Forecasting U.S. Monthly Housing Sales Using SARIMA Models: A Box-Jenkins Approach

Frederick Kiessling

2025-06-12

Abstract

i) This project applies the Box-Jenkins methodology to forecast U.S. monthly housing sales from 1965–1975 using SARIMA models. The data exhibits strong seasonality and trend, addressed via log transformation and differencing at lags 12 and 1 to induce stationarity. After narrowing the model space using ACF/PACF interpretation and filtering for valid root structures, I conducted a grid search ranked by AICc and residual diagnostics. Candidate 1 (SARIMA(1,1,0)(2,1,0)) achieved the lowest AICc but failed the Shapiro-Wilk test at the 5% level (passing at 1%) and captured only 4 of 12 test values within the 95% confidence band. Candidate 2 (SARIMA(1,1,1)(1,1,0)) had a slightly higher AICc but passed all diagnostics and captured 7 of 12 test values, making it the more reliable model for forecasting.

i) Introduction

i) The goal of this project is to forecast U.S. monthly housing sales from 1965–1975 using SARIMA models. The dataset, sourced from the Time Series Data Library (TSDL), contains 132 monthly observations and is of interest to me because I would like to determine what the best model is for forecasting real-world economic time series that exhibit both trend and seasonality, using statistically sound model selection and validation procedures. After transforming and differencing the data to induce stationarity, I applied the Box-Jenkins methodology to identify and evaluate candidate models. I reduced the SARIMA model space by interpreting the ACF and PACF of the stationary series, then applied a grid search algorithm to evaluate models across the reduced set. Models with parameters like sma1 *approx* -1 were eliminated early, as such values indicate near non-invertibility, which can lead to unstable forecasts. From the models with valid roots (i.e., all AR and MA roots outside the unit circle), I re-ranked by AICc and selected top candidates for further diagnostic testing, ultimately comparing them based on statistical fit and forecasting performance. I also evaluated the Shapiro-Wilk p-values of these invertible models and used both AICc and normality results to help determine the most statistically sound candidates for final comparison.

ii) Two models were selected for comparison. Candidate 1 (SARIMA(1,1,0)(2,1,0)) had the lowest AICc and passed all residual diagnostics except the Shapiro-Wilk test at the 5% level. However, it only captured 4 of 12 test values within the 95% forecast interval. Candidate 2 (SARIMA(1,1,1)(1,1,0)) had a higher AICc but passed all diagnostic checks and captured 7 of 12 forecast values. While Candidate 1 fit the training data better, Candidate 2 provided more accurate forecasts, making it the preferred model for forecasting. All analysis was conducted in R using the forecast and stats packages.

i)

The dataset contained 132 monthly observations from 1965 to 1975. I divided the data into a training set consisting of the first 120 months (1965–1974) and a test set of the final 12 months (1975). The training set was used for model selection and fitting, while the test set was held out for out-of-sample forecast evaluation.

ii)

See Figure 1 and 5 for referenced visualization.

i) In my immediate observation of the plot of the training data there is a clear upward trend to be seen. Furthermore, in the plot for the Decomposition of the log-transformed training data we can see a slightly increasing trend, however it may not be a strictly linear trend. To further confirm, I fitted a linear regression of time on the training data and observed a p-value of 0.0035, indicating a statistically significant upward trend over time. The ACF of the training data also shows a gradual decay over the first few lags, which suggests some level of trend.

ii)

i) Immediate observation included that there is seasonality to be seen in the plot of the training data. I explored this further by printing recurring annual patterns, looking and minimums and maximums of each year: peaks often occur between March and August, while troughs consistently appear in December confirming strong seasonal behavior. Also, the ACF and PACF of the 1st differenced training data shows very clear seasonality.

iii)

i) There appears to be non-constant variance, I calculated the variance of the training data at various intervals from which it increased significantly.

iii) Transformations:

See Figure 2–5 for visualization.

i) Histogram of training data appears to be symmetrical. However, the variance is large: 111.898. And, there seems to be non-constant variance, which I confirmed visually and by measuring variance at different intervals. The ACFs also remain large and periodic. In order to stabilize and lower the variance a transformation is needed. To remove seasonality and trend differencing is appropriate. I applied a Box-Cox transformation to estimate an optimal λ for stabilizing the variance. The estimated λ was approximately 0.72, suggesting a mild power transformation. The original training set had a high variance of 111.898, which was reduced to 12.998 after Box-Cox transformation. However, a simple log transformation (i.e., $\lambda = 0$) reduced the variance even further to 0.058, while maintaining a similar histogram shape and overall distribution. Since both histograms appeared approximately symmetric and the log transform performed better at variance stabilization, I proceeded with the log transformation for modeling. I also picked it for simplicity and interpretability. Because the decomposition of

See Figure 6–8 for visualization.

ii) Visually, I interpreted that the log transformed data had trend, seasonality, and a variance of 0.05825466. I differenced at lag 12, and Seasonality was no longer apparent, trend is insignificant, and received a variance of 0.05563058. Lastly, I differenced at lag 1 and variance decreased further to 0.01158268, while there was also no seasonality nor trend apparent.

See Figure 9–11 for Histogram Comparison.

iii) Next, I checked the ACF of the transformed series to further investigate stationarity. The ACF of the log-transformed data showed slow decay and strong seasonal spikes, indicating non-stationarity. After differencing at lag 12, the seasonal component disappeared, but the ACF still decayed slowly. Finally, after an additional first-order differencing (lag 1), the ACF showed rapid decay, consistent with a stationary process.

See Figure 12–14 for visualization.

iv) I concluded that the most appropriate series for modeling is the log-transformed data that has been both seasonally differenced at lag 12 and non-seasonally differenced at lag 1. I compared histograms of the log-transformed data and the log-transformed data that was both seasonally and non-seasonally differenced; the latter appeared symmetric and nearly Gaussian, indicating approximate normality.

iv) Plotting and Analyzing the ACF/PACF

See Figure 15 for visualization.

For seasonality terms, in the ACF, I observed a significant spike at lag 12, therefore I set Q to 1. In the PACF, I observed Lags 12 and 24 are significant, so P was set to 1 or 2.

For non-seasonal terms: In the ACF of the stationary series, lags 1–11 fall within the confidence bounds, suggesting a low non-seasonal MA order so I chose q =0.

The first significant spike occurred at lag 9, suggesting a possible AR(9) model. For the non-seasonal part, instead of fitting an AR(9) model, I rather fit a ARMA(1,1) model because there are less parameters making the model parsimonious.

The list of candidate models:

SARIMA of log-transformed data: $s = 12,\ D = 1,\ d = 1,\ p \in \{0,1\},\ q \in \{0,1\},\ P \in \{0,1,2\},\ Q \in \{0,1\}$

SARIMA of log-transformed data: s=12, D=1, d=1, p=0 or 1, q= 0 or 1, and P =0,1, or 2, and Q = 0 or 1. I choose not to model the higher order model AR(9) because the spike at lag 9 was isolated, and including such a high-order term increases model complexity without strong justification from the overall PACF pattern. I also limited p to 0 or 1 to be consistent with the Box-Jenkins methodology.

v) Fitting Model & Diagnostics

My approach to find the best model followed Box Jenkins approach: I iteratively checked through the candidate models ranked by AICc. Then I checked fitted models' coefficients, and filtered out those with near-non-invertible values (e.g., sma1 close to –1), because such values indicate that the model is approaching the boundary of invertibility, which can lead to unstable or unreliable forecasts and inflated standard errors. For other candidates, that I did not dismiss directly, I used the plot.roots function to visually check for stationarity and invertibility in the models. For models that were invertible, I ranked again by the AIC, and Shapiro-Wilks. Then I performed diagnostic tests.

Q: Is the model obtained by using AICc the same as one of the models suggested by ACF/PACF? Yes, the model obtained by using AICc is the same as the model suggest by ACF/PACF, as my candidate model 1 has the lowest AIC.

These were the top models that followed from this approach:

Candidate Models:

- **Candidate 1:** SARIMA(1,1,0)(2,1,0), AICc: -201.463 (13th best AICc)
- **Candidate 2:** SARIMA(1,1,1)(1,1,0), AICc: -190.5418 (18th best AICc)
- **Candidate 3:** SARIMA(1,1,1)(0,1,0), AICc: -170.3211 (24th best AICc)

1: SARIMA(1,1,0)(2,1,0) AICc: -201.463

```
Call:
arima(x = log.train.ts, order = c(1, 1, 0),
      seasonal = list(order = c(2, 1, 0), period = 12),
      method = "ML")

Coefficients:
```

```
          ar1     sar1     sar2
      -0.2123  -0.6278  -0.3777
s.e.   0.0979   0.0988   0.0981

sigma^2 estimated as 0.007762:
log likelihood = 104.83,  aic = -201.67
```

2: SARIMA(1,1,1)(1,1,0) AICc: -190.5418

```
Call:
arima(x = log.train.ts, order = c(1, 1, 1),
      seasonal = list(order = c(1, 1, 0), period = 12),
      method = "ML")

Coefficients:
          ar1     ma1     sar1
      -0.8409  0.6901  -0.4660
s.e.   0.1761  0.2287   0.0895

sigma^2 estimated as 0.008887:
log likelihood = 99.37,  aic = -190.75
```

3: SARIMA(1,1,1)(0,1,0) AICc: -170.3211

```
Call:
arima(x = log.train.ts, order = c(1, 1, 1),
      seasonal = list(order = c(0, 1, 0), period = 12),
      method = "ML")

Coefficients:
          ar1     ma1
      -0.7915  0.7046
s.e.   0.2995  0.3447

sigma^2 estimated as 0.01125:
log likelihood = 88.21,  aic = -170.42
```

Before proceeding, I decided to drop candidate model 3 as it has significantly higher AICc and because it failed the White Noise Test.

Both models are stationary because all SAR/AR roots lie outside the unit circle. Model 1 (SARIMA(1,1,0)(2,1,0)[12]) is stationary because all AR/SAR roots (non-seasonal and seasonal) lie outside the unit circle.

Model 2 (SARIMA(1,1,1)(1,1,0)[12]) is stationary because all AR/SAR roots lie outside the unit circle, and invertible because the non-seasonal MA root also lies outside the unit circle.

Via Figure 16 and 17: I use the plot.roots function to visualize this as well.

Test Name	**p-value**	**Pass?**
Shapiro-Wilk Test	0.03496	No
Box-Pierce Test	0.2726	Yes
Ljung-Box Test	0.2274	Yes
McLeod-Li Test (Ljung-Box on res^2)	0.4305	Yes
AR Fit Order (White Noise)	Order = 0	Yes

Table 1: Diagnostic Test Results for Candidate Model 1: SARIMA(1,1,0)(2,1,0)

*Note: Shapiro-Wilks passes at 1% significance interval

Test Name	p-value / Result	Pass?
Shapiro-Wilk Test	0.2424	Yes
Box-Pierce Test	0.2037	Yes
Ljung-Box Test	0.1622	Yes
McLeod-Li Test (Ljung-Box on res^2)	0.07476	Yes
AR Fit Order (White Noise)	Order = 0	Yes

Table 2: Diagnostic Test Results for Candidate Model 2: SARIMA(1,1,1)(1,1,0)

Diagnostics Plotting for Model 1 and 3

For both SARIMA models, in the Diagnostic Plots we can see there is no trend, no visible change of variance, no seasonality. The histogram and Q-Q plot indicate that the residuals are approximately normally distributed.

via Figure 17 and 18

The drawback to Candidate model 2 SARIMA(1,1,1)(1,1,0) is that the ACF and PACF of the residuals plot show a significant spike at lag 23.

Given Table 1 and 2: Explanation of Chosen Model:

Candidate 1 (SARIMA(1,1,0)(2,1,0)) had the lower AICc and passed all diagnostic checks except the Shapiro-Wilk test at the 5% level, though it did pass at the 1% level. Its residuals showed no signs of autocorrelation, non-constant variance, or structural issues, making it a strong in-sample model. However, it only captured 4 of 12 test points within the 95% forecast confidence interval, indicating weaker out-of-sample predictive accuracy. Candidate 2 (SARIMA(1,1,1)(1,1,0)) had a higher AICc but passed all diagnostic tests and achieved better forecasting, with 7 of 12 test points falling within the confidence interval. While it showed a slight residual spike in the ACF plot, its superior forecast performance makes it the more practical choice for real-world use.

In conclusion, given that the primary goal of this project is to produce accurate forecasts, Candidate 2 (SARIMA(1,1,1)(1,1,0)) is the preferred model. While Candidate 1 (SARIMA(1,1,0)(2,1,0)) had the lowest AICc and passed all residual diagnostics except Shapiro-Wilk at the 5% level, it performed poorly on the test set, capturing only 4 of 12 points within its 95% confidence interval. In contrast, Candidate 2 had slightly higher AICc but passed all diagnostic checks and captured 7 of 12 forecast points within the confidence band, demonstrating better predictive accuracy. Thus, although Candidate 1 provides a better in-sample fit (lower AIC =-201.463), as we see in Forecasting section, next, Candidate 2 offers more reliable out-of-sample forecasting performance. Given that the goal for this project is to forecast I would choose candidate 2.

2 Candidate Models in Algebraic Form:

Candidate 1: $(1 + 0.2123B)(1 + 0.6278B^{12} + 0.3777B^{24}) \quad \nabla_1\nabla_{12}\log(U_t) = Z_t, \quad \hat{\sigma}^2 = 0.00776$

Candidate 2: $(1 + 0.8409B)(1 + 0.446B^{12}) \quad \nabla_1\nabla_{12}\log(U_t) = (1 + 0.6901B)Z_t, \quad \hat{\sigma}^2 = 0.01125$

Given that the goal for this project is to forecast I would choose candidate 2.

vi) Forecasting

(Note* Plots only show on Log data but raw data is included in the Code)

Via Figure 20 and 21

In both plots, we can see 3/12 forecasted values are close to the observed beta (red line). This could be explained by what we observed earlier that the trend may not be strictly linear. However, Candidate 2

performs better overall than Candidate 1, as its 95% confidence interval contains 7 out of 12 actual values, compared to just 4 out of 12 for Candidate 1. This indicates that Candidate 2 provides more reliable interval coverage despite its tendency to underfit later observations.

Conclusion:

The goal of this project was to identify a SARIMA model capable of accurately forecasting U.S. monthly housing sales from 1965 to 1975. After narrowing the model space using ACF/PACF plots and transforming the data to achieve stationarity, I evaluated candidate models based on AICc, residual diagnostics, and forecast accuracy.

Candidate 1 (SARIMA(1,1,0)(2,1,0)) had the lowest AICc and passed all diagnostics except the Shapiro-Wilk normality test at the 5% level (though it passed at 1%). However, it captured only 3 out of 12 test values within the 95% prediction interval. Candidate 2 (SARIMA(1,1,1)(1,1,0)) passed all diagnostic checks and its forecasting band captures 7 out of 12 test values. However, residuals showed a spike at lag 23 in the ACF/PACF plot of residuals.

Given that the primary goal was forecasting accuracy, Candidate 2 is the more suitable model. All modeling and analysis were performed in R using the forecast, qpcR, and tsdl libraries.

vi) References

- Lecture 15 PSTAT 174 Slides
- Code from Professor
- PSTAT 174 Lab


```
knitr::opts_chunk$set(echo = TRUE)

library(tsdl)
library(forecast)

#Monthly U.S. house sales (1965-1975)
house.ts <- tsdl[[6]]
length(house.ts <- tsdl[[6]])

par(mfrow = c(1, 2))

plot.ts(house.ts, main = "Raw Data: Monthly U.S. House Sales")
fit <- lm(house.ts ~ time(house.ts))
plot.ts(house.ts, main = "Raw Data: U.S. House Sales (1965-1975)", ylab = "Thousands of Houses", xlab =
abline(fit, col = "red")
abline(h = mean(house.ts), col = "blue")
tsdat <- ts(house.ts, start = c(1965, 1), end = c(1975, 12), frequency = 12)
ts.plot(tsdat, main = "Raw Data (with Time Index)", ylab = "Thousands of Houses")
train.ts <- window(house.ts, end = c(1974, 12))
length(train.ts)
test.ts  <- window(house.ts, start = c(1975, 1))
length(test.ts)
par(mfrow = c(1, 2))
fit.train <- lm(train.ts ~ time(train.ts))
plot.ts(train.ts, main = "Training Set: 1965-1974", ylab = "Thousands of Houses")
abline(fit.train, col = "red")
abline(h = mean(train.ts), col = "blue")
par(mfrow = c(1, 2))

hist(train.ts, col = "blue", xlab = "", main = "Histogram of Training Set")
```

```r
acf(train.ts, lag.max = 40, main = "ACF of the Training Data")
resid.train <- residuals(fit.train)
par(mfrow = c(1, 2))
plot(resid.train, type = "l", main = "Residuals from Trend Fit", ylab = "Residuals")
abline(h = 0, col = "blue")
mean(resid.train)
shapiro.test(resid.train)
diffs <- diff(train.ts)
which(abs(diffs) > mean(abs(diffs)) + 2 * sd(diffs))
fit <- lm(train.ts ~ time(train.ts))
summary(fit)
for (i in seq(1, 120, by = 12)) {
  segment <- train.ts[i:(i + 11)]
  year_num <- (i - 1) / 12 + 1

  cat(sprintf("Year %d (Months %d-%d):\n", year_num, i, i + 11))
  cat(sprintf("Min = %.1f (Month %d)\n", min(segment), which.min(segment)))
  cat(sprintf("Max = %.1f (Month %d)\n\n", max(segment), which.max(segment)))
}
par(mfrow = c(2, 2))
plot(train.ts[1:12], main = "Months 1-12")
plot(train.ts[13:24], main = "Months 13-24")
plot(train.ts[25:36], main = "Months 25-36")
plot(train.ts[37:48], main = "Months 37-48")
plot(train.ts[49:60], main = "Months 49-60")
plot(train.ts[61:72], main = "Months 61-72")
plot(train.ts[73:84], main = "Months 73-84")
plot(train.ts[85:96], main = "Months 85-96")
plot(train.ts[97:108], main = "Months 97-108")
plot(train.ts[109:120], main = "Months 109-120")

var1 <- var(train.ts[1:40])
var2 <- var(train.ts[41:80])
var3 <- var(train.ts[81:120])
c(var1, var2, var3)  # If these are very different, variance is not constant
library(e1071)
skewness(train.ts)
library(MASS)  # For boxcox()
# Estimate optimal lambda
bcTransform <- boxcox(train.ts ~ time(train.ts), lambda = seq(-1, 1, 0.05))
lambda <- bcTransform$x[which.max(bcTransform$y)]
lambda

train.bc <- (train.ts^lambda - 1) / lambda
train.log <- log(train.ts)

# Time series plots (stacked vertically)
par(mfrow = c(1, 2))
plot.ts(train.log, main = "Log Transformed Series", ylab = "log(U_t)")
plot.ts(train.bc, main = paste("Box-Cox Transformed Series (lamda =", round(lambda, 3), ")"),
        ylab = expression((U^lambda - 1)/lambda))
# Histogram plots (stacked vertically)
hist(train.ts, col = "lightblue", xlab = "", main = "Histogram of Training Set")
```

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

hist(train.log, col = "lightblue", main = "Histogram: log(U_t)", xlab = "")
hist(train.bc, col = "lightblue", main = paste("Histogram: Box-Cox(U_t), lamda =", round(lambda, 3)), x


var(train.ts)
var(train.bc)
var(train.log)


#decompostion:

y <- ts(log.train.ts, frequency = 12)  # ensure time series with monthly seasonality
decom <- decompose(y)
plot(decom)


# Set plotting area: 3 rows, 1 column

# Log-transformed series
fit.log <- lm(log.train.ts ~ time(log.train.ts))
plot.ts(log.train.ts, main = "Log Transformed Series", ylab = "log(U_t)")
abline(fit.log, col = "red")        # Trend line
abline(h = mean(log.train.ts), col = "blue")  # Mean line
var(log.train.ts)

# Seasonal differenced log series
fit.log12 <- lm(log.train.12 ~ time(log.train.12))
plot.ts(log.train.12, main = "Seasonally Differenced Log Series", ylab = "log(U_t) - log(U_{t-12})")
abline(fit.log12, col = "red")
abline(h = mean(log.train.12), col = "blue")
var(log.train.12)

# Fully differenced stationary log series
fit.stat <- lm(log.train.stat ~ time(log.train.stat)) # differencing at lag 1
plot.ts(log.train.stat, main = "Stationary Log Series", ylab = "changelog(U_t)")
abline(fit.stat, col = "red")
abline(h = mean(log.train.stat), col = "blue")
var(log.train.stat)


# log transform ->seasonal diff -> 1st order diff
log.train.ts <- log(train.ts) # we are fitting many SARIMA models on the log version of our training da
log.train.12 <- diff(log.train.ts, lag = 12) # plot 2
log.train.stat <- diff(log.train.12, lag = 1) # plot 3


class(log.train.ts)

acf(log.train.ts)
acf(log.train.12)
acf(log.train.stat)
```

```r
knitr::opts_chunk$set(echo = TRUE)
hist(log.train.ts)
hist(log.train.stat)

hist(log.train.ts, density = 20, breaks = 20, col = "blue",
     xlab = "", prob = TRUE, main = "Histogram: log(U_t) differenced at lags 12 & 1")
m <- mean(log.train.ts)
std <- sqrt(var(log.train.ts))
curve(dnorm(x, m, std), add = TRUE)

hist(log.train.stat, density = 20, breaks = 20, col = "blue",
     xlab = "", prob = TRUE, main = "Histogram: log(U_t) differenced at lags 12 & 1")
m <- mean(log.train.stat)
std <- sqrt(var(log.train.stat))
curve(dnorm(x, m, std), add = TRUE)


# double differencing: at lag 12 and then lag 1

#note: log.train.stat is the log transform, diff seasonally and non-seasonally data

par(mfrow=c(1,2))
Acf(log.train.stat,40)
Pacf(log.train.stat,40)

#------------------------------- Explanation -------------------------------------------
# Explanation: Originally, I included Q = 2 in my option choice, this gave
# 36 possible model combinations. However, next I adjsuted it to only being Q = 1
# this reduced the number of possible models from 36 to 24.


# install.packages("qpcR")  # Run once if not installed
library(qpcR)

results <- list()
i <- 1

d <- 1
D <- 1
s <- 12

p_vals <- c(0, 1)
q_vals <- c(0, 1)
P_vals <- 0:2
Q_vals <- 0:2

for (p in p_vals) {
  for (q in q_vals) {
    for (P in P_vals) {
      for (Q in Q_vals) {
        fit <- arima(log.train.ts, order=c(p,d,q),
                     seasonal=list(order=c(P,D,Q), period=s),
                     method="ML")
```

```
        # AIC and AICc via qpcR
        AIC_val <- AIC(fit)
        AICc_val <- AICc(fit)

        results[[i]] <- list(model = c(p,d,q,P,D,Q), AICc = AICc_val, AIC = AIC_val, fit = fit)
        i <- i + 1
      }
    }
  }
}

# Sort the results by AICc
sorted <- results[order(sapply(results, function(x) x$AICc))]

# Add AICc rank as a new field to each sorted model
for (j in seq_along(sorted)) {
  sorted[[j]]$rank <- j
}

# Print sorted models with their rank and structure
for (j in seq_along(sorted)) {
  mod <- sorted[[j]]$model
  aicc <- round(sorted[[j]]$AICc, 4)
  cat(sprintf("Model %2d: (%d,%d,%d)(%d,%d,%d) AICc: %s\n",
              sorted[[j]]$rank, mod[1], mod[2], mod[3], mod[4], mod[5], mod[6], aicc))
}

#------------------------------ Explanation -------------------------------------------
# Explanation: Originally, I included Q = 2 in my option choice, this gave
# 36 possible model combinations. However, next I adjsuted it to only being Q = 1
# this reduced the number of possible models from 36 to 24. Here I apply the arima
# models with only Q = 1

# install.packages("qpcR")  # Run once if not installed
library(qpcR)

results <- list()
i <- 1

d <- 1
D <- 1
s <- 12

p_vals <- c(0, 1)
q_vals <- c(0, 1)
P_vals <- 0:2
Q_vals <- 0:1

for (p in p_vals) {
  for (q in q_vals) {
    for (P in P_vals) {
      for (Q in Q_vals) {
        fit <- arima(log.train.ts, order=c(p,d,q),
```

```
                    seasonal=list(order=c(P,D,Q), period=s),
                    method="ML")

        # AIC and AICc via qpcR
        AIC_val <- AIC(fit)
        AICc_val <- AICc(fit)

        results[[i]] <- list(model = c(p,d,q,P,D,Q), AICc = AICc_val, AIC = AIC_val, fit = fit)
        i <- i + 1
      }
    }
  }
}

# Sort the results by AICc
sorted <- results[order(sapply(results, function(x) x$AICc))]

# Add AICc rank as a new field to each sorted model
for (j in seq_along(sorted)) {
  sorted[[j]]$rank <- j
}

# Print sorted models with their rank and structure
for (j in seq_along(sorted)) {
  mod <- sorted[[j]]$model
  aicc <- round(sorted[[j]]$AICc, 4)
  cat(sprintf("Model %2d: (%d,%d,%d)(%d,%d,%d) AICc: %s\n",
              sorted[[j]]$rank, mod[1], mod[2], mod[3], mod[4], mod[5], mod[6], aicc))
}

# ----------------------------- Explanation -----------------------------------
# In my original approach I briefly tested the roots for all 36 models,
# but my new approach considers only 24 of these because I discarded 12 models that
# had Q =2 which I dismiss in my updated correct search.

# This narrowed the candidate pool to 24 models. For efficiency, I incorporated early filtering rules b
# - Invertibility violations: I excluded models where the seasonal MA coefficient (e.g., sma1) was clos
# - Root checks: For the remaining models, I examined root moduli to ensure they lie outside the unit c
# - AICc ranking and residual diagnostics were then applied only to valid models.
```

```
# Model 1: SARIMA(1,1,0)(0,1,1)
fit1 <- arima(log.train.ts,
              order = c(1,1,0),
              seasonal = list(order = c(0,1,1), period = 12),
              method = "ML")
fit1

# Conclusion: This model was discarded because the seasonal
# MA coefficient was close to -1, indicating potential invertibility issues.



# Model 2: SARIMA(1,1,1)(0,1,1)
fit2 <- arima(log.train.ts,
```

```
              order = c(1,1,1),
              seasonal = list(order = c(0,1,1), period = 12),
              method = "ML")
fit2
# Conclusion: This model was discarded because the seasonal
# MA coefficient was close to -1, indicating potential invertibility issues.

# Model 3: SARIMA(0,1,1)(0,1,1)
fit3 <- arima(log.train.ts,
              order = c(0,1,1),
              seasonal = list(order = c(0,1,1), period = 12),
              method = "ML")
fit3
# Conclusion: This model was discarded because the seasonal
# MA coefficient was close to -1, indicating potential invertibility issues.

# Note: I orginally checked this but in my new approach I discard it because it includes a
# seasonal MA term with Q = 2

# Model 4: SARIMA(1,1,1)(0,1,2)
fit4 <- arima(log.train.ts,
              order = c(1,1,1),
              seasonal = list(order = c(0,1,2), period = 12),
              method = "ML")
fit4
coefs4 <- coef(fit4)

ar_poly4 <- c(1, -coefs4["ar1"])
plot.roots(ar.roots = polyroot(ar_poly4), ma.roots = NULL, main="(A) roots of ar part, nonseasonal")

ma_poly4 <- c(1, coefs4["ma1"])
plot.roots(ar.roots = NULL, ma.roots = polyroot(ma_poly4), main="(A) roots of ma part, nonseasonal")

sma_poly4 <- c(1, coefs4["sma1"], coefs4["sma2"])
plot.roots(ar.roots = NULL, ma.roots = polyroot(sma_poly4), main="(A) roots of ma part, seasonal")


# Note: I orginally checked this but in my new approach I discard it because it includes a
# seasonal MA term with Q = 2

# Model 5: SARIMA(1,1,0)(0,1,2)
fit5 <- arima(log.train.ts,
              order = c(1,1,0),
              seasonal = list(order = c(0,1,2), period = 12),
              method = "ML")
fit5
coefs5 <- coef(fit5)

ar_poly5 <- c(1, -coefs5["ar1"])
plot.roots(ar.roots = polyroot(ar_poly5), ma.roots = NULL, main="(A) roots of ar part, nonseasonal")

sma_poly5 <- c(1, coefs5["sma1"], coefs5["sma2"])
plot.roots(ar.roots = NULL, ma.roots = polyroot(sma_poly5), main="(A) roots of ma part, seasonal")
```

```r
# Model 6: SARIMA(1,1,0)(1,1,1)
fit6 <- arima(log.train.ts,
              order = c(1,1,0),
              seasonal = list(order = c(1,1,1), period = 12),
              method = "ML")
fit6
# conclusion: dismiss bc sma1 =-1

# Model 7: SARIMA(1,1,1)(1,1,1)
fit7 <- arima(log.train.ts,
              order = c(1,1,1),
              seasonal = list(order = c(1,1,1), period = 12),
              method = "ML")
fit7
coefs7 <- coef(fit7)

ar_poly7 <- c(1, -coefs7["ar1"])
plot.roots(ar.roots = polyroot(ar_poly7), ma.roots = NULL, main="(A) roots of ar part, nonseasonal")

ma_poly7 <- c(1, coefs7["ma1"])
plot.roots(ar.roots = NULL, ma.roots = polyroot(ma_poly7), main="(A) roots of ma part, nonseasonal")

sar_poly7 <- c(1, -coefs7["sar1"])
plot.roots(ar.roots = polyroot(sar_poly7), ma.roots = NULL, main="(A) roots of ar part, seasonal")

sma_poly7 <- c(1, coefs7["sma1"])
plot.roots(ar.roots = NULL, ma.roots = polyroot(sma_poly7), main="(A) roots of ma part, seasonal")

# failed roots

# Model 8: SARIMA(1,1,0)(2,1,1)
fit8 <- arima(log.train.ts,
              order = c(1,1,0),
              seasonal = list(order = c(2,1,1), period = 12),
              method = "ML")
fit8



# Model 9: SARIMA(0,1,0)(0,1,1)
fit9 <- arima(log.train.ts,
              order = c(0,1,0),
              seasonal = list(order = c(0,1,1), period = 12),
              method = "ML")
fit9
coefs9 <- coef(fit9)

sma_poly9 <- c(1, coefs9["sma1"])
plot.roots(ar.roots = NULL, ma.roots = polyroot(sma_poly9), main="(A) roots of ma part, seasonal")

# Model 10: SARIMA(1,1,1)(2,1,1)
fit10 <- arima(log.train.ts,
               order = c(1,1,1),
```

```
              seasonal = list(order = c(2,1,1), period = 12),
              method = "ML")
fit10
```

```
# Model 11: SARIMA(0,1,1)(2,1,1)
fit11 <- arima(log.train.ts,
              order = c(0,1,1),
              seasonal = list(order = c(2,1,1), period = 12),
              method = "ML")
fit11


# Model 12: SARIMA(0,1,1)(0,1,2)
fit12 <- arima(log.train.ts,
              order = c(0,1,1),
              seasonal = list(order = c(0,1,2), period = 12),
              method = "ML")
fit12
coefs12 <- coef(fit12)

ma_poly12 <- c(1, coefs12["ma1"])
plot.roots(ar.roots = NULL, ma.roots = polyroot(ma_poly12), main="(A) roots of ma part, nonseasonal")

sma_poly12 <- c(1, coefs12["sma1"], coefs12["sma2"])
plot.roots(ar.roots = NULL, ma.roots = polyroot(sma_poly12), main="(A) roots of ma part, seasonal")

# Model 13: SARIMA(0,1,1)(1,1,1)
fit13 <- arima(log.train.ts,
              order = c(0,1,1),
              seasonal = list(order = c(1,1,1), period = 12),
              method = "ML")
fit13

# Model 14: SARIMA(0,1,0)(2,1,1)
fit14 <- arima(log.train.ts,
              order = c(0,1,0),
              seasonal = list(order = c(2,1,1), period = 12),
              method = "ML")
fit14

# Model 15: SARIMA(1,1,0)(1,1,2)
fit15 <- arima(log.train.ts,
              order = c(1,1,0),
              seasonal = list(order = c(1,1,2), period = 12),
              method = "ML")
fit15
coefs15 <- coef(fit15)

ar_poly15 <- c(1, -coefs15["ar1"])
plot.roots(ar.roots = polyroot(ar_poly15), ma.roots = NULL, main="(A) roots of ar part, nonseasonal")

sar_poly15 <- c(1, -coefs15["sar1"])
plot.roots(ar.roots = polyroot(sar_poly15), ma.roots = NULL, main="(A) roots of ar part, seasonal")
```

```r
sma_poly15 <- c(1, coefs15["sma1"], coefs15["sma2"])
plot.roots(ar.roots = NULL, ma.roots = polyroot(sma_poly15), main="(A) roots of ma part, seasonal")
```

```r
# Model 16: SARIMA(1,1,0)(2,1,2)
fit16 <- arima(log.train.ts,
               order = c(1,1,0),
               seasonal = list(order = c(2,1,2), period = 12),
               method = "ML")
fit16
coefs16 <- coef(fit16)

ar_poly16 <- c(1, -coefs16["ar1"])
plot.roots(ar.roots = polyroot(ar_poly16), ma.roots = NULL, main="(A) roots of ar part, nonseasonal")

sar_poly16 <- c(1, -coefs16["sar1"], -coefs16["sar2"])
plot.roots(ar.roots = polyroot(sar_poly16), ma.roots = NULL, main="(A) roots of ar part, seasonal")

sma_poly16 <- c(1, coefs16["sma1"], coefs16["sma2"])
plot.roots(ar.roots = NULL, ma.roots = polyroot(sma_poly16), main="(A) roots of ma part, seasonal")

# Model 17: SARIMA(1,1,1)(1,1,2)
fit17 <- arima(log.train.ts,
               order = c(1,1,1),
               seasonal = list(order = c(1,1,2), period = 12),
               method = "ML")
fit17
coefs17 <- coef(fit17)

ar_poly17 <- c(1, -coefs17["ar1"])
plot.roots(ar.roots = polyroot(ar_poly17), ma.roots = NULL, main="(A) roots of ar part, nonseasonal")

ma_poly17 <- c(1, coefs17["ma1"])
plot.roots(ar.roots = NULL, ma.roots = polyroot(ma_poly17), main="(A) roots of ma part, nonseasonal")

sar_poly17 <- c(1, -coefs17["sar1"])
plot.roots(ar.roots = polyroot(sar_poly17), ma.roots = NULL, main="(A) roots of ar part, seasonal")

sma_poly17 <- c(1, coefs17["sma1"], coefs17["sma2"])
plot.roots(ar.roots = NULL, ma.roots = polyroot(sma_poly17), main="(A) roots of ma part, seasonal")

# Model 18: SARIMA(1,1,1)(2,1,2)
fit18 <- arima(log.train.ts,
               order = c(1,1,1),
               seasonal = list(order = c(2,1,2), period = 12),
               method = "ML")
fit18
coefs18 <- coef(fit18)

ar_poly18 <- c(1, -coefs18["ar1"])
plot.roots(ar.roots = polyroot(ar_poly18), ma.roots = NULL, main="(A) roots of ar part, nonseasonal")

ma_poly18 <- c(1, coefs18["ma1"])
plot.roots(ar.roots = NULL, ma.roots = polyroot(ma_poly18), main="(A) roots of ma part, nonseasonal")
```

```r
sar_poly18 <- c(1, -coefs18["sar1"], -coefs18["sar2"])
plot.roots(ar.roots = polyroot(sar_poly18), ma.roots = NULL, main="(A) roots of ar part, seasonal")

sma_poly18 <- c(1, coefs18["sma1"], coefs18["sma2"])
plot.roots(ar.roots = NULL, ma.roots = polyroot(sma_poly18), main="(A) roots of ma part, seasonal")

# Model 19: SARIMA(0,1,0)(0,1,2)
fit19 <- arima(log.train.ts,
               order = c(0,1,0),
               seasonal = list(order = c(0,1,2), period = 12),
               method = "ML")
fit19
coefs19 <- coef(fit19)

sma_poly19 <- c(1, coefs19["sma1"], coefs19["sma2"])
plot.roots(ar.roots = NULL, ma.roots = polyroot(sma_poly19), main="(A) roots of ma part, seasonal")

# Model 20: SARIMA(0,1,0)(1,1,1)
fit20 <- arima(log.train.ts,
               order = c(0,1,0),
               seasonal = list(order = c(1,1,1), period = 12),
               method = "ML")
fit20
coefs20 <- coef(fit20)

sar_poly20 <- c(1, -coefs20["sar1"])
plot.roots(ar.roots = polyroot(sar_poly20), ma.roots = NULL, main="(A) roots of ar part, seasonal")

sma_poly20 <- c(1, coefs20["sma1"])
plot.roots(ar.roots = NULL, ma.roots = polyroot(sma_poly20), main="(A) roots of ma part, seasonal")

# Model 21: SARIMA(0,1,1)(1,1,2)
fit21 <- arima(log.train.ts,
               order = c(0,1,1),
               seasonal = list(order = c(1,1,2), period = 12),
               method = "ML")
fit21
coefs21 <- coef(fit21)

ma_poly21 <- c(1, coefs21["ma1"])
plot.roots(ar.roots = NULL, ma.roots = polyroot(ma_poly21), main="(A) roots of ma part, nonseasonal")

sar_poly21 <- c(1, -coefs21["sar1"])
plot.roots(ar.roots = polyroot(sar_poly21), ma.roots = NULL, main="(A) roots of ar part, seasonal")

sma_poly21 <- c(1, coefs21["sma1"], coefs21["sma2"])
plot.roots(ar.roots = NULL, ma.roots = polyroot(sma_poly21), main="(A) roots of ma part, seasonal")
# Model 22: SARIMA(0,1,1)(2,1,2)
fit22 <- arima(log.train.ts,
               order = c(0,1,1),
               seasonal = list(order = c(2,1,2), period = 12),
               method = "ML")
fit22
```

```r
coefs22 <- coef(fit22)

ma_poly22 <- c(1, coefs22["ma1"])
plot.roots(ar.roots = NULL, ma.roots = polyroot(ma_poly22), main="(A) roots of ma part, nonseasonal")

sar_poly22 <- c(1, -coefs22["sar1"], -coefs22["sar2"])
plot.roots(ar.roots = polyroot(sar_poly22), ma.roots = NULL, main="(A) roots of ar part, seasonal")

sma_poly22 <- c(1, coefs22["sma1"], coefs22["sma2"])
plot.roots(ar.roots = NULL, ma.roots = polyroot(sma_poly22), main="(A) roots of ma part, seasonal")
# Model 23: SARIMA(0,1,0)(2,1,2)
fit23 <- arima(log.train.ts,
               order = c(0,1,0),
               seasonal = list(order = c(2,1,2), period = 12),
               method = "ML")
fit23
coefs23 <- coef(fit23)
```

```r
sar_poly23 <- c(1, -coefs23["sar1"], -coefs23["sar2"])
plot.roots(ar.roots = polyroot(sar_poly23), ma.roots = NULL, main="(A) roots of ar part, seasonal")

sma_poly23 <- c(1, coefs23["sma1"], coefs23["sma2"])
plot.roots(ar.roots = NULL, ma.roots = polyroot(sma_poly23), main="(A) roots of ma part, seasonal")

# Model 24: SARIMA(0,1,0)(1,1,2)
fit24 <- arima(log.train.ts,
               order = c(0,1,0),
               seasonal = list(order = c(1,1,2), period = 12),
               method = "ML")
fit24
coefs24 <- coef(fit24)

library(lmtest)
coeftest(fit24)

sar_poly24 <- c(1, -coefs24["sar1"])
plot.roots(ar.roots = polyroot(sar_poly24), ma.roots = NULL, main="(A) roots of ar part, seasonal")

sma_poly24 <- c(1, coefs24["sma1"], coefs24["sma2"])
plot.roots(ar.roots = NULL, ma.roots = polyroot(sma_poly24), main="(A) roots of ma part, seasonal")

# Model 25: SARIMA(1,1,0)(2,1,0)
fit25 <- arima(log.train.ts,
               order = c(1,1,0),
               seasonal = list(order = c(2,1,0), period = 12),
               method = "ML")
fit25
coefs25 <- coef(fit25)

ar_poly25 <- c(1, -coefs25["ar1"])
plot.roots(ar.roots = polyroot(ar_poly25), ma.roots = NULL, size= 6,main="(A) roots of ar part, nonseas

sar_poly25 <- c(1, -coefs25["sar1"], -coefs25["sar2"])
plot.roots(ar.roots = polyroot(sar_poly25), ma.roots = NULL, main="(A) roots of ar part, seasonal")
```

```r
ar_poly25 <- c(1, -coefs25["ar1"])
sar_poly25 <- c(1, -coefs25["sar1"], -coefs25["sar2"])

Mod(polyroot(ar_poly25))
Mod(polyroot(sar_poly25))
Mod(polyroot(ar_poly25))
Mod(polyroot(sar_poly25))

# Model 26: SARIMA(0,1,1)(2,1,0)
fit26 <- arima(log.train.ts,
               order = c(0,1,1),
               seasonal = list(order = c(2,1,0), period = 12),
               method = "ML")
fit26
coefs26 <- coef(fit26)

ma_poly26 <- c(1, coefs26["ma1"])
plot.roots(ar.roots = NULL, ma.roots = polyroot(ma_poly26), main="(A) roots of ma part, nonseasonal")

sar_poly26 <- c(1, -coefs26["sar1"], -coefs26["sar2"])
plot.roots(ar.roots = polyroot(sar_poly26), ma.roots = NULL, main="(A) roots of ar part, seasonal")

sar_poly26 <- c(1, -coefs26["sar1"], -coefs26["sar2"])
ma_poly26 <- c(1, coefs26["ma1"])
Mod(polyroot(sar_poly26))
Mod(polyroot(ma_poly26))

# Model 27: SARIMA(1,1,1)(2,1,0)
fit27 <- arima(log.train.ts,
               order = c(1,1,1),
               seasonal = list(order = c(2,1,0), period = 12),
               method = "ML")
fit27
coefs27 <- coef(fit27)

ar_poly27 <- c(1, -coefs27["ar1"])
plot.roots(ar.roots = polyroot(ar_poly27), ma.roots = NULL, main="(A) roots of ar part, nonseasonal")

ma_poly27 <- c(1, coefs27["ma1"])
plot.roots(ar.roots = NULL, ma.roots = polyroot(ma_poly27), main="(A) roots of ma part, nonseasonal")

sar_poly27 <- c(1, -coefs27["sar1"], -coefs27["sar2"])
plot.roots(ar.roots = polyroot(sar_poly27), ma.roots = NULL, main="(A) roots of ar part, seasonal")

Mod(polyroot(ar_poly27))
Mod(polyroot(ma_poly27))
Mod(polyroot(sar_poly27))
Mod(polyroot(ma_poly27))

# Model 28: SARIMA(0,1,0)(2,1,0)
fit28 <- arima(log.train.ts,
               order = c(0,1,0),
               seasonal = list(order = c(2,1,0), period = 12),
```

```
               method = "ML")
fit28
coefs28 <- coef(fit28)

sar_poly28 <- c(1, -coefs28["sar1"], -coefs28["sar2"])
plot.roots(ar.roots = polyroot(sar_poly28), ma.roots = NULL, main="(A) roots of ar part, seasonal")

Mod(polyroot(sar_poly28))
Mod(polyroot(sar_poly28))

# Model 29: SARIMA(1,1,0)(1,1,0)
fit29 <- arima(log.train.ts,
               order = c(1,1,0),
               seasonal = list(order = c(1,1,0), period = 12),
               method = "ML")
fit29
coefs29 <- coef(fit29)

ar_poly29 <- c(1, -coefs29["ar1"])
plot.roots(ar.roots = polyroot(ar_poly29), ma.roots = NULL, main="(A) roots of ar part, nonseasonal")

sar_poly29 <- c(1, -coefs29["sar1"])
plot.roots(ar.roots = polyroot(sar_poly29), ma.roots = NULL, main="(A) roots of ar part, seasonal")

Mod(polyroot(ar_poly29))
Mod(polyroot(sar_poly29))
Mod(polyroot(ar_poly29))
Mod(polyroot(sar_poly29))

# Model 30: SARIMA(1,1,1)(1,1,0)
fit30 <- arima(log.train.ts,
               order = c(1,1,1),
               seasonal = list(order = c(1,1,0), period = 12),
               method = "ML")
fit30
coefs30 <- coef(fit30)

ar_poly30 <- c(1, -coefs30["ar1"])
plot.roots(ar.roots = polyroot(ar_poly30), ma.roots = NULL, main="(A) roots of ar part, nonseasonal")

ma_poly30 <- c(1, coefs30["ma1"])
plot.roots(ar.roots = NULL, ma.roots = polyroot(ma_poly30), main="(A) roots of ma part, nonseasonal")

sar_poly30 <- c(1, -coefs30["sar1"])
plot.roots(ar.roots = polyroot(sar_poly30), ma.roots = NULL, size =6,main="(A) roots of ar part, seasona

Mod(polyroot(ar_poly30))
Mod(polyroot(ma_poly30))
Mod(polyroot(sar_poly30))
Mod(polyroot(ma_poly30))
```

```
# here I am running Shapiro-Wilk test for each model
for (i in 1:length(models)) {
  mod <- models[[i]]
```

```
  fit <- arima(log.train.ts,
               order = c(mod[1], mod[2], mod[3]),
               seasonal = list(order = c(mod[4], mod[5], mod[6]), period = 12),
               method = "ML")

  res <- residuals(fit)
  shap <- shapiro.test(res)

  cat(sprintf("Model %2d: (%d,%d,%d)(%d,%d,%d)  Shapiro-Wilk p = %.4f\n",
              i + 24, mod[1], mod[2], mod[3], mod[4], mod[5], mod[6], shap$p.value))
}

# Model Candidate (earlier candidate 2): failed White Noise Test & has high AIC
#so I discard it ----------------------------------------

fit36 <- arima(log.train.ts,
               order = c(1,1,1),
               seasonal = list(order = c(0,1,0), period = 12),
               method = "ML")
fit36
res <- residuals(fit36)
ar(res, aic = TRUE, order.max = NULL, method = c("yule-walker"))

acf(res)
pacf(res)

# 2 Candidate value diagnostic checks:

selected_models <- list(
  list(idx = 13, mod = c(1,1,0,2,1,0)),
  list(idx = 18, mod = c(1,1,1,1,1,0))
)

# Run diagnostics for each
for (m in selected_models) {
  cat(sprintf("Model %2d: (%d,%d,%d)(%d,%d,%d)\n",
              m$idx, m$mod[1], m$mod[2], m$mod[3], m$mod[4], m$mod[5], m$mod[6]))

  fit <- arima(log.train.ts,
               order = c(m$mod[1], m$mod[2], m$mod[3]),
               seasonal = list(order = c(m$mod[4], m$mod[5], m$mod[6]), period = 12),
               method = "ML")

  res <- residuals(fit)
  n <- length(log.train.ts)
  fitdf <- length(coef(fit))
  lag_value <- floor(sqrt(n))

  print(shapiro.test(res))
  print(Box.test(res, lag = lag_value, type = "Box-Pierce", fitdf = fitdf))
  print(Box.test(res, lag = lag_value, type = "Ljung-Box", fitdf = fitdf))
  print(Box.test(res^2, lag = lag_value, type = "Ljung-Box", fitdf = 0))
  print(ar(res, aic = TRUE, order.max = NULL, method = "yule-walker"))
```

```r
  cat("------------------------------------------------------------\n\n")
}

# canddidate model 2: ACF/PACF diagnostics residuals ------------------------------

par(mfrow = c(1, 2))

fitted_model_18 <- arima(log.train.ts,
                         order = c(1, 1, 1),
                         seasonal = list(order = c(1, 1, 0), period = 12),
                         method = "ML")

residuals_model_18 <- residuals(fitted_model_18)

series_length_18 <- length(log.train.ts)
num_parameters_18 <- length(coef(fitted_model_18))
acf_lag_18 <- floor(sqrt(series_length_18))

print(Box.test(residuals_model_18, lag = acf_lag_18, type = "Box-Pierce", fitdf = num_parameters_18))
print(Box.test(residuals_model_18, lag = acf_lag_18, type = "Ljung-Box", fitdf = num_parameters_18))
print(Box.test(residuals_model_18^2, lag = acf_lag_18, type = "Ljung-Box", fitdf = 0))
print(ar(residuals_model_18, aic = TRUE, order.max = NULL, method = "yule-walker"))

acf(residuals_model_18, lag.max = 40, main = "ACF of Residuals (Model 18)")
pacf(residuals_model_18, lag.max = 40, main = "PACF of Residuals (Model 18)")

# Fit Model 29: SARIMA(1,1,0)(1,1,0)
fitted_model_29 <- arima(log.train.ts,
                         order = c(1, 1, 0),
                         seasonal = list(order = c(1, 1, 0), period = 12),
                         method = "ML")

residuals_model_29 <- residuals(fitted_model_29)

hist(residuals_model_29, density = 20, breaks = 20, col = "blue", xlab = "", prob = TRUE)
residuals_mean_29 <- mean(residuals_model_29)
residuals_sd_29 <- sqrt(var(residuals_model_29))
curve(dnorm(x, residuals_mean_29, residuals_sd_29), add = TRUE)

plot.ts(residuals_model_29)
trend_line_model_29 <- lm(residuals_model_29 ~ as.numeric(1:length(residuals_model_29)))
abline(trend_line_model_29, col = "red")
abline(h = residuals_mean_29, col = "blue")

qqnorm(residuals_model_29, main = "Normal Q-Q Plot for Model 29")
qqline(residuals_model_29, col = "blue")

acf(residuals_model_29, lag.max = 40)
pacf(residuals_model_29, lag.max = 40)

shapiro.test(residuals_model_29)

total_length_model_29 <- length(log.train.ts)
```

```r
num_parameters_model_29 <- length(coef(fitted_model_29))
acf_lag_model_29 <- floor(sqrt(total_length_model_29))

Box.test(residuals_model_29, lag = acf_lag_model_29, type = "Box-Pierce", fitdf = num_parameters_model_
Box.test(residuals_model_29, lag = acf_lag_model_29, type = "Ljung-Box", fitdf = num_parameters_model_2
Box.test(residuals_model_29^2, lag = acf_lag_model_29, type = "Ljung-Box", fitdf = 0)

acf(residuals_model_29^2, lag.max = 40)
ar(residuals_model_29, aic = TRUE, order.max = NULL, method = "yule-walker")

# Fit Model 27: SARIMA(1,1,1)(2,1,0)
fitted_model_27 <- arima(log.train.ts,
                         order = c(1, 1, 1),
                         seasonal = list(order = c(2, 1, 0), period = 12),
                         method = "ML")

residuals_model_27 <- residuals(fitted_model_27)

hist(residuals_model_27, density = 20, breaks = 20, col = "blue", xlab = "", prob = TRUE)
residuals_mean <- mean(residuals_model_27)
residuals_sd <- sqrt(var(residuals_model_27))
curve(dnorm(x, residuals_mean, residuals_sd), add = TRUE)

plot.ts(residuals_model_27)
trend_line_model_27 <- lm(residuals_model_27 ~ as.numeric(1:length(residuals_model_27)))
abline(trend_line_model_27, col = "red")
abline(h = residuals_mean, col = "blue")

qqnorm(residuals_model_27, main = "Normal Q-Q Plot for Model 27")
qqline(residuals_model_27, col = "blue")

acf(residuals_model_27, lag.max = 40)
pacf(residuals_model_27, lag.max = 40)

shapiro.test(residuals_model_27)

time_series_length <- length(log.train.ts)
model_27_num_coeff <- length(coef(fitted_model_27))
acf_lag_setting <- floor(sqrt(time_series_length))

Box.test(residuals_model_27, lag = acf_lag_setting, type = "Box-Pierce", fitdf = model_27_num_coeff)
Box.test(residuals_model_27, lag = acf_lag_setting, type = "Ljung-Box", fitdf = model_27_num_coeff)
Box.test(residuals_model_27^2, lag = acf_lag_setting, type = "Ljung-Box", fitdf = 0)

acf(residuals_model_27^2, lag.max = 40)
ar(residuals_model_27, aic = TRUE, order.max = NULL, method = "yule-walker")

# model 25: (CANDIDATE MODEL 1, LOWEST AIC) ------------------------------------------
fitted_model_25 <- arima(log.train.ts,
                         order = c(1, 1, 0),
                         seasonal = list(order = c(2, 1, 0), period = 12),
                         method = "ML")
fitted_model_25
```

```r
residuals_model_25 <- residuals(fitted_model_25)

# Histogram of residuals with normal curve
hist(residuals_model_25, density = 20, breaks = 20, col = "blue", xlab = "", prob = TRUE)
res_mean <- mean(residuals_model_25)
res_std_dev <- sqrt(var(residuals_model_25))
curve(dnorm(x, res_mean, res_std_dev), add = TRUE)

# Time series plot with trend line and mean
plot.ts(residuals_model_25)
residual_trend_fit <- lm(residuals_model_25 ~ as.numeric(1:length(residuals_model_25)))
abline(residual_trend_fit, col = "red")
abline(h = res_mean, col = "blue")

# Q-Q plot
qqnorm(residuals_model_25, main = "Normal Q-Q Plot for Model 25")
qqline(residuals_model_25, col = "blue")

# ACF and PACF of residuals
acf(residuals_model_25, lag.max = 40)
pacf(residuals_model_25, lag.max = 40)

# Normality test
shapiro.test(residuals_model_25)
```

```r
# Set lag and fitdf
series_length <- length(log.train.ts)
num_parameters_estimated <- length(coef(fitted_model_25))
autocorr_lag_value <- floor(sqrt(series_length))

# Box-Pierce and Ljung-Box tests on residuals
Box.test(residuals_model_25, lag = autocorr_lag_value, type = "Box-Pierce", fitdf = num_parameters_esti
Box.test(residuals_model_25, lag = autocorr_lag_value, type = "Ljung-Box", fitdf = num_parameters_estim

# Ljung-Box on squared residuals (McLeod-Li test)
Box.test(residuals_model_25^2, lag = autocorr_lag_value, type = "Ljung-Box", fitdf = 0)

# ACF of squared residuals
acf(residuals_model_25^2, lag.max = 40)

# AR model on residuals to test remaining autocorrelation
ar(residuals_model_25, aic = TRUE, order.max = NULL, method = "yule-walker")

# Additional ACF/PACF for visual inspection
acf(residuals_model_25)
pacf(residuals_model_25)

# CANDIDATE MODEL 2: Model 36: SARIMA(1,1,1)(0,1,0)-------------------------------------------------

fitted_model_36 <- arima(log.train.ts,
                         order = c(1, 1, 1),
                         seasonal = list(order = c(0, 1, 0), period = 12),
                         method = "ML")
```

```r
fitted_model_36

residuals_model_36 <- residuals(fitted_model_36)

# Histogram of residuals with normal curve
hist(residuals_model_36, density = 20, breaks = 20, col = "blue", xlab = "", prob = TRUE)
res_mean <- mean(residuals_model_36)
res_sd <- sqrt(var(residuals_model_36))
curve(dnorm(x, res_mean, res_sd), add = TRUE)

# Time series plot with trend line and mean
plot.ts(residuals_model_36)
res_trend_model <- lm(residuals_model_36 ~ as.numeric(1:length(residuals_model_36)))
abline(res_trend_model, col = "red")
abline(h = res_mean, col = "blue")

# Q-Q plot
qqnorm(residuals_model_36, main = "Normal Q-Q Plot for Model 36")
qqline(residuals_model_36, col = "blue")

# ACF and PACF of residuals
acf(residuals_model_36, lag.max = 40)
pacf(residuals_model_36, lag.max = 40)

# Normality test
shapiro.test(residuals_model_36)

series_length <- length(log.train.ts)
num_parameters <- length(coef(fitted_model_36))
acf_lag_limit <- floor(sqrt(series_length))

# Box-Pierce and Ljung-Box tests
Box.test(residuals_model_36, lag = acf_lag_limit, type = "Box-Pierce", fitdf = num_parameters)
Box.test(residuals_model_36, lag = acf_lag_limit, type = "Ljung-Box", fitdf = num_parameters)

# McLeod-Li test on squared residuals
Box.test(residuals_model_36^2, lag = acf_lag_limit, type = "Ljung-Box", fitdf = 0)

# ACF of squared residuals
acf(residuals_model_36^2, lag.max = 40)

ar(residuals_model_36, aic = TRUE, order.max = NULL, method = "yule-walker")

# ACF and PACF of residuals again for visual confirmation
acf(residuals_model_36)
pacf(residuals_model_36)

#double checking seasonal AR root modulus: ------------------------------------------------------------
fit1 <- arima(log.train.ts,
              order = c(1,1,0),
              seasonal = list(order = c(2,1,0), period = 12),
              method = "ML")
```

```r
# Non-seasonal AR roots
ar_poly1 <- c(1, -coef(fit1)["ar1"])
cat("Non-seasonal AR root modulus:\n")
print(Mod(polyroot(ar_poly1)))

# Seasonal AR roots
sar_poly1 <- c(1, -coef(fit1)["sar1"], -coef(fit1)["sar2"])
cat("Seasonal AR root modulus:\n")
print(Mod(polyroot(sar_poly1)))

#double checking seasonal AR root modulus: ------------------------------------------------------------
fit2 <- arima(log.train.ts,
              order = c(1,1,1),
              seasonal = list(order = c(1,1,0), period = 12),
              method = "ML")

# Non-seasonal AR roots
ar_poly2 <- c(1, -coef(fit2)["ar1"])
cat("Non-seasonal AR root modulus:\n")
print(Mod(polyroot(ar_poly2)))

# Seasonal AR roots
sar_poly2 <- c(1, -coef(fit2)["sar1"])
cat("Seasonal AR root modulus:\n")
print(Mod(polyroot(sar_poly2)))

# Fit Model 30: SARIMA(1,1,1)(1,1,0)
fit_model_30 <- arima(log.train.ts,
                      order = c(1, 1, 1),
                      seasonal = list(order = c(1, 1, 0), period = 12),
                      method = "ML")

residuals_model_30 <- residuals(fit_model_30)

# Histogram with normal curve overlay
hist(residuals_model_30, density = 20, breaks = 20, col = "blue", xlab = "", prob = TRUE)
residual_mean <- mean(residuals_model_30)
residual_sd <- sqrt(var(residuals_model_30))
curve(dnorm(x, residual_mean, residual_sd), add = TRUE)

# Residual time series plot with trend
plot.ts(residuals_model_30)
residual_trend_fit <- lm(residuals_model_30 ~ as.numeric(1:length(residuals_model_30)))
abline(residual_trend_fit, col = "red")
abline(h = residual_mean, col = "blue")

# Q-Q plot
qqnorm(residuals_model_30, main = "Normal Q-Q Plot for Model 30")
qqline(residuals_model_30, col = "blue")

# ACF and PACF
acf(residuals_model_30, lag.max = 40)
pacf(residuals_model_30, lag.max = 40)
```

```r
# Normality test
shapiro.test(residuals_model_30)

# Diagnostic tests
sample_size <- length(log.train.ts)
num_parameters <- length(coef(fit_model_30))
lag_threshold <- floor(sqrt(sample_size))

Box.test(residuals_model_30, lag = lag_threshold, type = "Box-Pierce", fitdf = num_parameters)
Box.test(residuals_model_30, lag = lag_threshold, type = "Ljung-Box", fitdf = num_parameters)
Box.test(residuals_model_30^2, lag = lag_threshold, type = "Ljung-Box", fitdf = 0)

acf(residuals_model_30^2, lag.max = 40)
ar(residuals_model_30, aic = TRUE, order.max = NULL, method = "yule-walker")

# Fit Model 30: SARIMA(1,1,1)(1,1,0)
fit_model_30 <- arima(log.train.ts,
                      order = c(1, 1, 1),
                      seasonal = list(order = c(1, 1, 0), period = 12),
                      method = "ML")

residuals_model_30 <- residuals(fit_model_30)

# Histogram with normal curve overlay
hist(residuals_model_30, density = 20, breaks = 20, col = "blue", xlab = "", prob = TRUE)
residual_mean <- mean(residuals_model_30)
residual_sd <- sqrt(var(residuals_model_30))
curve(dnorm(x, residual_mean, residual_sd), add = TRUE)

# Residual time series plot with trend
plot.ts(residuals_model_30)
residual_trend_fit <- lm(residuals_model_30 ~ as.numeric(1:length(residuals_model_30)))
abline(residual_trend_fit, col = "red")
abline(h = residual_mean, col = "blue")

# Q-Q plot
qqnorm(residuals_model_30, main = "Normal Q-Q Plot for Model 30")
qqline(residuals_model_30, col = "blue")

# ACF and PACF
acf(residuals_model_30, lag.max = 40)
pacf(residuals_model_30, lag.max = 40)

# Normality test
shapiro.test(residuals_model_30)

# Diagnostic tests
sample_size <- length(log.train.ts)
num_parameters <- length(coef(fit_model_30))
lag_threshold <- floor(sqrt(sample_size))

Box.test(residuals_model_30, lag = lag_threshold, type = "Box-Pierce", fitdf = num_parameters)
Box.test(residuals_model_30, lag = lag_threshold, type = "Ljung-Box", fitdf = num_parameters)
```

```r
Box.test(residuals_model_30^2, lag = lag_threshold, type = "Ljung-Box", fitdf = 0)

acf(residuals_model_30^2, lag.max = 40)
ar(residuals_model_30, aic = TRUE, order.max = NULL, method = "yule-walker")

# Model 30 [Candidate Model 2]: Forecasting SARIMA(1,1,1)(1,1,0)
library(forecast)

sarima_model_30 <- arima(log.train.ts,
                         order = c(1, 1, 1),
                         seasonal = list(order = c(1, 1, 0), period = 12),
                         method = "ML")

# Forecast 12 months ahead
sarima_forecast_30 <- predict(sarima_model_30, n.ahead = 12)
log_forecast_upper_30 <- sarima_forecast_30$pred + 2 * sarima_forecast_30$se
log_forecast_lower_30 <- sarima_forecast_30$pred - 2 * sarima_forecast_30$se

# Plot forecast on log-transformed scale
ts.plot(as.numeric(log.train.ts),
        xlim = c(1, length(log.train.ts) + 12),
        ylim = c(min(log.train.ts), max(log_forecast_upper_30)),
        main = "Forecast on Log Scale: Model 30")
lines(log_forecast_upper_30, col = "blue", lty = "dashed")
lines(log_forecast_lower_30, col = "blue", lty = "dashed")
points((length(log.train.ts) + 1):(length(log.train.ts) + 12), sarima_forecast_30$pred, col = "red")

# Back-transform forecast to original scale
forecast_original_30 <- exp(sarima_forecast_30$pred)
forecast_upper_30 <- exp(log_forecast_upper_30)
forecast_lower_30 <- exp(log_forecast_lower_30)

ts.plot(as.numeric(train.ts),
        xlim = c(1, length(train.ts) + 12),
        ylim = c(min(house.ts), max(house.ts)),
        main = "Forecast on Original Scale: Model 30",
        col = "red")
lines(forecast_upper_30, col = "blue", lty = "dashed")
lines(forecast_lower_30, col = "blue", lty = "dashed")
points((length(train.ts) + 1):(length(train.ts) + 12), forecast_original_30, col = "black")

# Zoomed-in final plot
ts.plot(as.numeric(house.ts),
        xlim = c(100, length(train.ts) + 12),
        ylim = c(0, max(forecast_upper_30)),
        col = "red")
lines(forecast_upper_30, col = "blue", lty = "dashed")
lines(forecast_lower_30, col = "blue", lty = "dashed")
points((length(train.ts) + 1):(length(train.ts) + 12), forecast_original_30, col = "green")
points((length(train.ts) + 1):(length(train.ts) + 12), forecast_original_30, col = "black")

forecast_mean_30 <- forecast(sarima_model_30, h = 12)$mean
data.frame(
```

```
  Month = 121:132,
  Forecast = round(as.numeric(exp(forecast_mean_30)), 2),
  Actual = house.ts[121:132]
)

# Full forecast object and 95% upper confidence interval
forecast_object_30 <- forecast(sarima_model_30, h = 12)
upper_bound_95_30 <- exp(forecast_object_30$upper[, "95%"])
data.frame(
  Month = 121:132,
  Actual = house.ts[121:132],
  Upper_95 = round(upper_bound_95_30, 2)
)

# Model 25: (Candidate 1 Model) Forecasting SARIMA(1,1,0)(2,1,0)
library(forecast)

sarima_model_25 <- arima(log.train.ts,
                         order = c(1, 1, 0),
                         seasonal = list(order = c(2, 1, 0), period = 12),
                         method = "ML")

# Forecast 12 months ahead
sarima_forecast_25 <- predict(sarima_model_25, n.ahead = 12)
log_forecast_upper_25 <- sarima_forecast_25$pred + 2 * sarima_forecast_25$se
log_forecast_lower_25 <- sarima_forecast_25$pred - 2 * sarima_forecast_25$se

# Plot forecast on log-transformed scale
ts.plot(as.numeric(log.train.ts),
        xlim = c(1, length(log.train.ts) + 12),
        ylim = c(min(log.train.ts), max(log_forecast_upper_25)),
        main = "Forecast on Log Scale: Model 25")
lines(log_forecast_upper_25, col = "blue", lty = "dashed")
lines(log_forecast_lower_25, col = "blue", lty = "dashed")
points((length(log.train.ts) + 1):(length(log.train.ts) + 12), sarima_forecast_25$pred, col = "red")

# Back-transform forecast to original scale
forecast_original_25 <- exp(sarima_forecast_25$pred)
forecast_upper_25 <- exp(log_forecast_upper_25)
forecast_lower_25 <- exp(log_forecast_lower_25)

ts.plot(as.numeric(train.ts),
        xlim = c(1, length(train.ts) + 12),
        ylim = c(min(house.ts), max(house.ts)),
        main = "Forecast on Original Scale: Model 25",
        col = "red")
lines(forecast_upper_25, col = "blue", lty = "dashed")
lines(forecast_lower_25, col = "blue", lty = "dashed")
points((length(train.ts) + 1):(length(train.ts) + 12), forecast_original_25, col = "black")

# Zoomed-in final plot
ts.plot(as.numeric(house.ts),
        xlim = c(100, length(train.ts) + 12),
```

```r
        ylim = c(0, max(house.ts)),
        col = "red")
lines(forecast_upper_25, col = "blue", lty = "dashed")
lines(forecast_lower_25, col = "blue", lty = "dashed")
points((length(train.ts) + 1):(length(train.ts) + 12), forecast_original_25, col = "green")
points((length(train.ts) + 1):(length(train.ts) + 12), forecast_original_25, col = "black")

sarima_forecast_25 <- predict(sarima_model_25, n.ahead = 12)
forecast_original_25 <- exp(sarima_forecast_25$pred)
forecast_upper_25 <- exp(sarima_forecast_25$pred + 2 * sarima_forecast_25$se)
forecast_lower_25 <- exp(sarima_forecast_25$pred - 2 * sarima_forecast_25$se)

actual_test_values <- house.ts[121:132]
forecast_outside <- sum(actual_test_values > forecast_upper_25 | actual_test_values < forecast_lower_25)
forecast_inside <- sum(actual_test_values >= forecast_lower_25 & actual_test_values <= forecast_upper_25)

cat("Number of values outside 95% CI:", forecast_outside, "\n")
cat("Number of values inside 95% CI:", forecast_inside, "\n")

abline(v = 121, col = "green", lty = "dashed")
abline(v = 122, col = "green", lty = "dashed")
abline(v = 123, col = "green", lty = "dashed")
abline(v = 129, col = "green", lty = "dashed")

forecast_mean_25 <- forecast(sarima_model_25, 12)$mean

data.frame(
  Month = 121:132,
  Forecast = round(as.numeric(exp(forecast_mean_25)), 2),
  Actual = house.ts[121:132]
)

forecast_object_25 <- forecast(sarima_model_25, h = 12)

upper_bound_95 <- exp(forecast_object_25$upper[, "95%"])

data.frame(
  Month = 121:132,
  Actual = house.ts[121:132],
  Upper_95 = round(upper_bound_95, 2)
)
```

Figure 1:

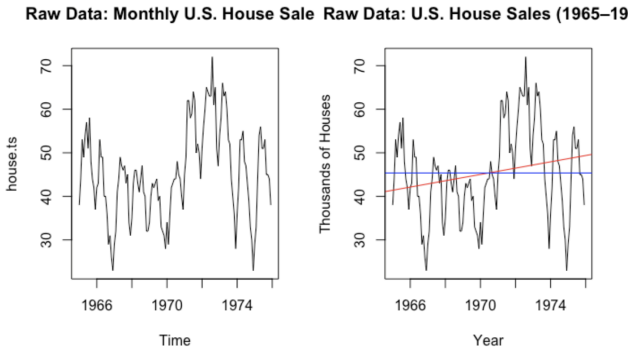


Figure 2:

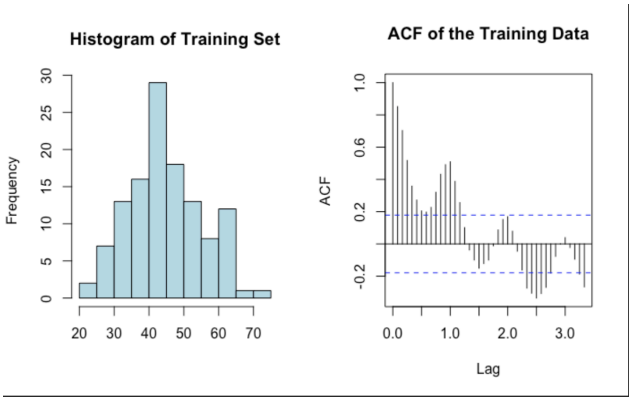


Figure 3:

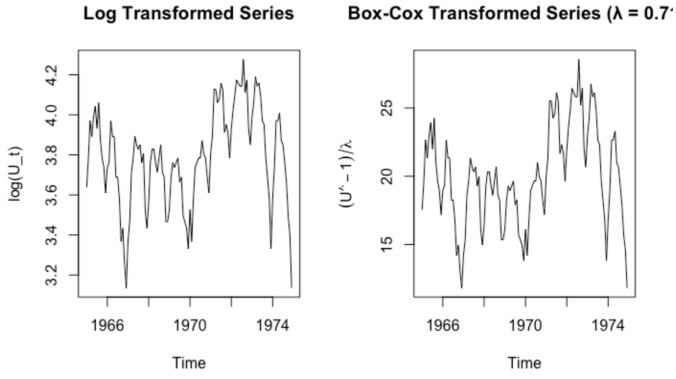


Figure 4:

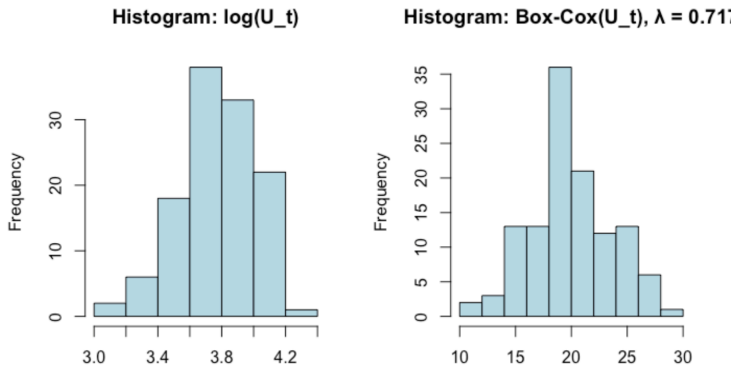


Figure 5:

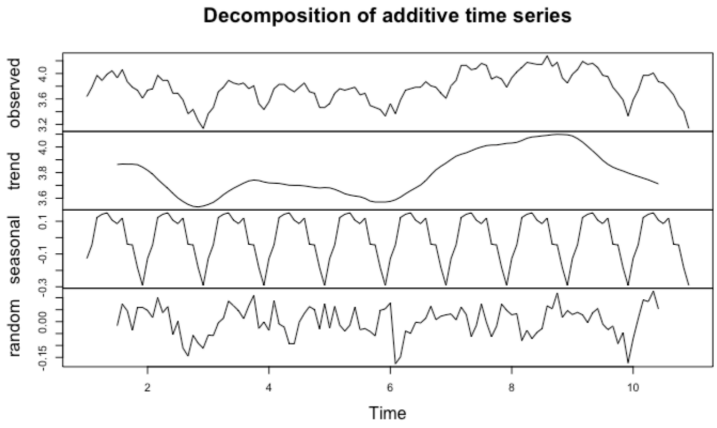


Figure 6:

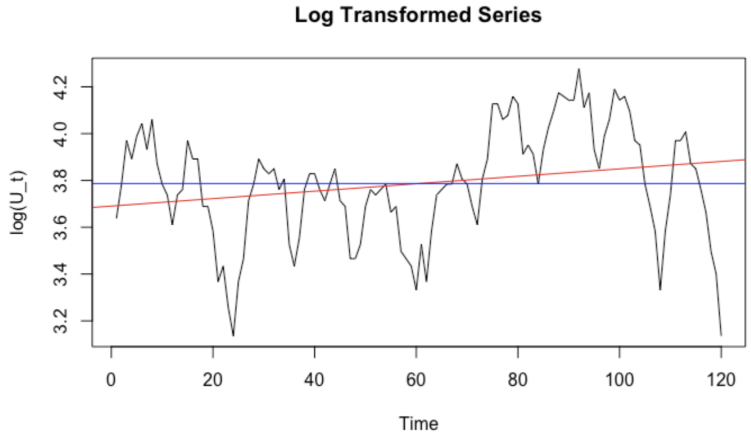


Figure 7:

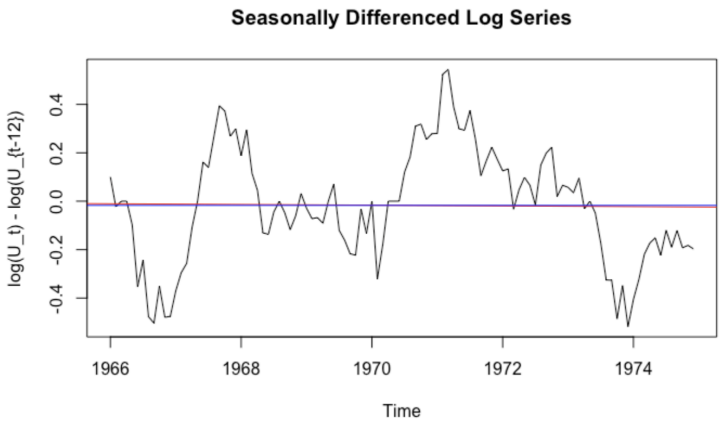


Figure 8:

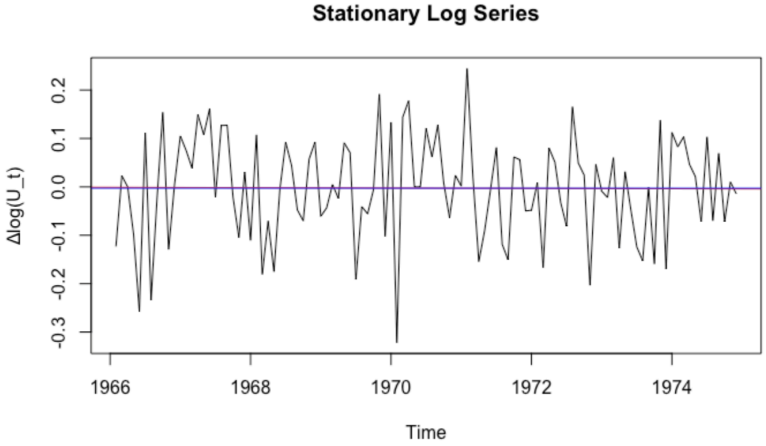


Figure 9:

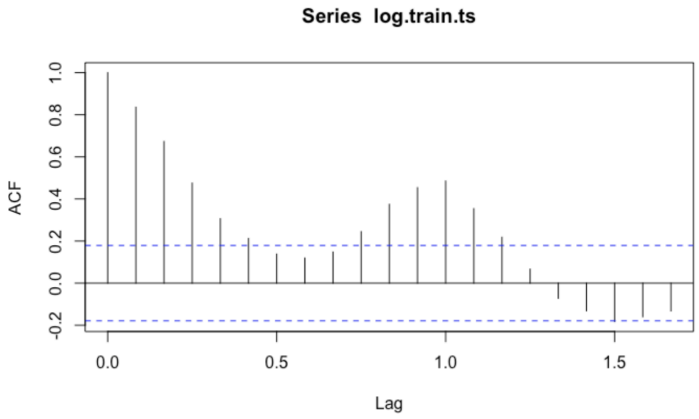


Figure 10:

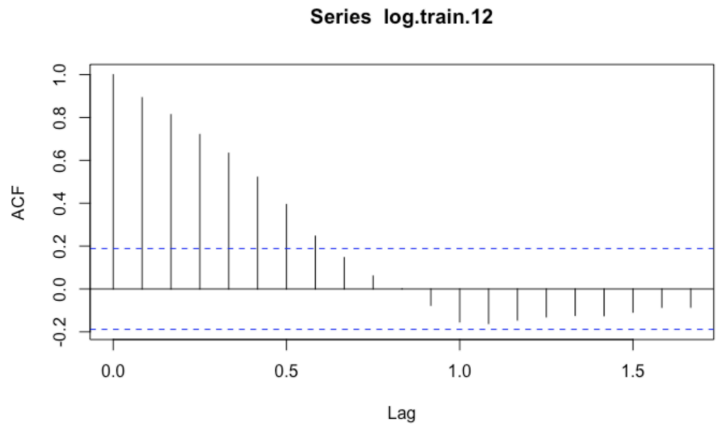


Figure 11:

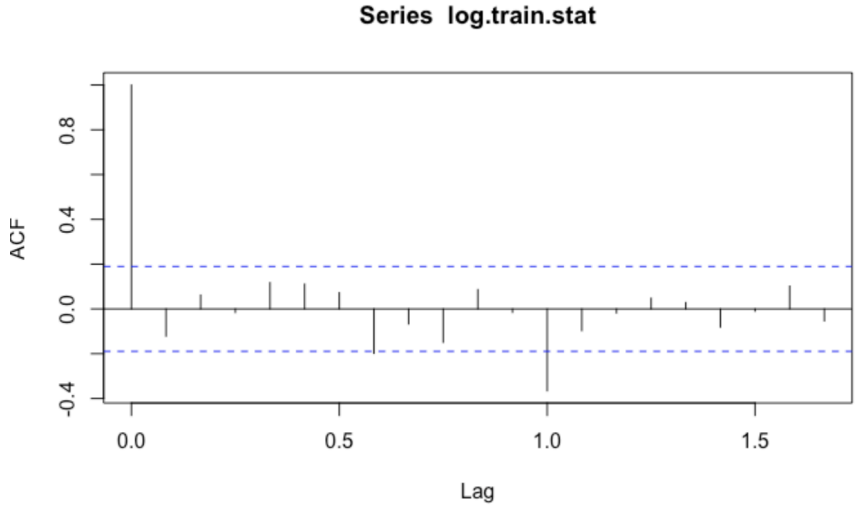


Figure 12:

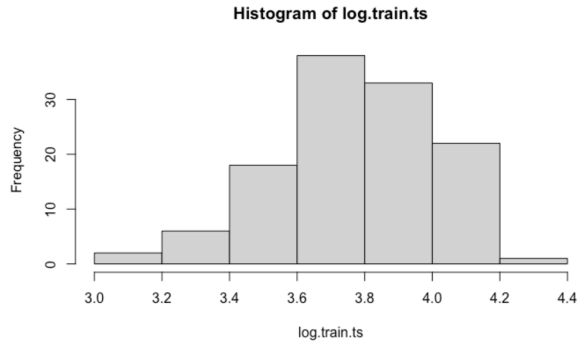


Figure 13:

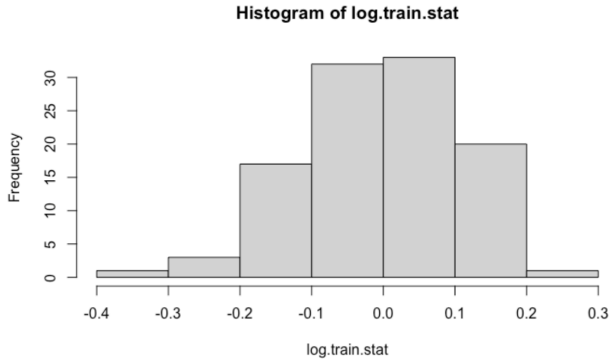


Figure 14:

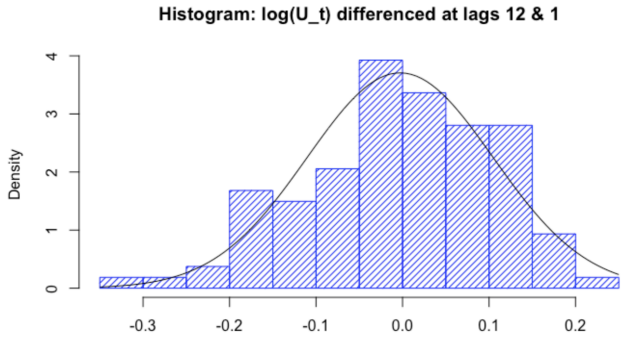
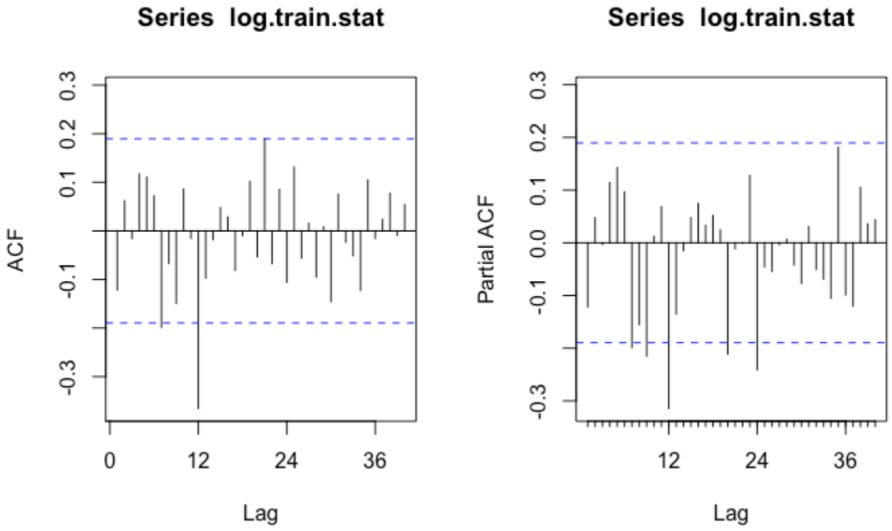
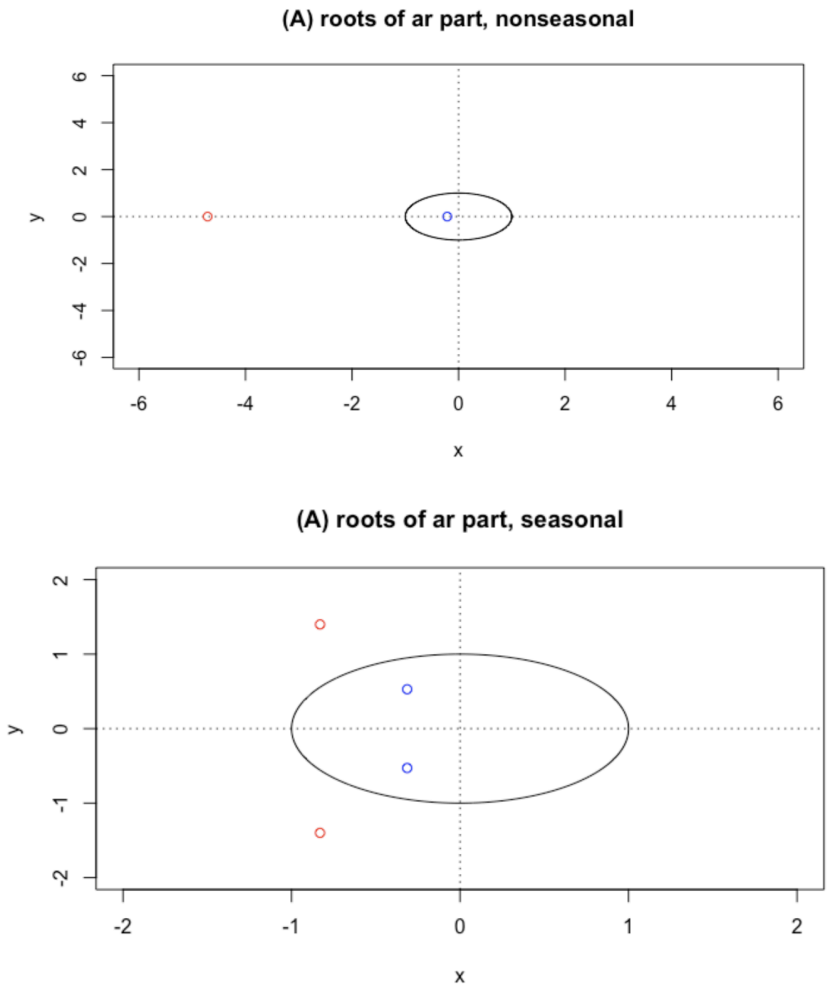


Figure 15:


Series log.train.stat
ACF
0.3
0.2
0.1
-0.1
-0.3
0
12
24
36
Lag
Series log.train.stat
Partial ACF
0.3
0.2
0.1
0.0
-0.1
-0.3
12
24
36
Lag

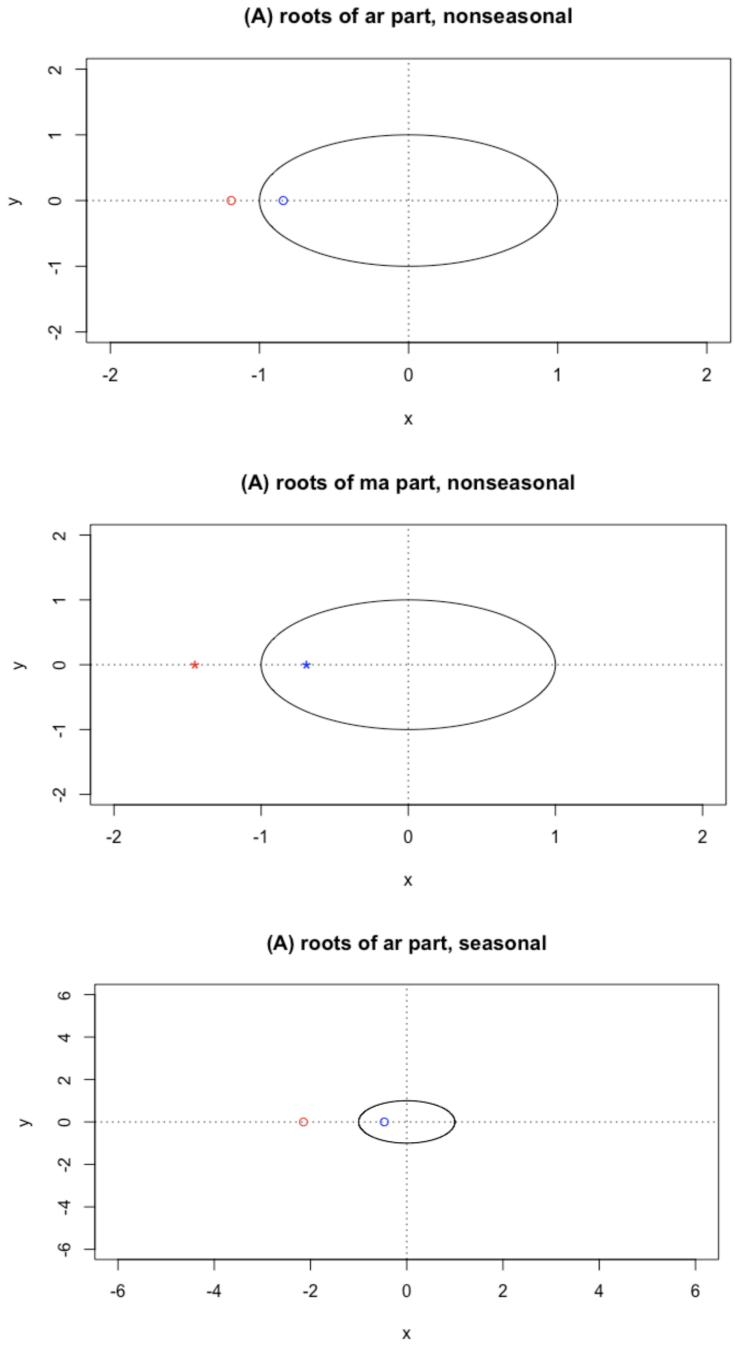
SARIMA(1,1,0)(2,1,0)[12]):

Figure 16:



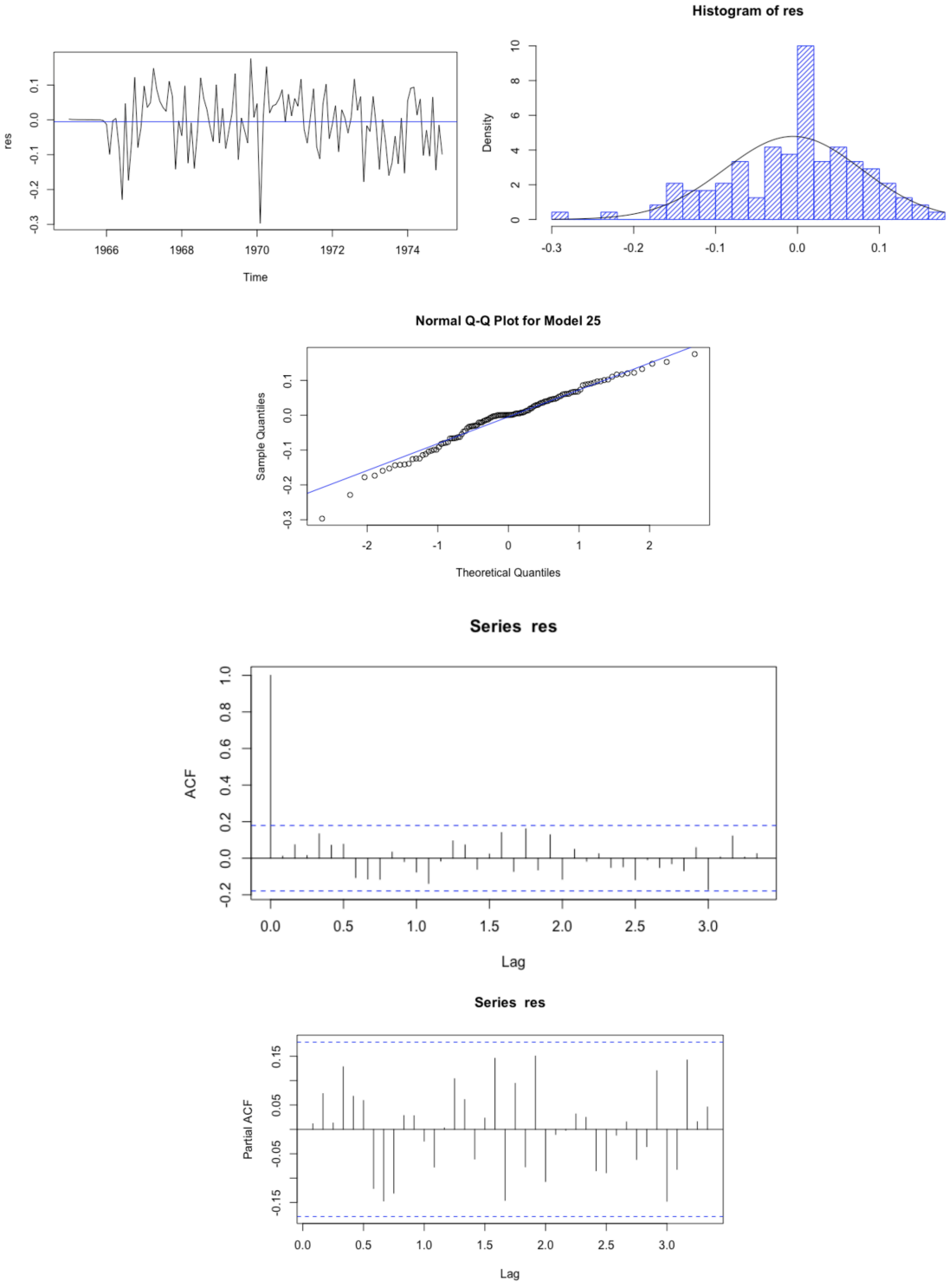
SARIMA(1,1,1)(1,1,0)[12])

Figure 17:



SARIMA(1,1,0)(2,1,0)[12]):

Figure 18:



Candidate 2: SARIMA(1,1,1)(1,1,0)[12])

Figure 19:

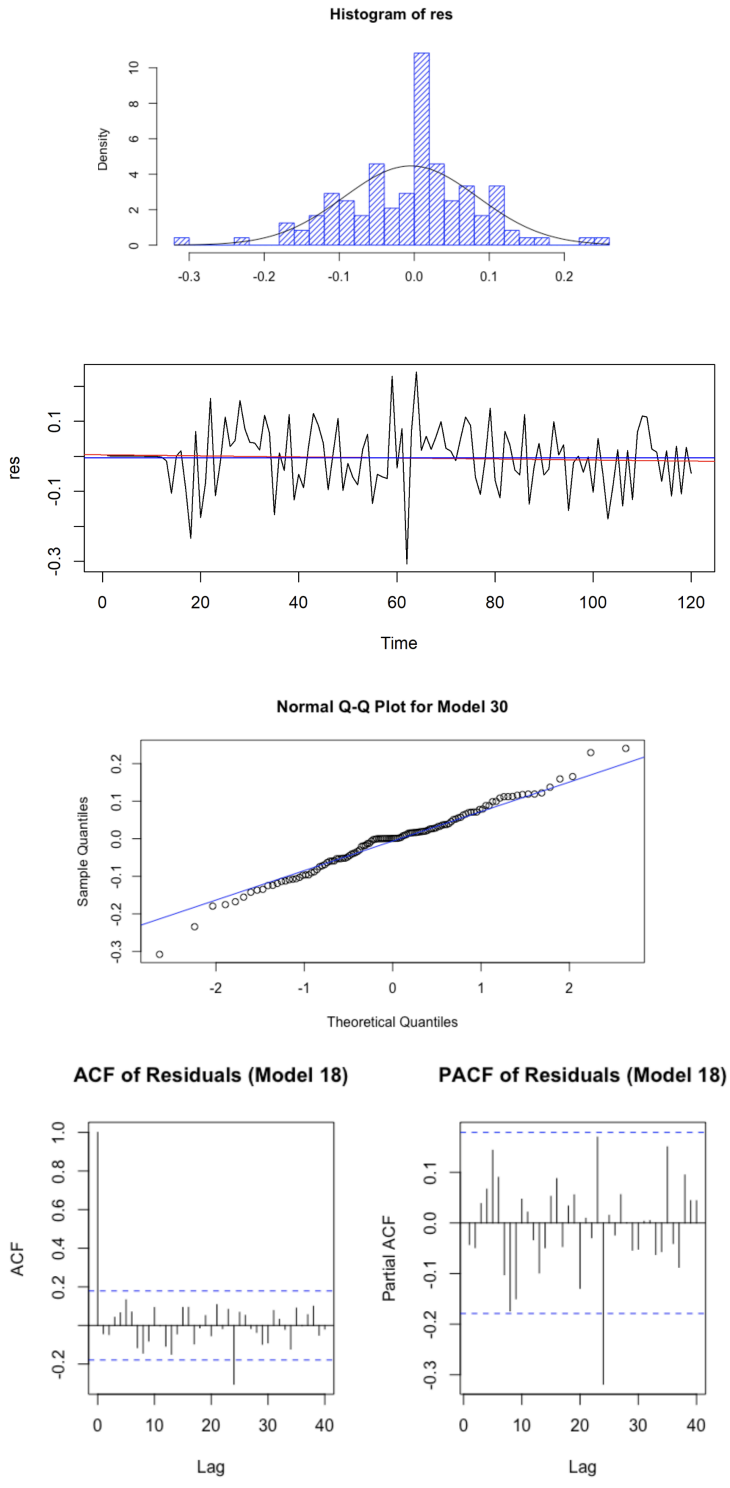


Figure 20 (Candidate 1):

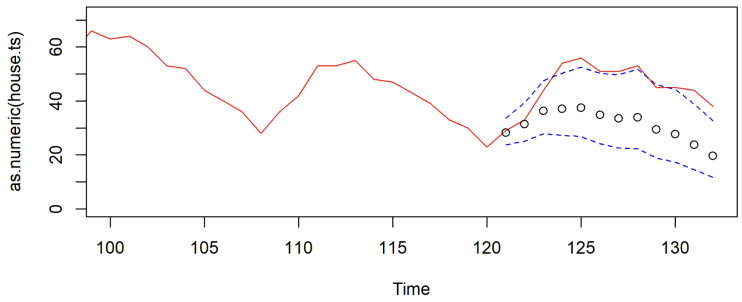


Figure 21 (Candidate 2):