”, “O-”, etc.).

Variable	Type	Summary of the Variables
Medical Condition	Character	This column specifies the primary medical condition or diagnosis associated with the patient, such as "Diabetes," "Hypertension," "Asthma," and more.
Date of Admission	Date	The date on which the patient was admitted to the healthcare facility.
Doctor	Character	The name of the doctor responsible for the patient's care during their admission.
Hospital	Character	Identifies the healthcare facility or hospital where the patient was admitted.

Variable	Type	Summary of the Variables
Insurance Provider	Character	This column indicates the patient's insurance provider, which can be one of several options, including "Aetna," "Blue Cross," "Cigna," "UnitedHealthcare," and "Medicare."
Billing Amount	Numeric	The amount of money billed for the patient's healthcare services during their admission. This is expressed as a floating-point number.
Room Number	Numeric	The room number where the patient was accommodated during their admission.

Variable	Type	Summary of the Variables
Admission Type	Character	Specifies the type of admission, which can be "Emergency," "Elective," or "Urgent," reflecting the circumstances of the admission.
Discharge Date	Date	The date on which the patient was discharged from the healthcare facility, based on the admission date and a random number of days within a realistic range.

Variable	Type	Summary of the Variables
Medication	Character	Identifies a medication prescribed or administered to the patient during their admission. Examples include "Aspirin," "Ibuprofen," "Penicillin," "Paracetamol," and "Lipitor."
Test Results	Character	Describes the results of a medical test conducted during the patient's admission. Possible values include "Normal," "Abnormal," or "Inconclusive," indicating the outcome of the test.

8.2 Relevant R-code

```
plot.roots <- function(ar.roots=NULL, ma.roots=NULL, size=2, angles=FALSE,
special=NULL, sqecial=NULL,my.pch=1,first.col="blue",second.col="red",main=NULL)
{xylims <- c(-size,size)
    omegas <- seq(0,2*pi,pi/500)
    temp <- exp(complex(real=rep(0,length(omegas)),imag=omegas))
    plot(Re(temp),Im(temp),typ="l",xlab="x",ylab="y",xlim=xylims,ylim=xylims,main=main)
    abline(v=0,lty="dotted")
    abline(h=0,lty="dotted")
    if(!is.null(ar.roots))
     {
      points(Re(1/ar.roots),Im(1/ar.roots),col=first.col,pch=my.pch)
      points(Re(ar.roots),Im(ar.roots),col=second.col,pch=my.pch)
     }
    if(!is.null(ma.roots))
     {
      points(Re(1/ma.roots),Im(1/ma.roots),pch="*",cex=1.5,col=first.col)
      points(Re(ma.roots),Im(ma.roots),pch="*",cex=1.5,col=second.col)
     }
    if(angles)
     {
      if(!is.null(ar.roots))
        {
         abline(a=0,b=Im(ar.roots[1])/Re(ar.roots[1]),lty="dotted")
         abline(a=0,b=Im(ar.roots[2])/Re(ar.roots[2]),lty="dotted")
        }
      if(!is.null(ma.roots))
        {
         sapply(1:length(ma.roots), function(j)
abline(a=0,b=Im(ma.roots[j])/Re(ma.roots[j]),lty="dotted"))
        }
```

```
    }
  if(!is.null(special))
    {
      lines(Re(special),Im(special),lwd=2)
    }
  if(!is.null(sqecial))
    {
      lines(Re(sqecial),Im(sqecial),lwd=2)
    }
    }
# Loading libraries
library(readr)
library(dplyr)
library(ggplot2)
library(forecast)
library(TSA)

#Setting
options(scipen = 999)

# Loading documents
hc_data <- read_csv("healthcare_dataset.csv")

# Cleaning data
BA <- hc_data$`Billing Amount`
DOA <- hc_data$`Date of Admission`
df <- as.data.frame(DOA)
df$BA <- BA
df <- df[order(df$DOA), ][-c(1:8),]
df <- df %>%
  mutate(month = format(DOA, "%Y-%m"))
```

```
monthly_sum <- df %>%
  group_by(month) %>%
  summarise(total_BA = sum(BA))

# Check how the data look like
cat("Providing a comprehensive overview of the data")
ggplot(monthly_sum, aes(x = month, y = total_BA, group = 1)) +
  geom_line(color = "skyblue") +
  geom_point(color = "red") +
  labs(title = "Total Billing Amount Over Time", x = "Month", y = "Total Billing Amount") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Divide the data into traning set and test set
training <- monthly_sum[c(1:48),]
testing <- monthly_sum[c(49:60),]

# Convert data into time series format
ba <- ts(training$total_BA, start =c(2018, 11), frequency = 12)

# Checking main features
cat("Inspecting key characteristics")
plot(decompose(ba))
library(MASS)
t <- 1:length(ba)
fit <- lm(ba ~ t)
cat("Performing Box-Cox Transformation")
bcTrans <- boxcox(ba ~ t, plotit = TRUE, lambda = seq(-4, 3))

lambda <- bcTrans$x[which(bcTrans$y == max(bcTrans$y))]
ba.bc <- ba^lambda
```

```
plot(decompose(ba.bc))
var(ba.bc)

ba.bc_lag1 <- diff(ba.bc, lag = 1)
plot(decompose(ba.bc_lag1))
var(ba.bc_lag1)

ba.bc_lag12 <- diff(ba.bc_lag1, lag = 12)
plot(decompose(ba.bc_lag12))
var(ba.bc_lag12)

par(mfrow = c(1,2))
acf(ba.bc_lag12, lag.max = 60)
pacf(ba.bc_lag12, lag.max = 60)

fit1 <- arima(as.numeric(ba.bc), order = c(1, 1, 1), seasonal = list(order = c(0, 1, 0), period =
12), method = "ML")
fit1

fit2 <- arima(as.numeric(ba.bc), order = c(1, 1, 2), seasonal = list(order = c(0, 1, 0), period =
12), method = "ML")
fit2
plot.roots(NULL, polyroot(c(1, -1.2141, 0.3685)))

fit3 <- arima(as.numeric(ba.bc), order = c(1, 1, 4), seasonal = list(order = c(0, 1, 0), period =
12), method = "ML")
fit3
plot.roots(NULL, polyroot(c(1, 1.544, 0.0615, 0.6532, -0.4808)))

cat("Checking residuals")
res <- residuals(fit1)
```

```
par(mfrow = c(2,2))
hist(res, col = "blue", xlab = "", prob = TRUE, main = "Histogram of residuals of model 1")
m <- mean(res)
std <- sqrt(var(res))
curve(dnorm(x, m, std), add = TRUE)
plot.ts(res,ylab= "residuals of model 1",main="Residuals plot of model 1")
fitt <- lm(res ~ as.numeric(1:length(res)))
abline(fitt, col="red")
abline(h=mean(res), col="blue")
qqnorm(res,main= "Normal Q-Q Plot for model 1")
qqline(res,col="blue")

# Shapiro test for normality
cat("Shapiro test")
shapiro.test(res)

#Box-Pierce test
cat("Box-Pierce test")
Box.test(res, lag = 7, type = c("Box-Pierce"), fitdf = 2)

#Ljung-Box test
cat("Ljung-Box test")
Box.test(res, lag = 7, type = c("Ljung-Box"), fitdf = 2)

#McLeod-Li test
cat("McLeod-Li test")
Box.test(res^2, lag = 7, type = c("Ljung-Box"), fitdf = 0)

ar(res, aic = TRUE, order.max = NULL, method = c("yule-walker"))

par(mfrow = c(1,2))
```

```
acf(res, lag.max = 60, main = "ACF of the residuals of model 1")
pacf(res, lag.max = 60, main = "PACF of the residuals of model 1")

cat("Checking residuals")
res2 <- residuals(fit2)
par(mfrow = c(2,2))
hist(res, col = "blue", xlab = "", prob = TRUE, main = "Histogram of residuals of model 2")
m <- mean(res)
std <- sqrt(var(res))
curve(dnorm(x, m, std), add = TRUE)
plot.ts(res,ylab= "residuals of model 2",main="Residuals plot of model 2")
fitt <- lm(res ~ as.numeric(1:length(res)))
abline(fitt, col="red")
abline(h=mean(res), col="blue")
qqnorm(res,main= "Normal Q-Q Plot for model 2")
qqline(res,col="blue")

# Shapiro test for normality
cat("Shapiro test")
shapiro.test(res)

#Box-Pierce test
cat("Box-Pierce test")
Box.test(res, lag = 7, type = c("Box-Pierce"), fitdf = 3)

#Ljung-Box test
cat("Ljung-Box test")
Box.test(res, lag = 7, type = c("Ljung-Box"), fitdf = 3)

#McLeod-Li test
cat("McLeod-Li test")
```

```
Box.test(res^2, lag = 7, type = c("Ljung-Box"), fitdf = 0)

ar(res, aic = TRUE, order.max = NULL, method = c("yule-walker"))

par(mfrow = c(1,2))
acf(res, lag.max = 60, main = "ACF of the residuals of model 2")
pacf(res, lag.max = 60, main = "PACF of the residuals of model 2")

cat("Checking residuals")
res3 <- residuals(fit3)
par(mfrow = c(2,2))
hist(res, col = "blue", xlab = "", prob = TRUE, main = "Histogram of residuals of model 3")
m <- mean(res)
std <- sqrt(var(res))
curve(dnorm(x, m, std), add = TRUE)
plot.ts(res,ylab= "residuals of model 3",main="Residuals plot of model 3")
fitt <- lm(res ~ as.numeric(1:length(res)))
abline(fitt, col="red")
abline(h=mean(res), col="blue")
qqnorm(res,main= "Normal Q-Q Plot for model 3")
qqline(res,col="blue")

# Shapiro test for normality
cat("Shapiro test")
shapiro.test(res)

#Box-Pierce test
cat("Box-Pierce test")
Box.test(res, lag = 7, type = c("Box-Pierce"), fitdf = 5)

#Ljung-Box test
```

```
cat("Ljung-Box test")
Box.test(res, lag = 7, type = c("Ljung-Box"), fitdf = 5)

#McLeod-Li test
cat("McLeod-Li test")
Box.test(res^2, lag = 7, type = c("Ljung-Box"), fitdf = 0)

ar(res, aic = TRUE, order.max = NULL, method = c("yule-walker"))

par(mfrow = c(1,2))
acf(res, lag.max = 60, main = "ACF of the residuals of model 3")
pacf(res, lag.max = 60, main = "PACF of the residuals of model 3")

periodogram(training$total_BA, main = "Periodogram")
detach(package:TSA, unload = TRUE)
library(GeneCycle)
fisher.g.test(res2)
fisher.g.test(res3)
cpgram(res2,main="K-S test for model 2")
cpgram(res3,main="K-S test for model 3")

pred.tr <- predict(fit2, n.ahead = 12)
U.tr= pred.tr$pred + 2*pred.tr$se
L.tr= pred.tr$pred - 2*pred.tr$se
ts.plot(as.numeric(ba.bc), xlim=c(1,length(ba.bc)+12), ylim =
c(min(ba.bc)-0.000001,max(U.tr)), main = "Visualization of forecasting on transformed
testing set")
lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")
points((length(ba.bc)+1):(length(ba.bc)+12), pred.tr$pred, col="red")
```

```
pred.orig <- (pred.tr$pred)^(1/lambda)
U = (U.tr)^(1/lambda)
L = (L.tr)^(1/lambda)
ts.plot(monthly_sum$total_BA,col="red",
ylab="Armed Robberies",main="Visualization of forecasting on testing set", xlim = c(0, 60),
ylim = c(3000000, 6000000))
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(training)+48):(length(training)+59), pred.orig, col="black")
ts.plot(monthly_sum$total_BA,col="red", xlim = c(48, length(ba)+12), ylim = c(3000000,
6000000), main = "Zoomed in visualization of forecasting on
testing set")
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(ba)+1):(length(ba)+12), pred.orig, col="black")

pred.tr <- predict(fit3, n.ahead = 12)
U.tr= pred.tr$pred + 2*pred.tr$se
L.tr= pred.tr$pred - 2*pred.tr$se
ts.plot(as.numeric(ba.bc), xlim=c(1,length(ba.bc)+12), ylim =
c(min(ba.bc)-0.000001,max(U.tr)),  main = "Visualization of forecasting on transformed
testing set")
lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")
points((length(ba.bc)+1):(length(ba.bc)+12), pred.tr$pred, col="red")

pred.orig <- (pred.tr$pred)^(1/lambda)
U = (U.tr)^(1/lambda)
L = (L.tr)^(1/lambda)
ts.plot(monthly_sum$total_BA,col="red",
```

```
ylab="Armed Robberies",main="Visualization of forecasting on testing set", xlim = c(0, 60),
ylim = c(3000000, 6000000))
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(training)+48):(length(training)+59), pred.orig, col="black")
ts.plot(monthly_sum$total_BA,col="red", xlim = c(48, length(ba)+12), ylim = c(3000000,
6000000), main = "Zoomed in visualization of forecasting on
testing set")
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(ba)+1):(length(ba)+12), pred.orig, col="black")
```