Electric Production Time Series

Leslie Cervantes

2025-05-21

Abstract

This project analyzes monthly electricity production data with the goal of building an accurate forecasting model. The data demonstrated trend, seasonality, and fluctuations of variance. Hence, a Box-Cox transformation was used to stabilize the variance and make the data normally distributed. The data was then seasonally and non-seasonally differenced to achieve stationarity. Several SARIMA models were assessed based on AICc, stationarity, invertibility, and residual diagnostic checking.

The final model was $SARIMA(2,1,1) \times (2,1,1)_{12}$ with AR(1) and SAR(1) fixed to zero. The forecasts closely followed the true values for both the transformed and testing datasets. This model works well for forecasting electric production and may be used for future data.

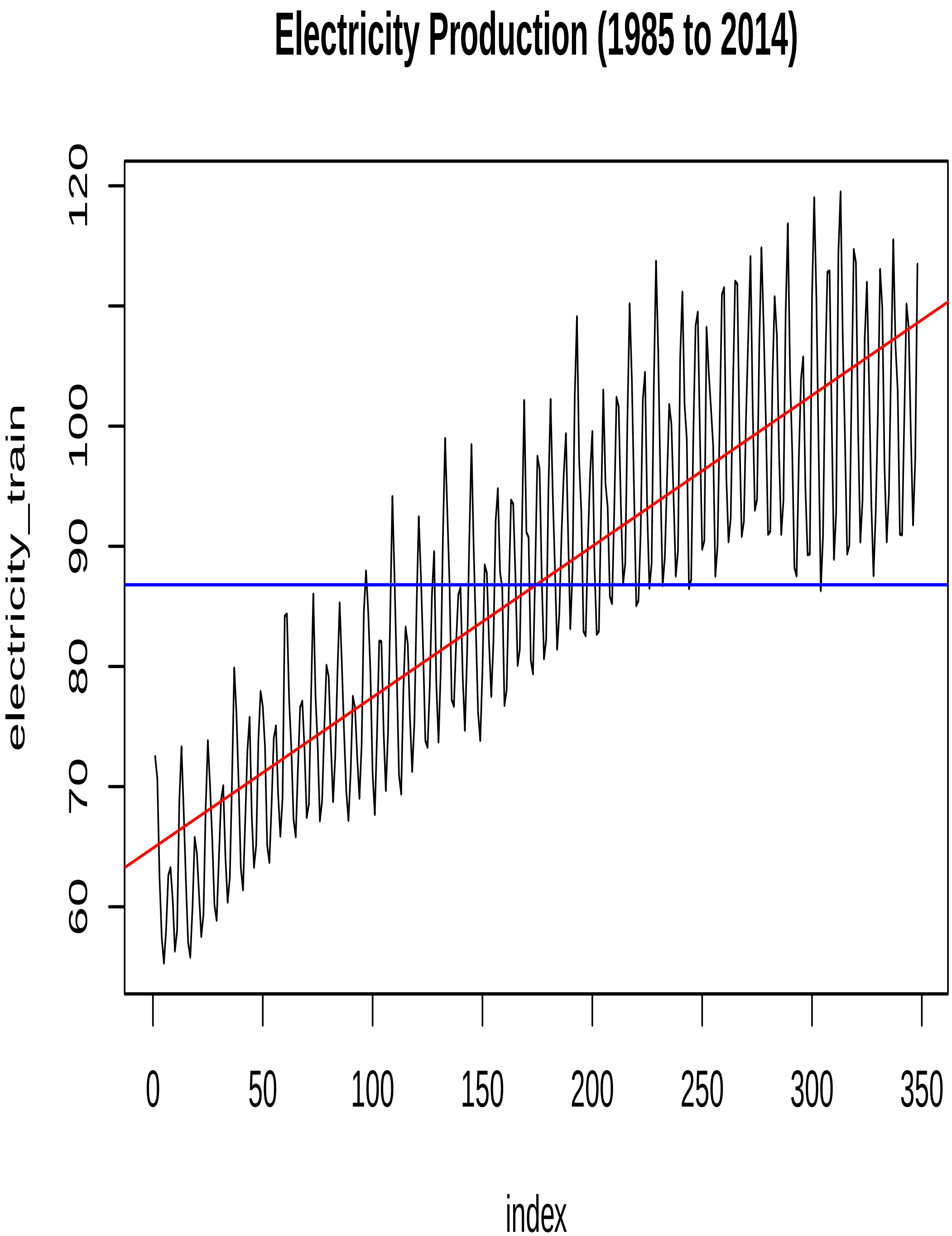
Introduction

The dataset I will be using is Electric Production from Kaggle provided by user ShenbagaKumarS. It contains monthly production data from January 1985 to January 2018, totaling 397 observations. This dataset tracks electric production trends over time and can be used to analyze seasonal patterns and long-term growth.

This data is interesting because it offers a long-term view of electric production. It is important because electricity is essential in modern life and drives economic growth. This analysis aims to determine whether electric production continues to increase over time. Understanding these trends is important because rising production will contribute to higher greenhouse gas emissions, worsening climate change and impact human health.

The goal of this project is to forecast future electric production using historical monthly data. This was achieved using time series techniques, including Box-Cox transformation, seasonal and non-seasonal differencing, SARIMA modeling, and residual diagnostics. The final model demonstrated good predictive performance, with forecasts closely following true values, making it suitable for future data. Additionally, I used R version 2024.12.01 and the following libraries: dplyr, tidyverse, MASS, knitr, qpcR, forecast, and astsa.

Electric Production

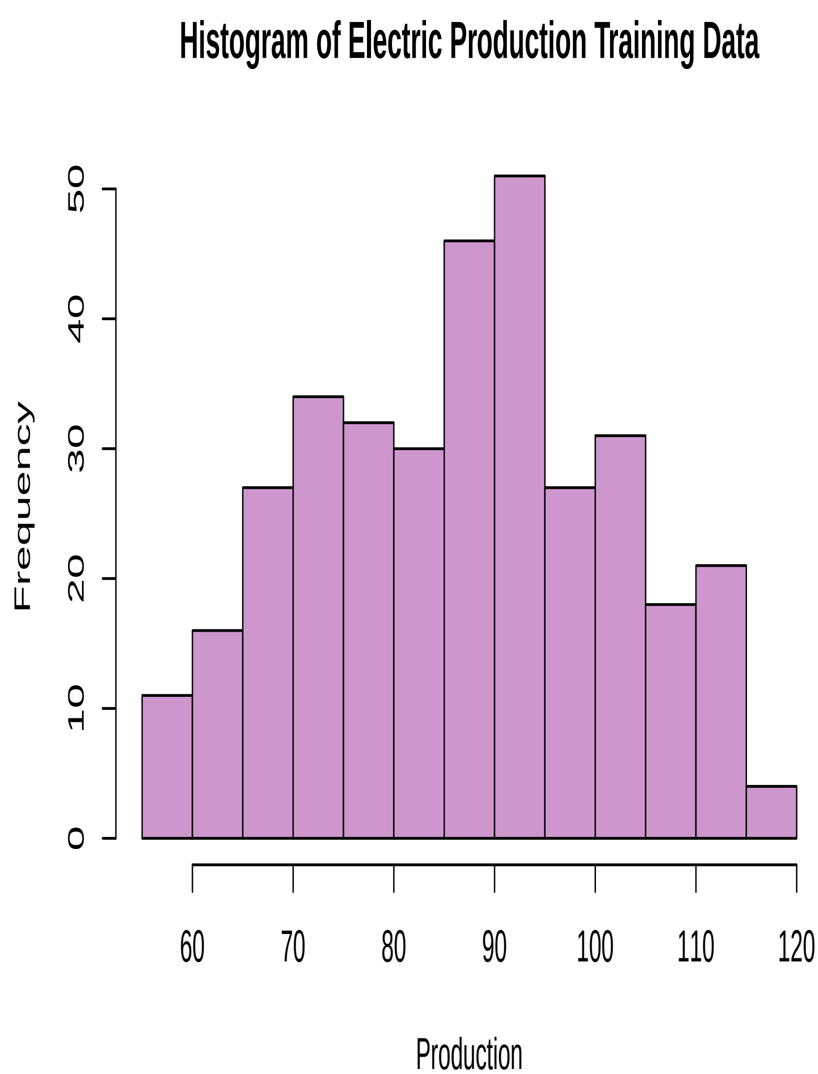



```
## [1] "The variance is: 221.551464966789"
```

The plot demonstrates there is an increasing trend indicating electric production has grown. The variability in production increases over time, indicating the fluctuations in production become bigger as the years progress. The data shows a seasonal component, with recurring peaks at regular intervals. This may reflect higher usage during winter and summer, as people increase their use of heating and cooling appliances. Additionally, a noticeable sharp change occurs around index 300, where electricity production decreased significantly by approximately 20 units.

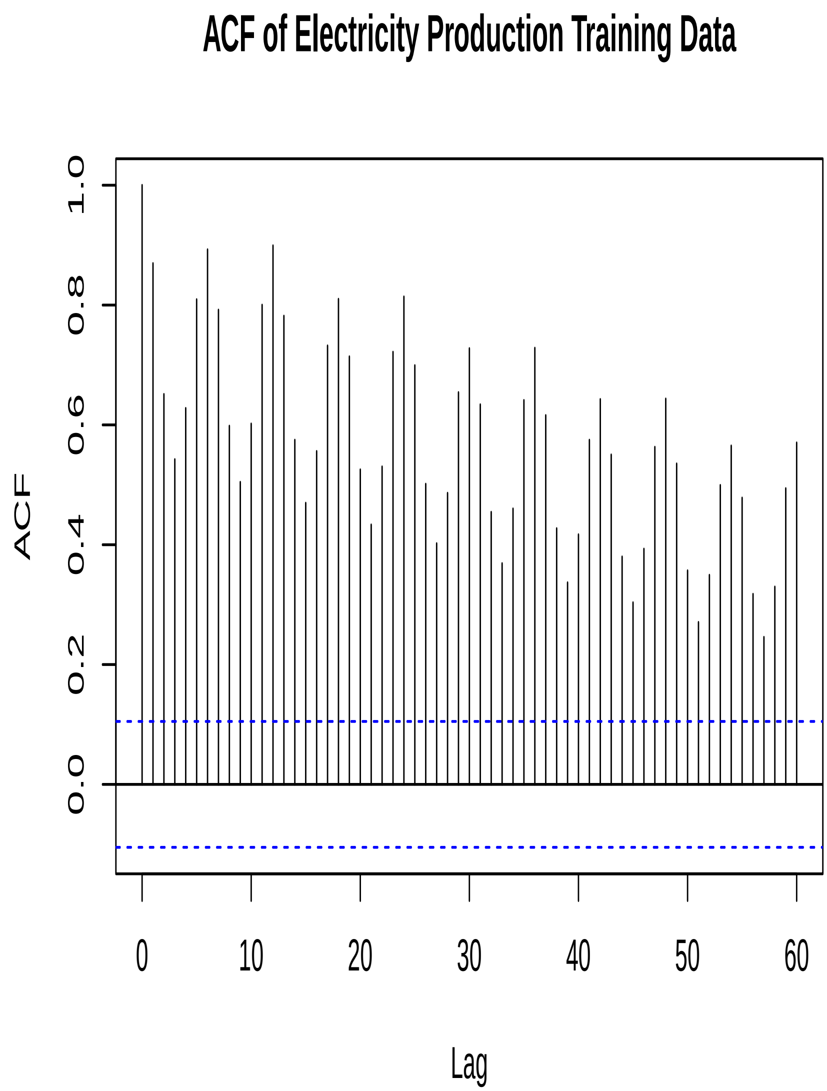
My training and testing datasets include 348 and 50 observations, with a variance of 221.5515 for the training dataset.

Histogram



The histogram shows a distribution of electric production that is approximately symmetric but not perfectly normal. There are peaks in the 85 to 95 range, indicating higher frequencies in that interval.

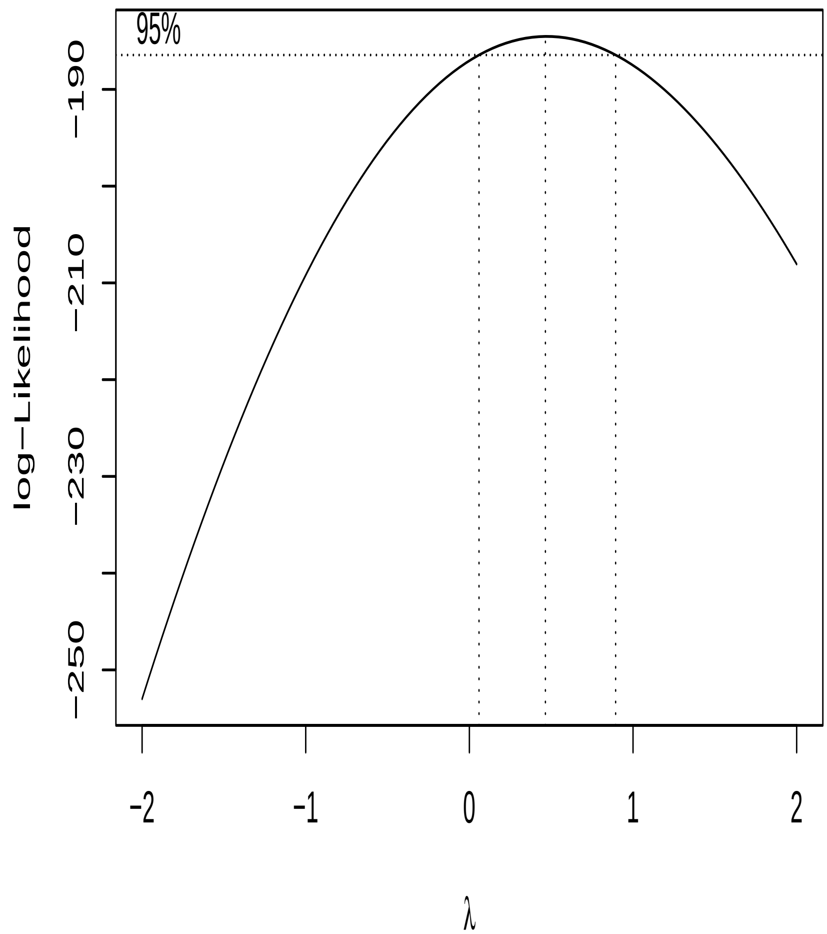
ACF



The ACF plot of the training data, demonstrates large and periodic autocorrelations, suggesting a strong seasonal pattern. The wave like shape indicates repeating occurrences at regular intervals. Additionally, the slow decay of the autocorrelations indicate the presence of trend.

Box-Cox Tranformation

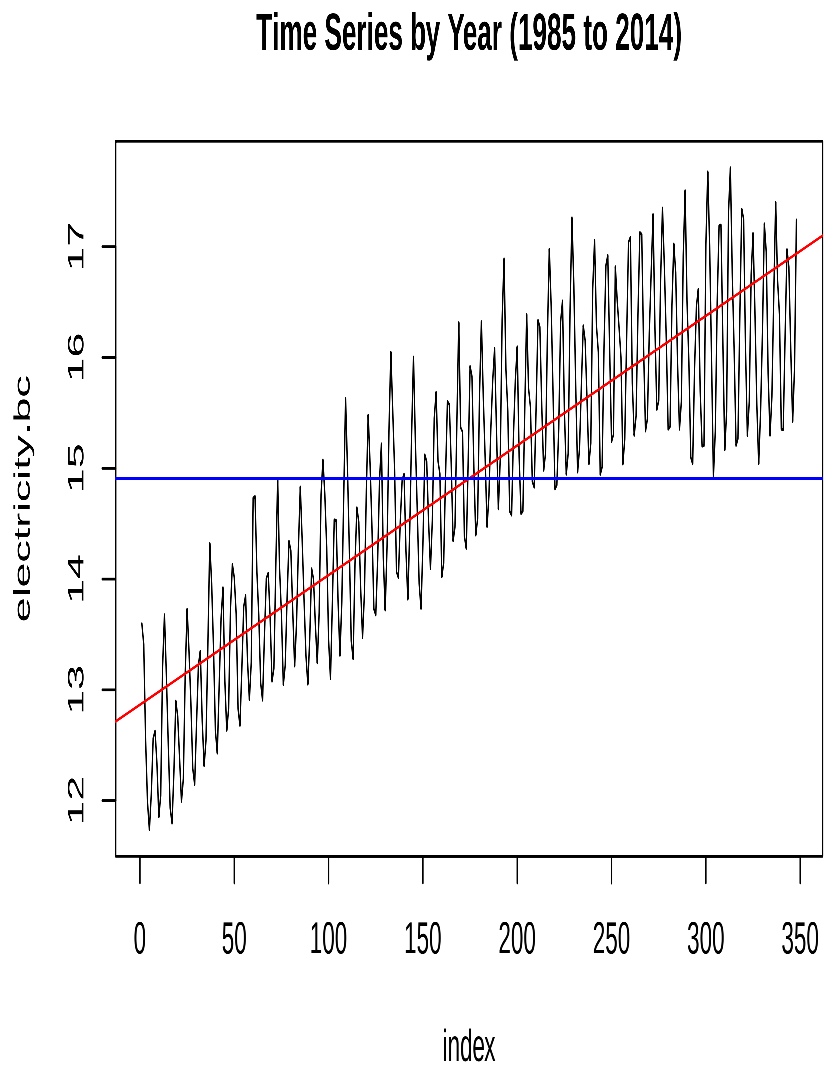
The training data demonstrated variability in production and increased over time. To stabilize the variance, a Box-Cox transformation is needed.




```
## [1] "Lambda: 0.464646464646465"
```

The Box-Cox transformation estimates the optimal value of the parameter λ that best normalizes the data. The plot shows the 95% confidence interval for λ to be approximately [0.05, 0.90], and the estimated lambda is 0.4646. An important thing to note is that 0 is not in the confidence interval, making it not appropriate to use a log transformation. Instead, I will be using $\lambda = 0.4646$ to stabilize the variance. My transformation is X $= \frac{U^{0.4646-1}}{0.4646}$.

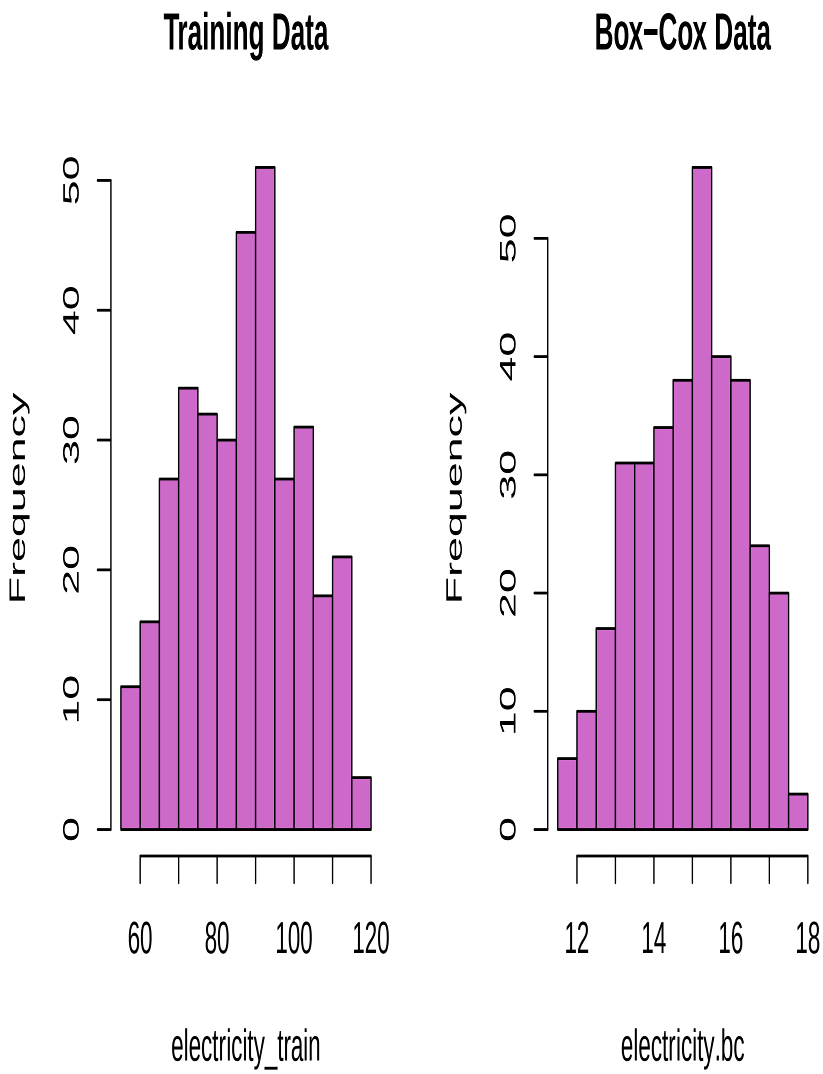
Plotting Box-Cox Transformation




```
## [1] "The variance is: 1.9047406068498"
```

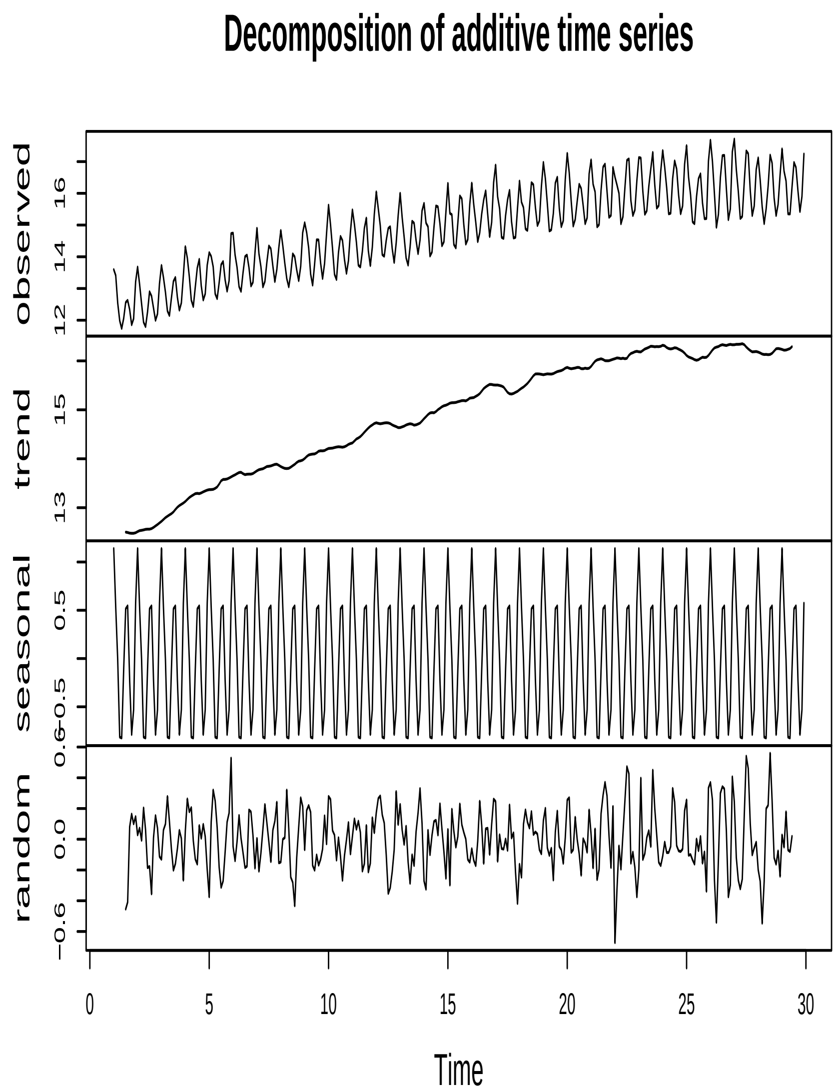
The plot of the transformed training set demonstrates a stabilized variance, 1.905, but the positive trend and seasonal patterns are still present.

Histogram Comparison



The histogram on the left shows the distribution of the original data, while on the right, it shows the Box-Cox transformed data. As mentioned before, the original data shows a non-symmetric distribution. As a result of the Box-Cox transformation, the data demonstrates an approximately symmetric distribution with peaks around 15 to 16, indicating higher frequencies in that interval.

Decomposition of Transformed Data



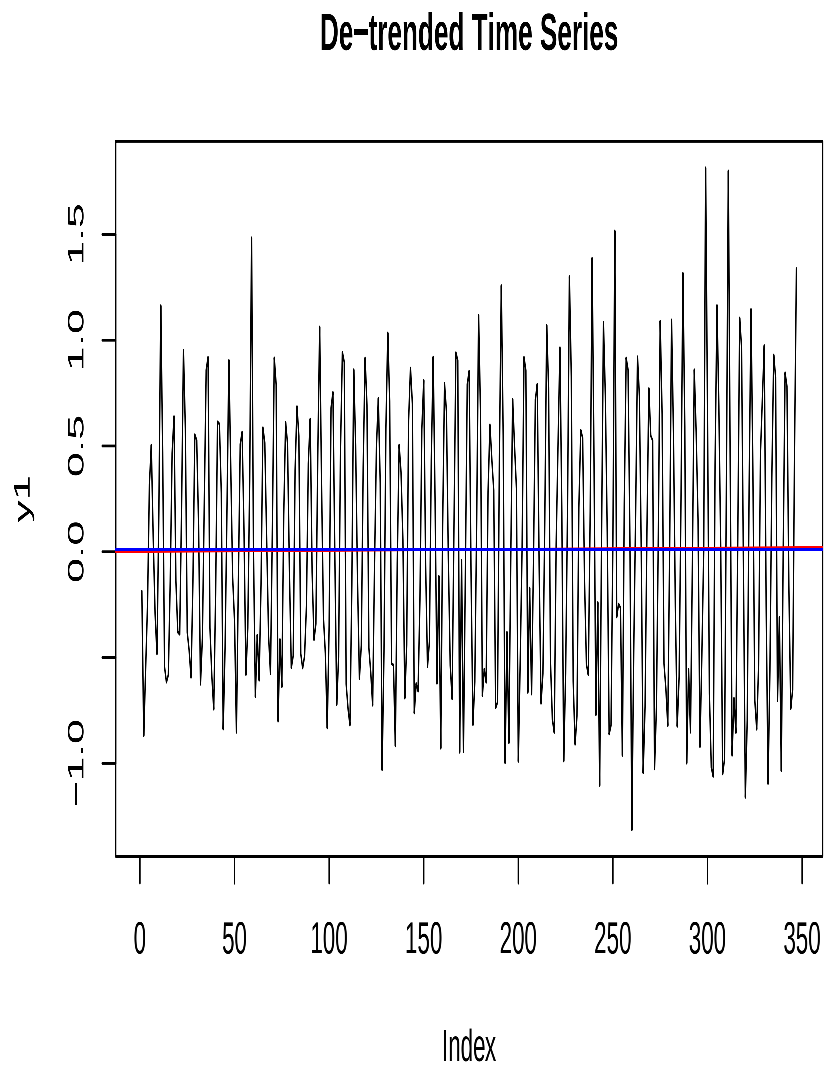
In the decomposition plot, I can see an upward trend, with short-term dips, between approximately time 3 and 23. Additionally, a clear seasonal pattern is present in the plot, the observations regularly go up and down. The variance is relatively stable up until time 22, demonstrating greater fluctuations than before. Therefore, the transformed data is not stationary.

Differencing

The decomposition plot showed non-stationary transformed data, displaying patterns at regular intervals and a positive trend. To make the data stationary, I am going to apply differencing until I achieve stationarity.

Differencing at Lag 1

To remove trend, I am going to difference at lag 1.

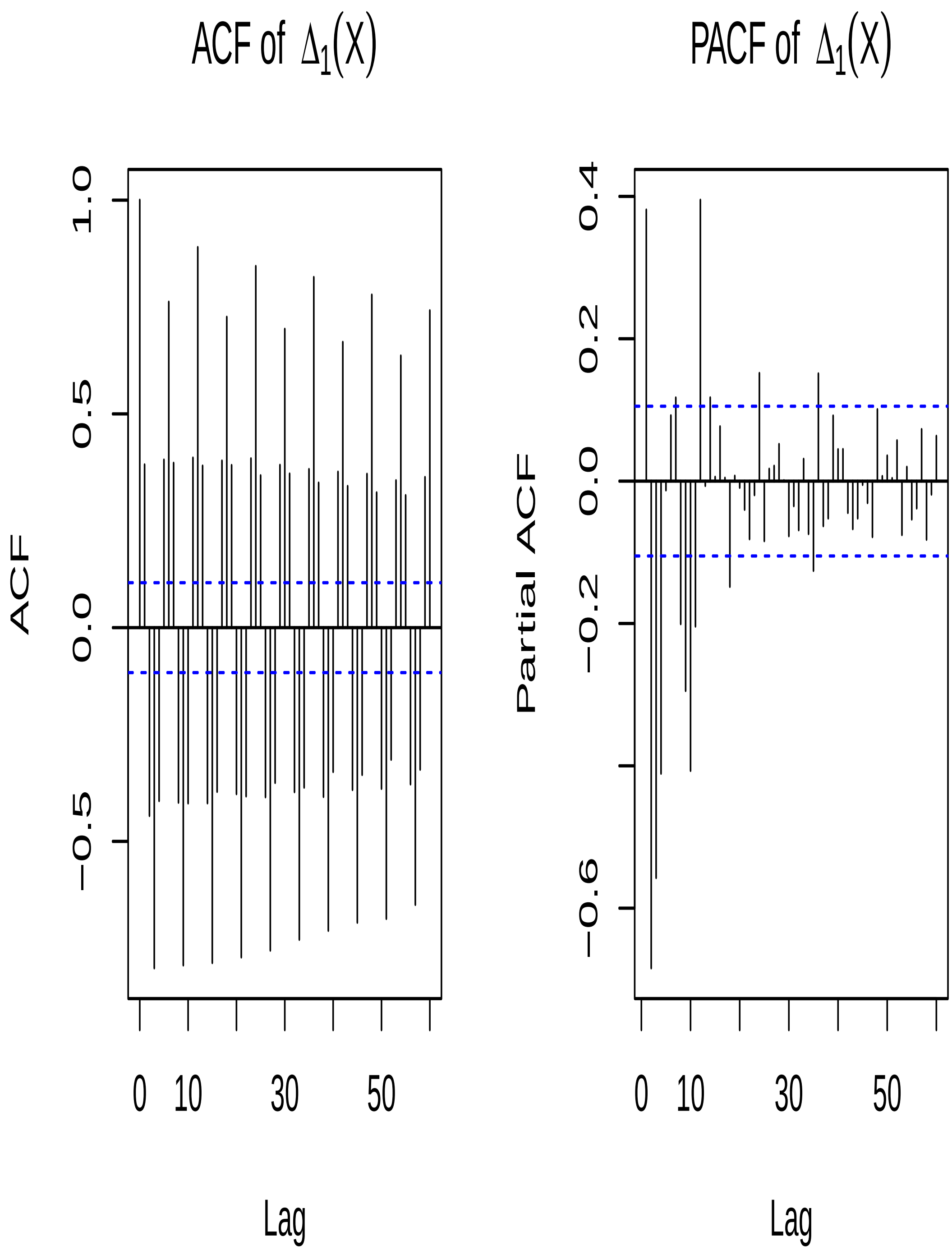



```
## [1] "Variance after differencing at lag 1: 0.442806852937977"
```

In the de-trended time series plot, the positive trend has been removed with a mean approximately equal to 0. As a result of differencing, the variance decreased to 0.4428, compared to the original variance, 1.905. This is a good sign as it indicates over-differencing did not occur. Note the red regression line is very close to the blue horizontal line (mean), indicating that differencing at lag 1 removed the positive linear trend.

ACF & PACF of De-trended Data

I will now check the ACF & PACF to see if the model is stationary or non-stationary.

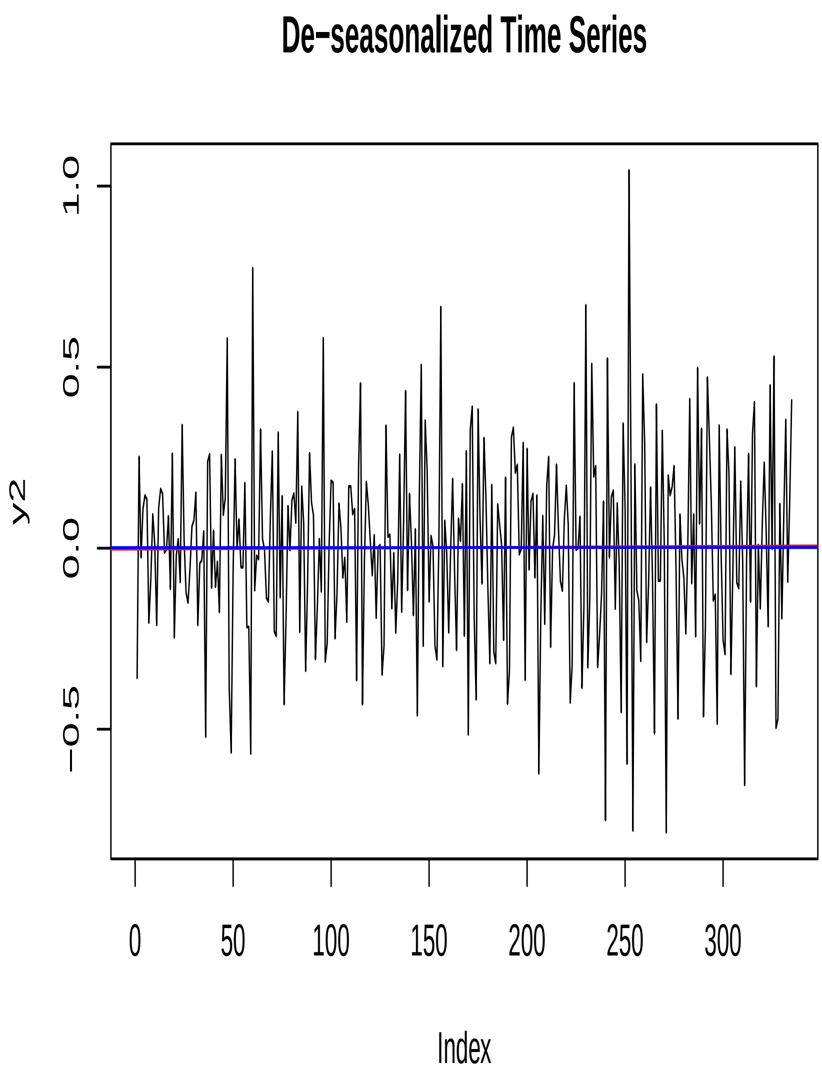


The slow decay of the autocorrelations in the ACF has disappeared, confirming the removal of the trend. However, seasonality is present due to the large autocorrelations and regular patterns in the ACF plot. In the PACF plot, there is no indication of significant lags due to the high number of significant spikes, making interpretation difficult. However, I will try differencing at lag 4 and 12 as they are significant.

Differencing at Lag 12

Differencing at lag 4 gave me a variance of 1.2462, while at lag 12 it is 0.0712. Therefore to remove seasonality, I am going to difference at lag 12.

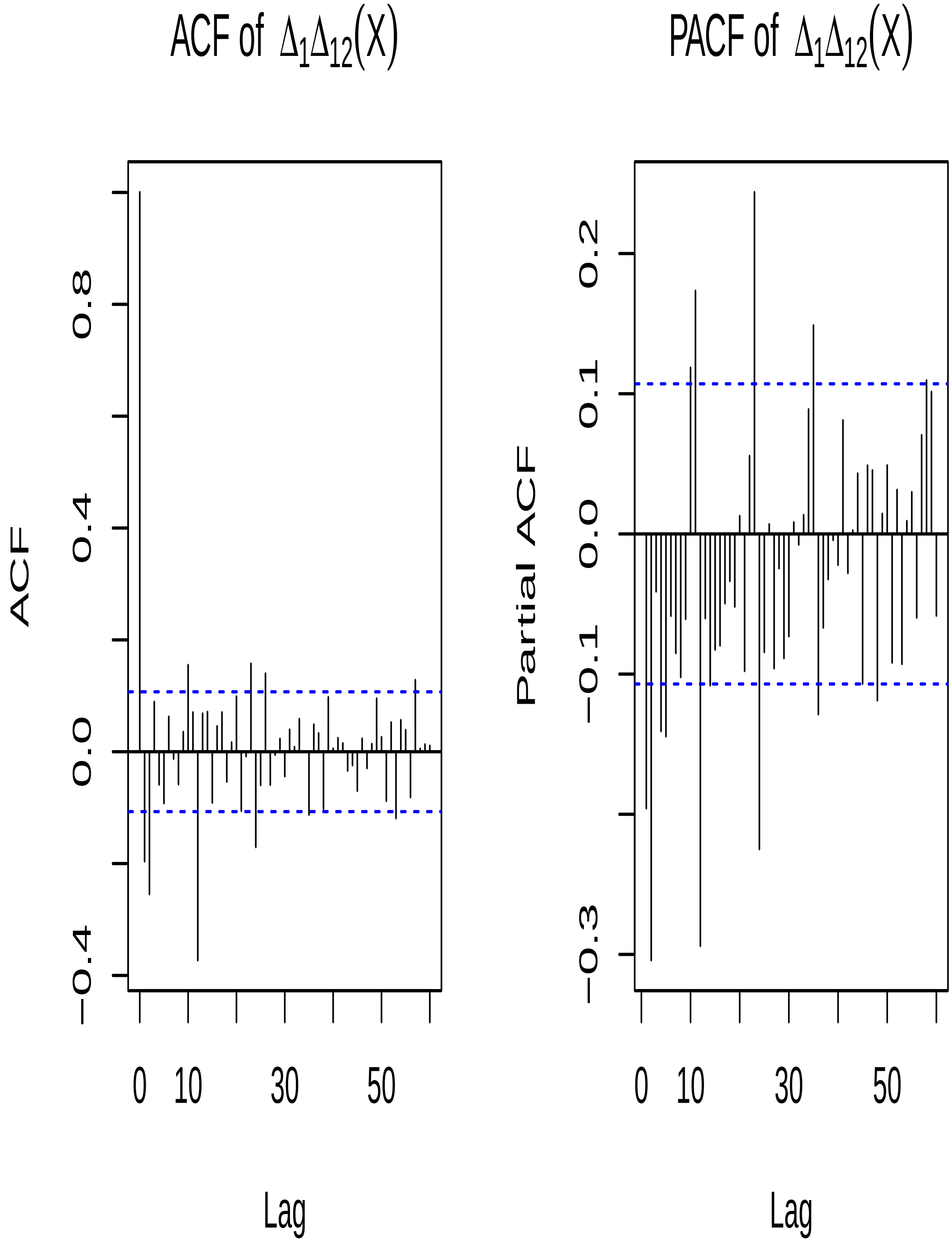
```
## [1] "Variance after differencing at lag 12: 0.0712068212906078"
```



In the de-seasonalized time series plot, the seasonality has been removed with a new variance of 0.0712. The trend and mean line are 0, indicating stationarity.

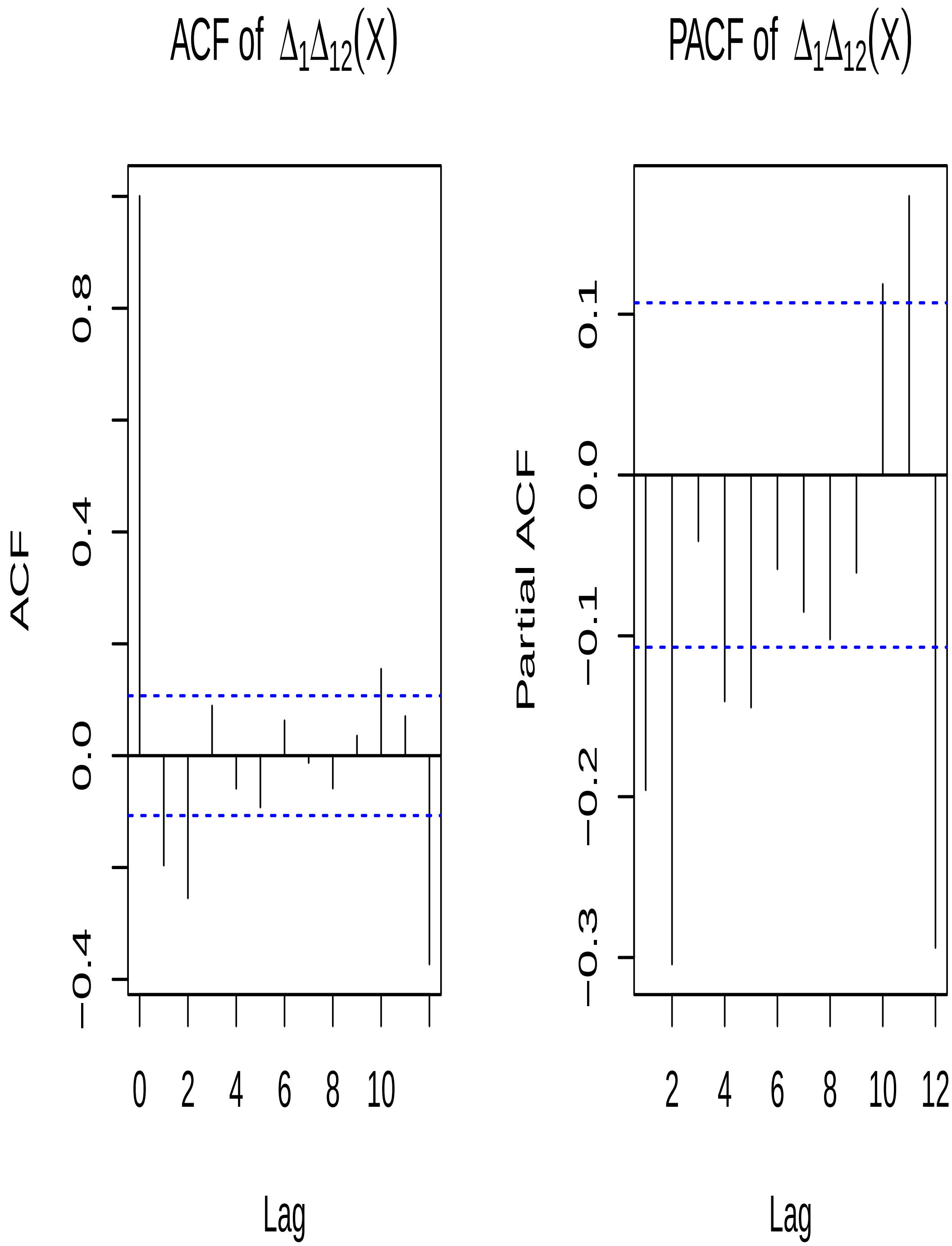
ACF & PACF of De-seasonalized Data

To confirm the model achieved stationarity, I am going to plot the ACF and PACF.



In the ACF plot, there is no trend or seasonality present. However, there are a few significant spikes in both the ACF and PACF plots. I will now choose the candidate seasonal coefficients for the model.

In both the sample ACF and PACF plots, lags 12 and 24 are significant. Therefore, I will choose Q = 1 or 2 for the seasonal moving average (SMA) terms and P = 1 or 2 for the seasonal autoregressive (SAR) terms, as they are seasonal lags (multiples of 12).



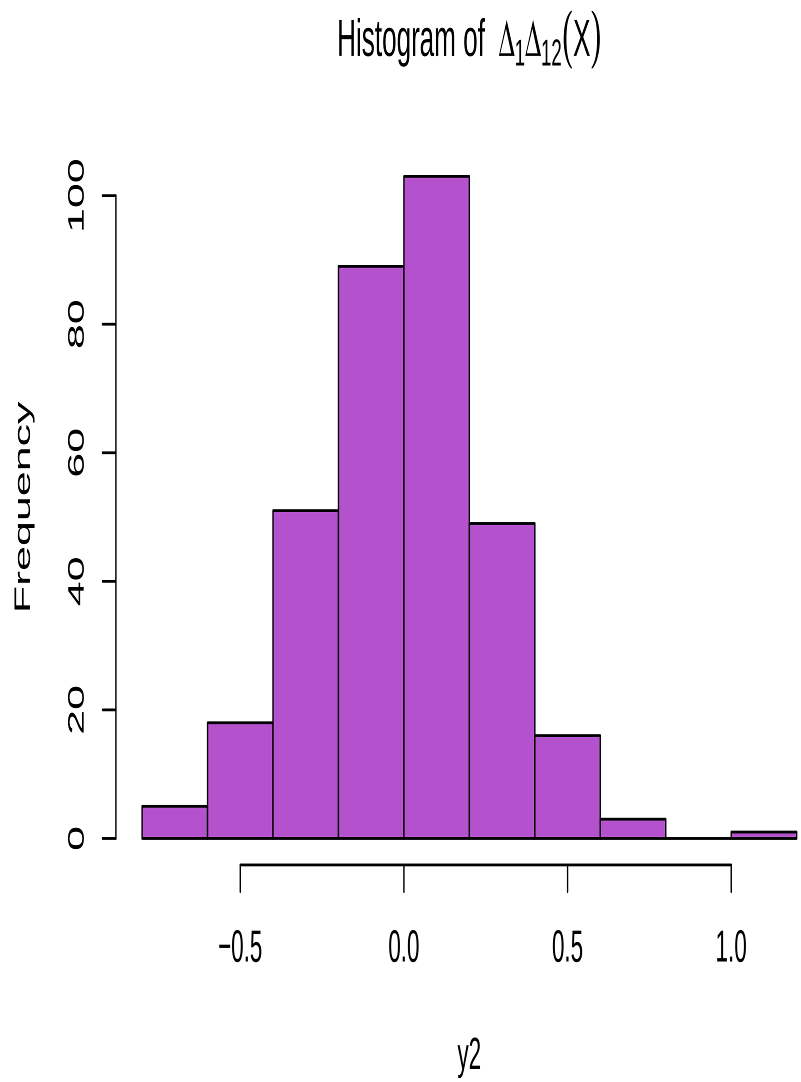
Viewing the sample ACF and PACF plots from within one seasonality, lags 1 and 2 (ACF) and 2, 4, and 5 (PACF) are significant. Therefore, I will choose p = 2, 4, or 5 and q = 1 or 2 for the non-seasonal autoregressive and moving average terms.

List of Candidate Models:

My SARIMA model would be the following:

SARIMA: p = 2, 4, or 5, d = 1, q = 1 or 2; P = 1 or 2, D = 1, Q = 1 or 2, s = 12.

Histogram of De-seasonalized Data



The de-seasonalized histogram demonstrates a slight left skewed distribution with an outlier greater than 1.0.

Models with Lowest AICc

I tested different models such as:

$$
\begin{aligned}
SARIMA(2,1,1) \times (1,1,1)_{12} \\
SARIMA(2,1,1) \times (2,1,1)_{12} \\
SARIMA(2,1,1) \times (2,1,2)_{12} \\
SARIMA(2,1,2) \times (2,1,2)_{12} \\
SARIMA(4,1,1) \times (1,1,1)_{12} \\
SARIMA(4,1,1) \times (2,1,1)_{12} \\
SARIMA(4,1,2) \times (1,1,2)_{12} \\
SARIMA(4,1,2) \times (2,1,2)_{12} \\
SARIMA(5,1,1) \times (1,1,1)_{12} \\
SARIMA(5,1,1) \times (2,1,1)_{12} \\
SARIMA(5,1,2) \times (1,1,2)_{12} \\
SARIMA(5,1,2) \times (2,1,2)_{12} \\
...
\end{aligned}
$$

The following SARIMAs are the top 3 with the lowest AICc.

$SARIMA(2,1,1) \times (2,1,1)_{12}$

```
##
## Call:
## arima(x = electricity.bc, order = c(2, 1, 1), seasonal = list(order = c(2, 1,
##     1), period = 12), method = "ML")
##
## Coefficients:
##          ar1      ar2      ma1     sar1     sar2     sma1
##       0.5349  -0.0620  -0.9262  -0.0312  -0.1870  -0.7076
## s.e.  0.0599   0.0587   0.0277   0.0750   0.0658   0.0605
##
## sigma^2 estimated as 0.03583:  log likelihood = 75.43,  aic = -136.85
```

```
## [1] -136.6063
```

My initial model for $SARIMA(2,1,1) \times (2,1,1)_{12}$ included all the coefficients and had an AICc of -136.6063. However, AR(2) and SAR(1) had confidence intervals that included 0, indicating they may not be statistically significant. This means there is a possibility AR(2) and SAR(1) could be 0.

```
##
## Call:
## arima(x = electricity.bc, order = c(2, 1, 1), seasonal = list(order = c(2, 1,
##     1), period = 12), fixed = c(NA, 0, NA, 0, NA, NA))
##
## Coefficients:
##          ar1  ar2      ma1  sar1     sar2     sma1
##       0.5178    0  -0.9360     0  -0.1786  -0.7181
## s.e.  0.0554    0   0.0224     0   0.0608   0.0431
```

```
##
## sigma^2 estimated as 0.03598:  log likelihood = 74.78,  aic = -139.56
```

```
## [1] -139.3177
```

To test this, I re-ran the model with AR(2) and SAR(1) fixed to zero. The result had a lower AICc value of -139.3177.

$SARIMA(4,1,1) \times (2,1,1)_{12}$

```
##
## Call:
## arima(x = electricity.bc, order = c(4, 1, 1), seasonal = list(order = c(2, 1,
##     1), period = 12), method = "ML")
##
## Coefficients:
##          ar1      ar2     ar3      ar4      ma1     sar1     sar2     sma1
##       0.5355  -0.0848  0.0615  -0.0583  -0.9237  -0.0302  -0.1783  -0.6998
## s.e.  0.0629   0.0636  0.0667   0.0593   0.0336   0.0779   0.0673   0.0637
##
## sigma^2 estimated as 0.03574:  log likelihood = 76.08,  aic = -134.17
```

```
## [1] -133.7444
```

For the model $SARIMA(4,1,1) \times (2,1,1)_{12}$, all the coefficients were included, resulting in an AICc value of -133.7444. However, AR(2), AR(3), AR(4), and SAR(1) had confidence intervals that included 0, indicating they might not be statistically significant.

```
##
## Call:
## arima(x = electricity.bc, order = c(4, 1, 1), seasonal = list(order = c(2, 1,
##     1), period = 12), fixed = c(NA, 0, 0, 0, NA, 0, NA, NA))
##
## Coefficients:
##          ar1  ar2  ar3  ar4      ma1  sar1     sar2     sma1
##       0.5177    0    0    0  -0.9360     0  -0.1787  -0.7181
## s.e.  0.0554    0    0    0   0.0224     0   0.0608   0.0431
##
## sigma^2 estimated as 0.03598:  log likelihood = 74.78,  aic = -139.56
```

```
## [1] -139.1392
```

I re-ran the model fixing AR(2), AR(3), AR(4), and SAR(1) to zero, which led to a lower AICc value of -139.1392.

$SARIMA(5,1,1) \times (2,1,1)_{12}$

```
##
## Call:
## arima(x = electricity.bc, order = c(5, 1, 1), seasonal = list(order = c(2, 1,
```

```
##     1), period = 12), method = "ML")
##
## Coefficients:
##          ar1      ar2     ar3      ar4     ar5      ma1     sar1     sar2
##       0.5372  -0.0848  0.0627  -0.0609  0.0067  -0.9251  -0.0293  -0.1777
## s.e.  0.0628   0.0638  0.0679   0.0632  0.0588   0.0352   0.0788   0.0676
##          sma1
##       -0.7005
## s.e.   0.0641
##
## sigma^2 estimated as 0.03574:  log likelihood = 76.09,  aic = -132.18
```

```
## [1] -131.6495
```

Continuing with the model $SARIMA(5,1,1) \times (2,1,1)_{12}$, all the coefficients were included with an AICc value of -131.6495. However, the confidence intervals for AR(2), AR(3), AR(4), AR(5), and SAR(1) included 0.

```
##
## Call:
## arima(x = electricity.bc, order = c(5, 1, 1), seasonal = list(order = c(2, 1,
##     1), period = 12), fixed = c(NA, 0, 0, 0, 0, NA, 0, NA, NA), method = "ML")
##
## Coefficients:
##          ar1  ar2  ar3  ar4  ar5      ma1  sar1     sar2     sma1
##       0.5178    0    0    0    0  -1.0683     0  -0.1786  -0.7181
## s.e.  0.0554    0    0    0    0   0.0256     0   0.0608   0.0431
##
## sigma^2 estimated as 0.03153:  log likelihood = 74.78,  aic = -139.56
```

```
## [1] -139.0316
```

After fixing AR(2), AR(3), AR(4), AR(5), and SAR(1) to 0, the AICc improved to -139.0316.

Model C is not invertible because non-seasonal MA is greater than 1, therefore I will not consider Model C.

Models

Here are the following models with white noise variance.

Model A:

$$(1-B)(1-B^{12})(1-0.5178B)(1+0.1786B^{24})X_t = (1-0.9360B)(1-0.7181B^{12})Z_t$$
$$\hat{\sigma}_Z^2 = 0.03598$$

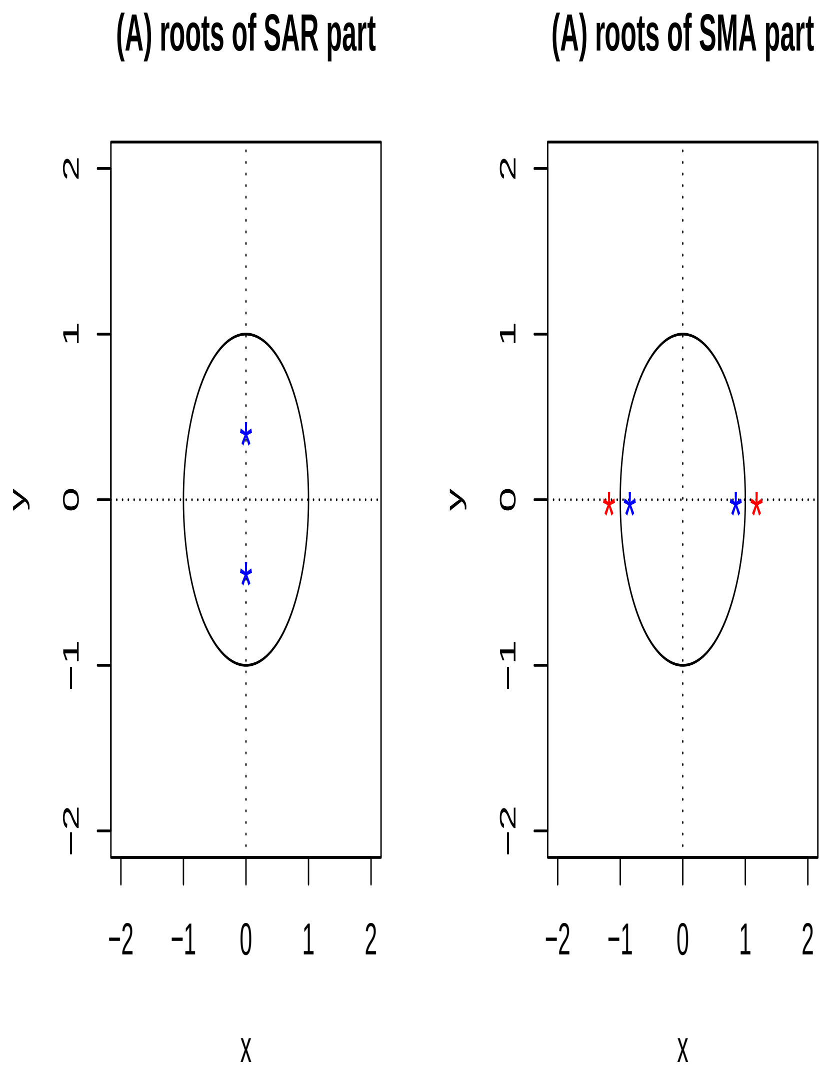
Model B:

$$(1-B)(1-B^{12})(1-0.5177B)(1+0.1787B^{24})X_t = (1-0.9360B)(1-0.7181B^{12})Z_t$$
$$\hat{\sigma}_Z^2 = 0.03598$$

These following models show almost identical equations with an estimated variance, $\hat{\sigma}_Z^2 = 0.03598$, suggesting a good fit.

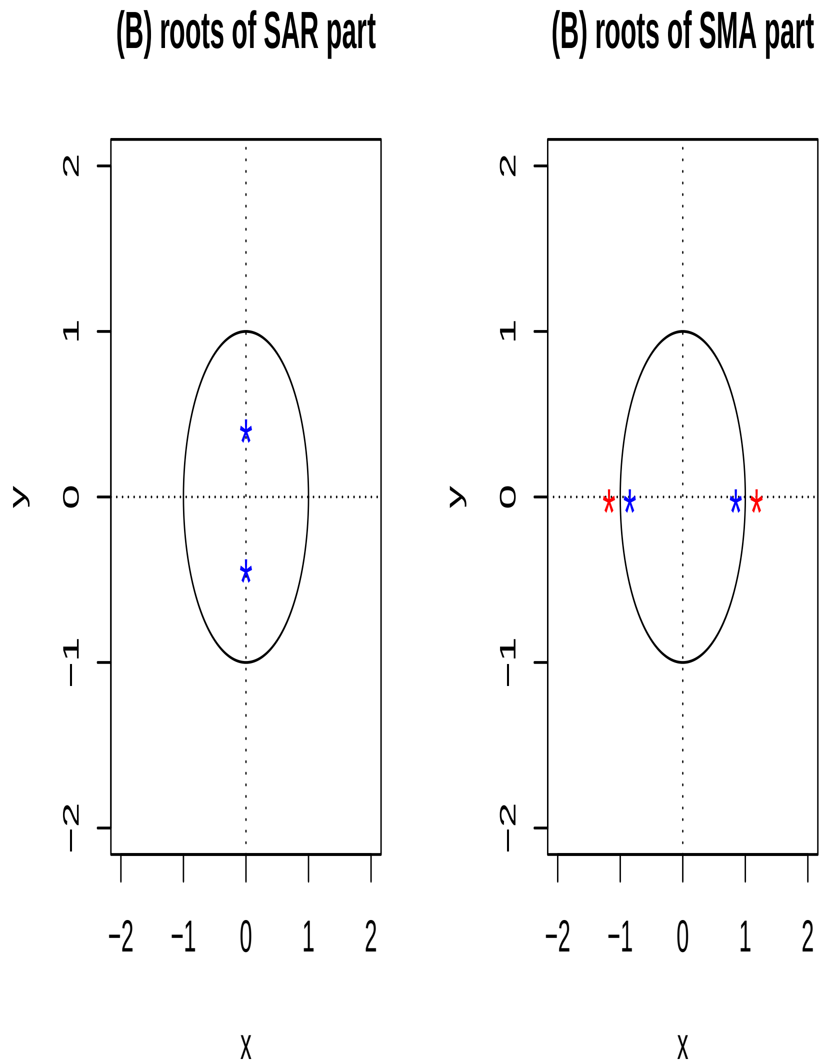
Stationary and Invertibility Check

The following plots demonstrate whether the red points are outside or inside of the unit circle.



Model A: $(1-B)(1-B^{12})(1-0.5178B)(1+0.1786B^{24})X_t = (1-0.9360B)(1-0.7181B^{12})Z_t$

Model A is stationary and invertible as the non-seasonal AR and MA parts are less than 1 and for seasonal AR and MA the red points are outside the unit circle.



Model B: $(1-B)(1-B^{12})(1-0.5177B)(1+0.1787B^{24})X_t = (1-0.9360B)(1-0.7181B^{12})Z_t$

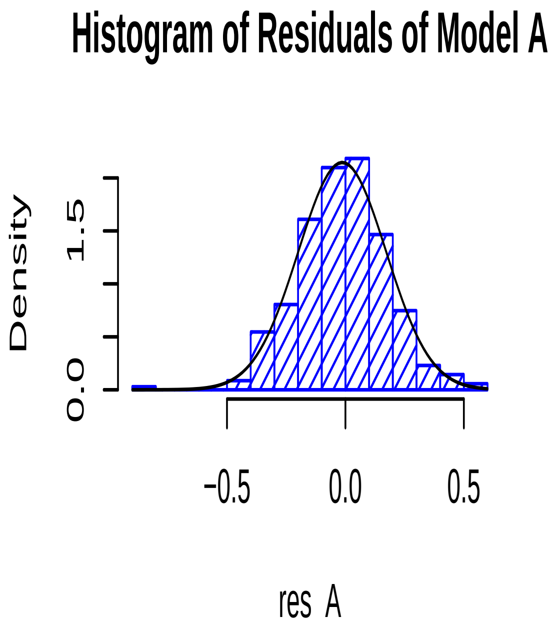
Model B shows the same pattern as Model A, non-seasonal AR and MA are less than 1 and for seasonal AR and MA the red points are outside the unit circle. This indicates Model B as stationary and invertible.

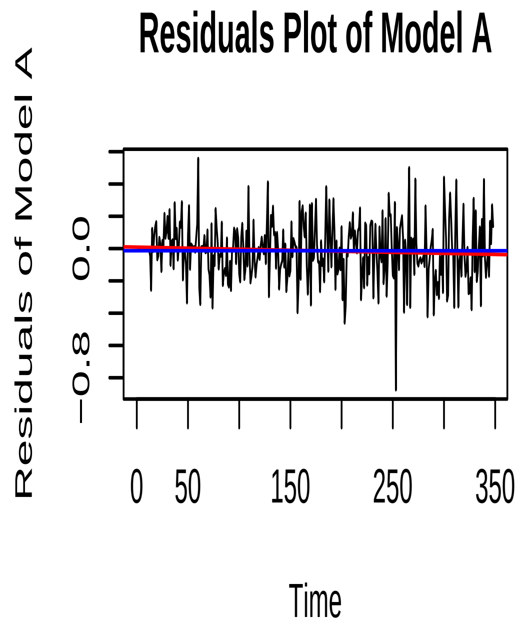
Diagnostic Check

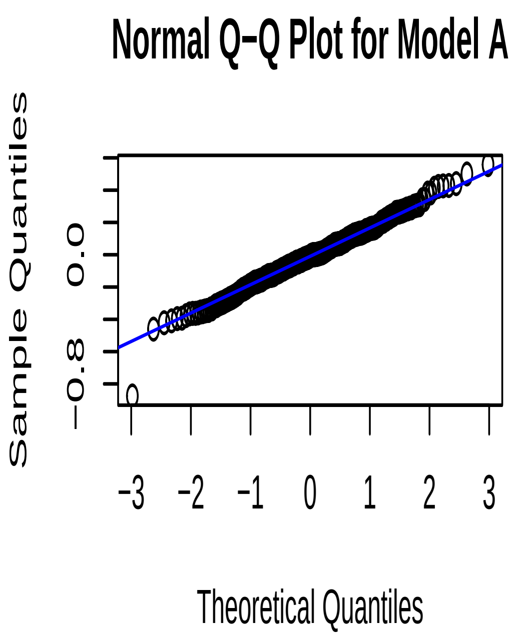
Diagnostic checking is used to check whether the models give a satisfactory representation of the data or if one model fits the data better than the others.

Model A

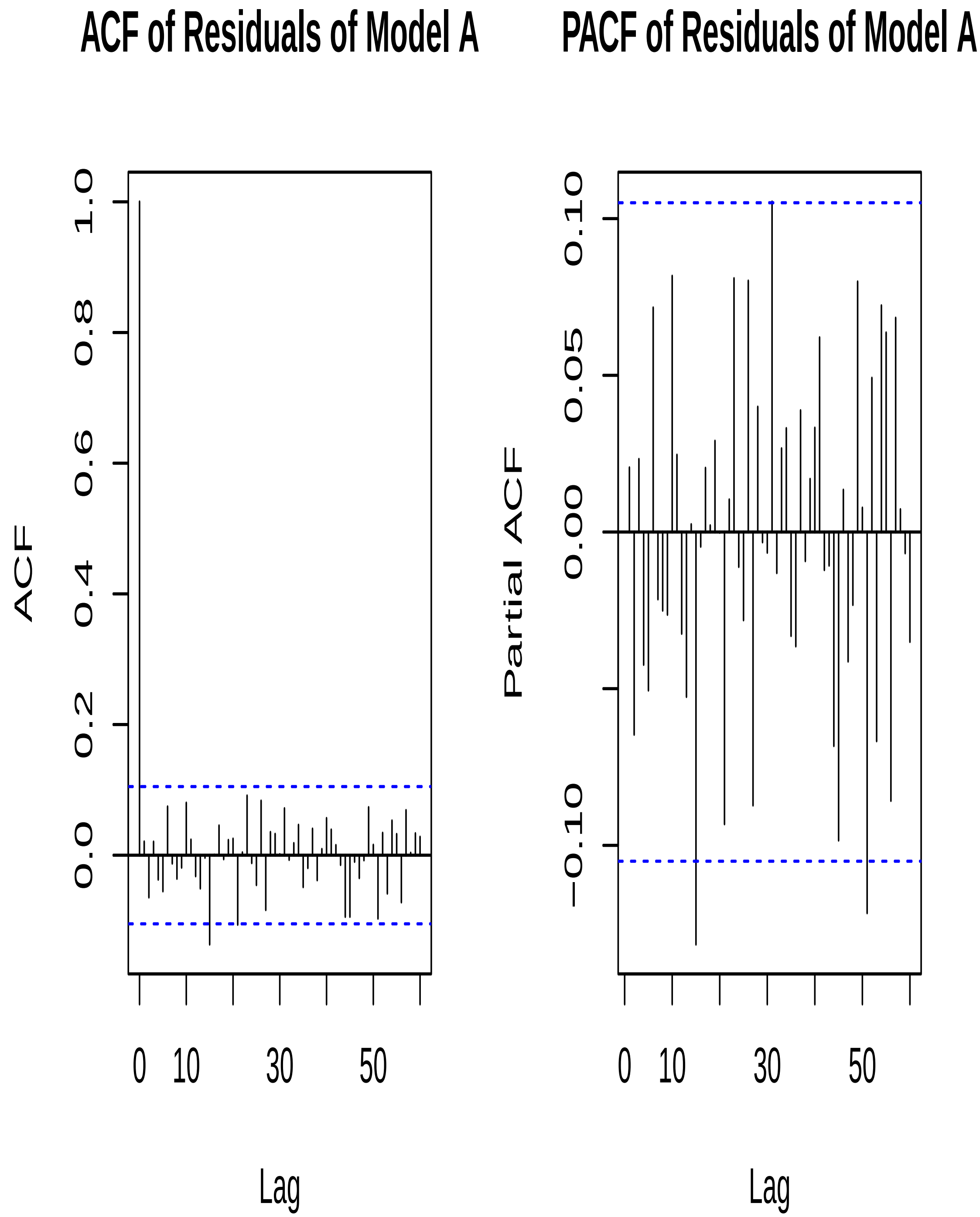
$(1-B)(1-B^{12})(1-0.5178B)(1+0.1786B^{24})X_t = (1-0.9360B)(1-0.7181B^{12})Z_t$







The following plots demonstrate no trend, seasonality, or visible change of variance. The histogram and Q-Q plot look ok. However, there seems to be an outlier present in all of the graphs.



Most of the lags in the ACF and PACF of the residuals are within the confidence intervals. However, there are a few spikes outside. To confirm if the model is adequate, I will perform the following tests.

As a reminder, my training set has 348 observations, making my lag be approximately 19.

Shapiro-Wilk Test

The Shapiro-Wilk test is used to test how close the data is to Gaussian.

```
##
##  Shapiro-Wilk normality test
##
## data:  res_A
## W = 0.99032, p-value = 0.0216
```

My p-value is 0.0216, therefore I reject the hypothesis that the data comes from a Gaussian distribution. This suggests my data is not consistent with a normal distribution. Considering the histogram and Q-Q plot from the previous section, I would assume the outlier is causing the model to fail the Shapiro-Wilk test.

Box-Pierce Test

The Box-Pierce tests whether the residuals are correlated.

```
##
##  Box-Pierce test
##
## data:  res_A
## X-squared = 16.791, df = 15, p-value = 0.3315
```

My p-value is 0.3315, therefore I do not reject the hypothesis that the residuals are uncorrelated.

Box-Ljung Test

The Box-Ljung tests for linear dependence.

```
##
##  Box-Ljung test
##
## data:  res_A
## X-squared = 17.428, df = 15, p-value = 0.294
```

My p-value is 0.2940, therefore I do not reject the hypothesis that my residuals are linearly independent.

McLeod-Li Test

The McLeod-Li tests whether the squared residuals are correlated.

```
##
##  Box-Ljung test
##
## data:  (res_A)^2
## X-squared = 21.149, df = 19, p-value = 0.3286
```

My p-value is 0.3286, therefore I do not reject the hypothesis that the squared residuals are uncorrelated.

Yule-Walker

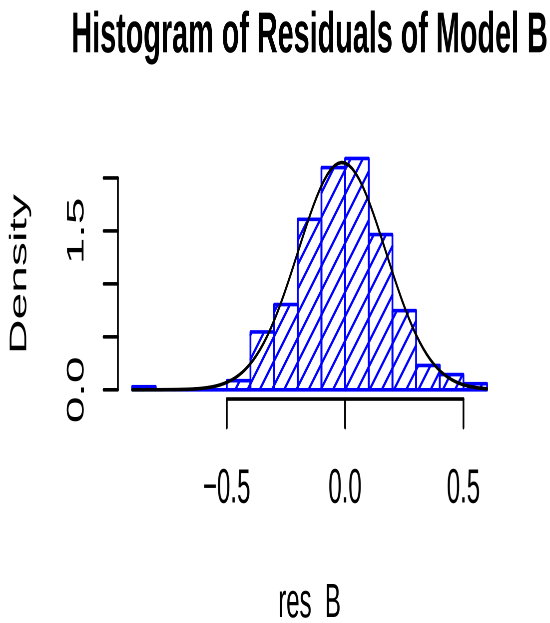
This tests whether the residuals of the model contain an autoregressive structure. The estimated variance of the residuals quantifies how much unexplained variance remains after fitting to the AR structure.

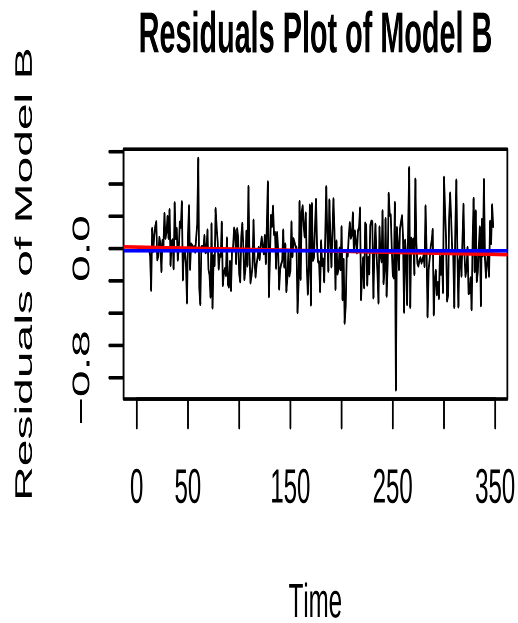
```
##
## Call:
## ar(x = res_A, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  0.03456
```

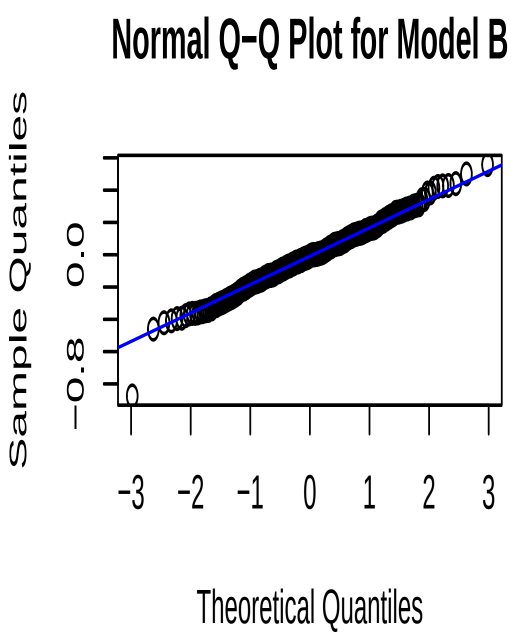
The estimated variance of the residuals is 0.03456, indicating the AR model fits the series well.

Model B

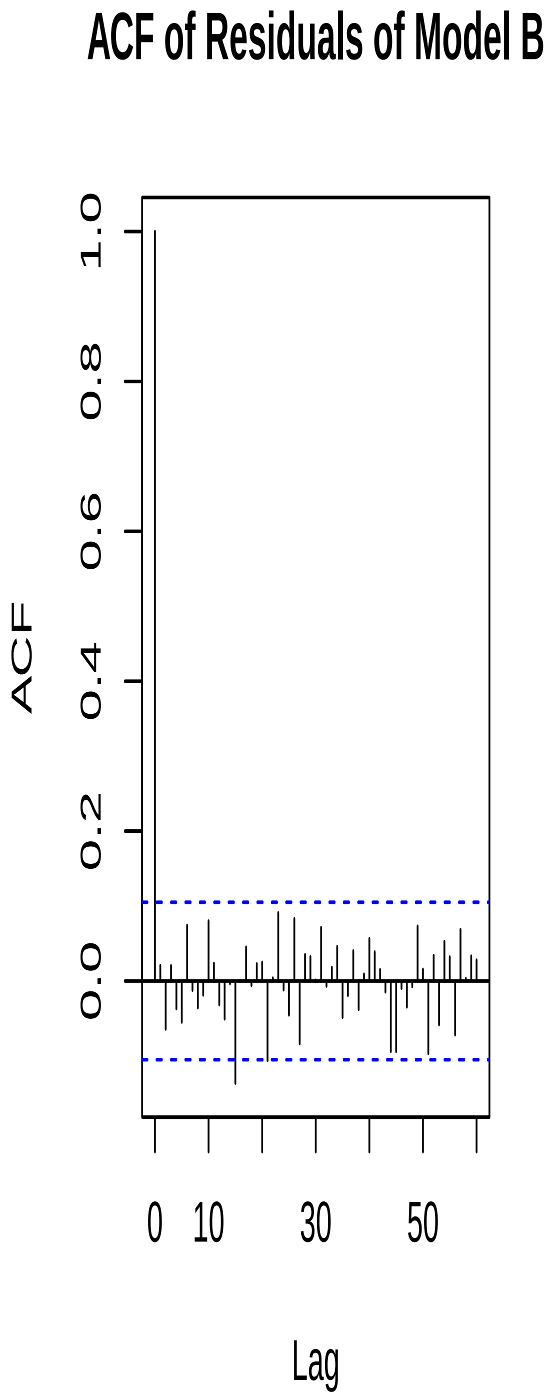
$(1-B)(1-B^{12})(1-0.5177B)(1+0.1787B^{24})X_t = (1-0.9360B)(1-0.7181B^{12})Z_t$

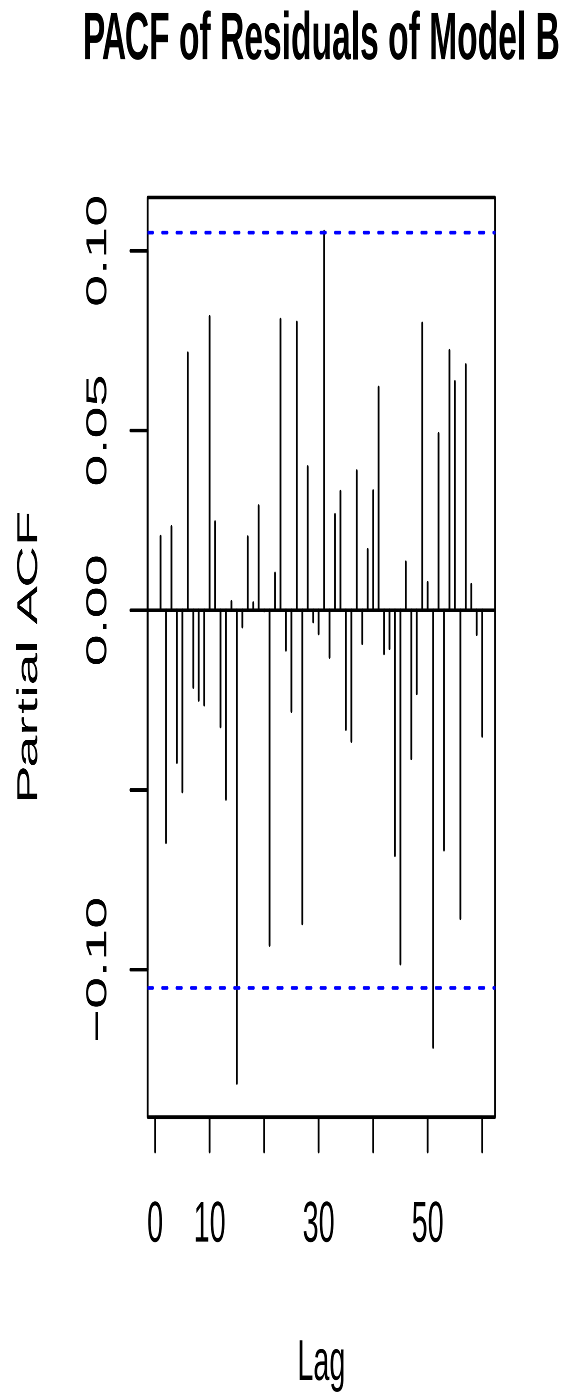






Model B's plots demonstrate no trend, seasonality, or visible change of variance. Both the histogram and Q-Q plot look ok, but with an outlier present.





Both the ACF and PACF plots show some lags outside the confidence interval. Therefore, I will conduct tests to see if the model is adequate.

Shapiro-Wilk Test

```
##
##  Shapiro-Wilk normality test
##
## data:  res_B
## W = 0.99033, p-value = 0.0216
```

Model B fails the Shapiro-Wilk test with a p-value of 0.0216, the data is not consistent with a normal distribution. Once more, I would assume the outlier present in the histogram and Q-Q plot is causing the model to fail the Shapiro-Wilk test.

Box-Pierce Test

```
##
##  Box-Pierce test
##
## data:  res_B
## X-squared = 16.79, df = 15, p-value = 0.3316
```

The model passes the Box-Pierce test with a p-value of 0.3316, indicating the residuals are uncorrelated.

Box-Ljung Test

```
##
##  Box-Ljung test
##
## data:  res_B
## X-squared = 17.426, df = 15, p-value = 0.294
```

The model passes the Box-Ljung test with a p-value of 0.2940, indicating the residuals are linearly independent.

McLeod-Li Test

```
##
##  Box-Ljung test
##
## data:  (res_B)^2
## X-squared = 21.149, df = 19, p-value = 0.3286
```

Model B passes the McLeod-Li test with a p-value of 0.3286, indicating the squared residuals are uncorrelated.

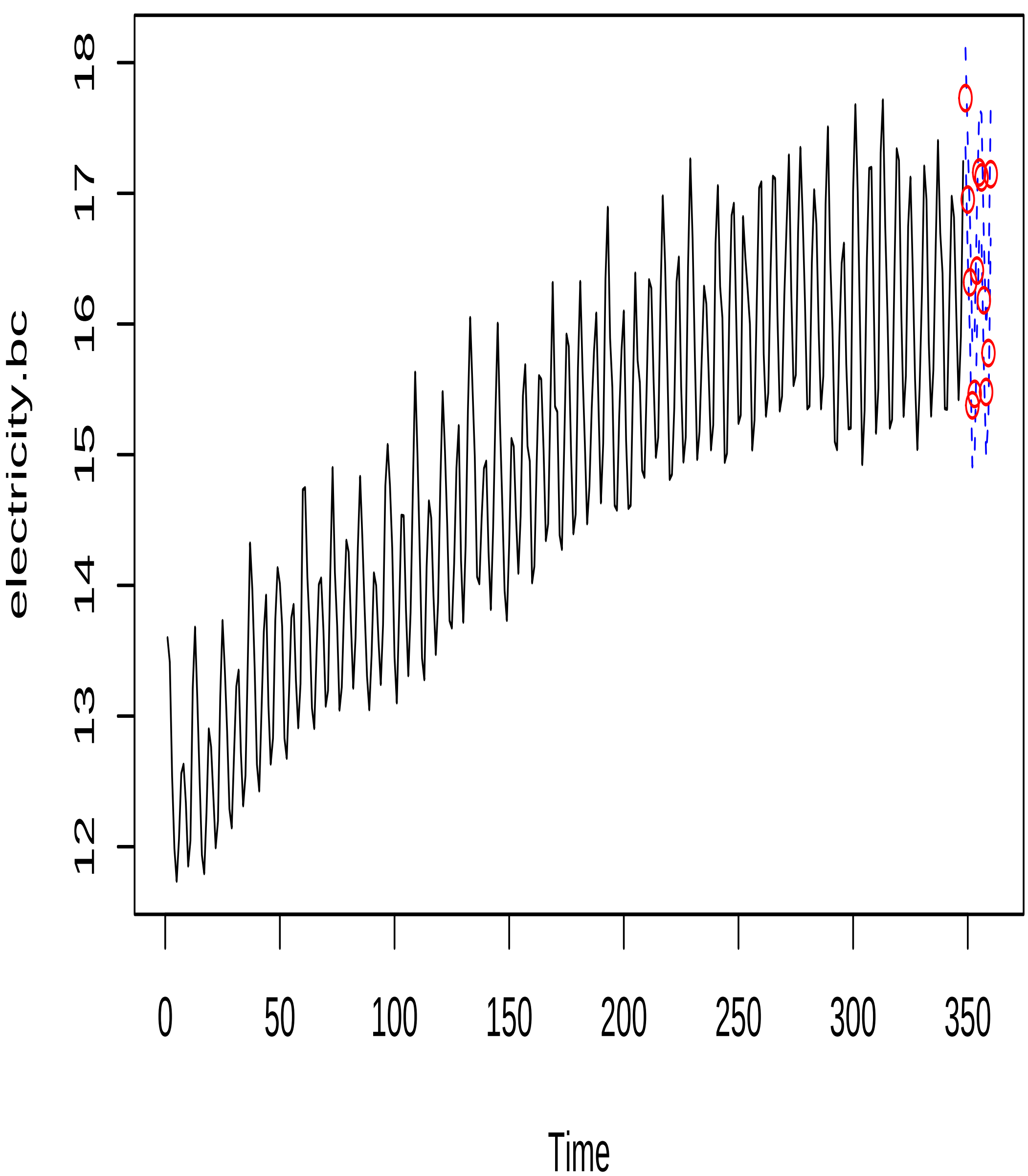
```
##
## Call:
## ar(x = res_B, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  0.03456
```

The estimated variance of the residuals is 0.03456, indicating the AR model fits well.

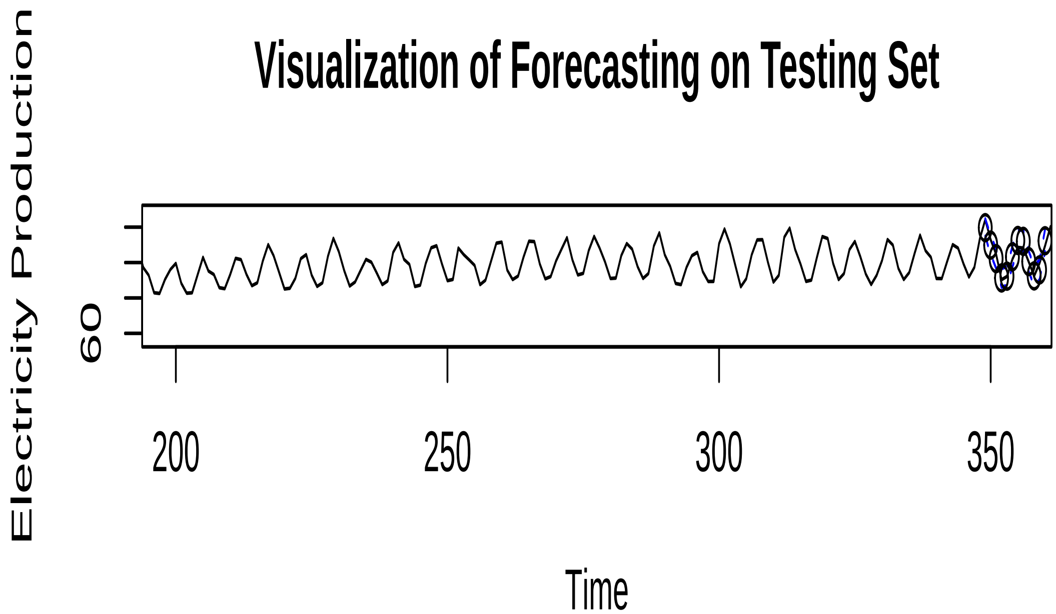
Model A and B passed all diagnostic tests except for the Shapiro-Wilk test. Both Model A and B have 4 non-zero coefficients, but Model A has the lowest AICc. Therefore, Model A will be used as the final model.

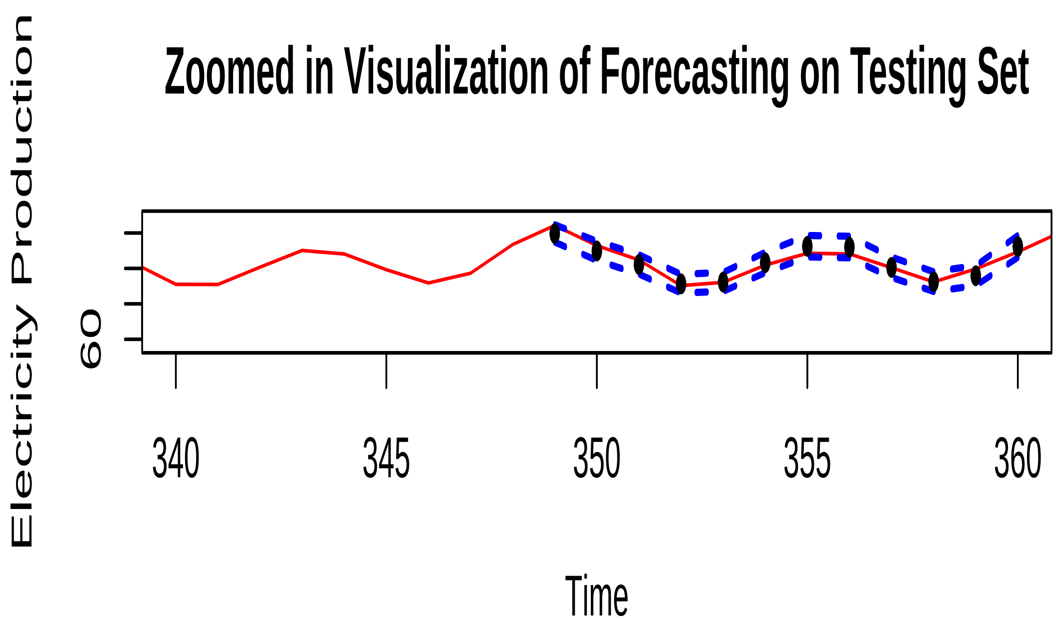
Forecasting

I will now use Model A to forecast future values in the time series. The forecast will be applied to the transformed data and then use the original data to interpret results.



The above plot shows the forecast of the transformed data. The true values appear to fall within the forecast's 95% confidence intervals, suggesting the model is capturing the historical data's structure fairly well.





The above plots demonstrate the predicted values over the testing dataset. Overall, the forecasts closely follow the true values and remain within the 95% confidence intervals, indicating good predictive performance. Model A appears to be a great model for forecasting electric production.

Conclusion

The initial electric production dataset included trend and seasonality. To address this, I performed a Box-Cox transformation to help normalize the data and stabilize the variance. Following that, I applied differencing to remove both seasonality and trend. I then explored various SARIMA models by testing different parameters. The top 3 models were selected based on low AICc values. This led to the selection of the final model based on stationarity, invertibility, diagnostic checking, and a low AICc value. As a result, the final model was $SARIMA(2,1,1) \times (2,1,1)_{12}$ with AR(1) and SAR(1) fixed to zero (Model A).

The equation for Model A, $(1-B)(1-B^{12})(1-0.5178B)(1+0.1786B^{24})X_t = (1-0.9360B)(1-0.7181B^{12})Z_t$, was used to forecast values using the transformed data, showing the true values within the 95% confidence intervals. As a result, the model was then used to forecast values on the testing dataset, demonstrating the values closely follow the true values.

Overall, this model would be helpful in forecasting values of electric production, achieving my goal. This is important as it can help society lower their electricity usage. As a result, help to reduce the effect of climate change and decrease water use.

I would like to thank Isaiah, Lihao, and Professor Feldman in helping me produce this project.

References

Hyndman R, Athanasopoulos G, Bergmeir C, Caceres G, Chhay L, O'Hara-Wild M, Petropoulos F, Razbash S, Wang E, Yasmeen F (2025). *forecast: Forecasting functions for time series and linear models*. R package version 8.24.0, https://pkg.robjhyndman.com/forecast/.

Hyndman RJ, Khandakar Y (2008). "Automatic time series forecasting: the forecast package for R." *Journal of Statistical Software*, *27*(3), 1-22. doi:10.18637/jss.v027.i03 https://doi.org/10.18637/jss.v027.i03.

R Core Team (2024). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/.

Shenbagakumars. Kaggle. Time Series Datasets. Version 1.0, Kaggle, https://www.kaggle.com/datasets/shenba/time-series-datasets?resource=download

Spiess A (2018). *qpcR: Modelling and Analysis of Real-Time PCR Data*. R package version 1.4-1, https://CRAN.R-project.org/package=qpcR.

Stoffer D (2025). *astsa: Applied Statistical Time Series Analysis*. R package version 2.2, https://CRAN.R-project.org/package=astsa.

Venables, W. N. & Ripley, B. D. (2002) Modern Applied Statistics with S. Fourth Edition. Springer, New York. ISBN 0-387-95457-0

Wickham H, Averick M, Bryan J, Chang W, McGowan LD, François R, Grolemund G, Hayes A, Henry L, Hester J, Kuhn M, Pedersen TL, Miller E, Bache SM, Müller K, Ooms J, Robinson D, Seidel DP, Spinu V, Takahashi K, Vaughan D, Wilke C, Woo K, Yutani H (2019). "Welcome to the tidyverse." *Journal of Open Source Software*, *4*(43), 1686. doi:10.21105/joss.01686 https://doi.org/10.21105/joss.01686.

Wickham H, François R, Henry L, Müller K, Vaughan D (2023). *dplyr: A Grammar of Data Manipulation*. R package version 1.1.4, https://CRAN.R-project.org/package=dplyr.

Xie Y (2024). *knitr: A General-Purpose Package for Dynamic Report Generation in R*. R package version 1.47, https://yihui.org/knitr/.

Yihui Xie (2015) Dynamic Documents with R and knitr. 2nd edition. Chapman and Hall/CRC. ISBN 978-1498716963

Yihui Xie (2014) knitr: A Comprehensive Tool for Reproducible Research in R. In Victoria Stodden, Friedrich Leisch and Roger D. Peng, editors, Implementing Reproducible Computational Research. Chapman and Hall/CRC. ISBN 978-1466561595

Appendix

Libraries

```r
#libraries
library(dplyr)
library(tidyverse)
library(MASS)
library(knitr)
library(qpcR)
library(forecast)
library(astsa)
#setting echo to true
knitr::opts_chunk$set(echo=TRUE,
                      #does not speed results
                      cache=FALSE,
                      #width and height to 70%
                      out.width="70%",
                      out.height="70%",
                      #centering images
                      fig.align='center')
```

Loading in data

```r
#reading in data into R
electricity_production <- read.csv('~/Desktop/Electric_Production.csv')
```

Converting Data to Time Series

```r
#converting data into a time series
ts_electric <- ts(electricity_production$IPG2211A2N, start = c(1985, 01), frequency = 12)
```

Training and Testing Datasets

```r
#creating training set
electricity_train = ts_electric[1:348]
#creating testing set
electricity_test = ts_electric[349:398]
```

```r
#plotting training dataset
plot(1:length(electricity_train), electricity_train,
     main = "Time Series by Year (1985 to 2014)", type = 'l', xlab = 'index')
#creating index
index = 1:length(electricity_train)
#creating trend
trend <- lm(electricity_train ~ index)
#adding trend line
abline(trend, col = "red")
#adding mean line
abline(h = mean(electricity_train), col = "blue")
#variance of electricity_train
paste0("The variance is: ", var(electricity_train))
```

Histogram of Training Data

```r
#plotting histogram of the training data
hist(electricity_train, col = 'plum3', main = "Histogram of Electric Production Training Data",
     xlab = "Production")
```

ACF

```r
#acf graph of training data
acf(electricity_train, lag.max = 60, main = "ACF of Electricity Production Training Data")
```

Box-Cox

```r
#creating t with a length of 1 to the length of electricity_train
t = 1:length(electricity_train)

#using boxcox to create the transformation
bcTransform = boxcox(electricity_train ~ t, plotit = TRUE)

#creating lambda from the transformation
lambda = bcTransform$x[which(bcTransform$y == max(bcTransform$y))]

#creating the transformed data
electricity.bc = (1/lambda)*(electricity_train^lambda-1)

#printing lambda
paste0("Lambda: ", lambda)
```

```r
#plotting the Box-Cox transformed data
plot(1:length(electricity.bc), electricity.bc, main = "Time Series by Year (1985 to 2014)",
     type = 'l', xlab = 'index')

#creating index
```

```r
index = 1:length(electricity.bc)

#creating trend line
trend <- lm(electricity.bc ~ index)

#adding trend line
abline(trend, col = "red")

#adding mean line
abline(h = mean(electricity.bc), col = "blue")

#printing the variance
paste0("The variance is: ", var(electricity.bc))
```

Histogram Comparison

```r
#set graphical parameters
op <- par(mfrow = c(1,2))
#histogram of electricity_train
hist(electricity_train, col = 'orchid3', main = "Histogram of Electric Production Training Data")
#histogram of electricity.bc
hist(electricity.bc, col = 'orchid3', main = "Histogram of Box-Cox Electric Production")
```

Decomposition

```r
#converting electricity.bc into a time series
y <- ts(as.ts(electricity.bc), frequency = 12)
#decomposition of y
plot(decompose(y))
```

Differencing at Lag 1

```r
#differencing at lag 1
y1 <- diff(electricity.bc, 1)

#plotting the de-trended time series
plot(y1, main = "De-trended Time Series", type = "l")

#fitting y1 to 1 to the length of y1
fit_y1 <- lm(y1 ~ as.numeric(1:length(y1)))

#adding trend line
abline(fit_y1, col="red")

#adding mean line
abline(h=mean(y1), col="blue")
```

```r
#printing the de-trended variance
paste0("Variance after differencing at lag 1: ", var(y1))
```

ACF and PACF of De-trended Time Series

```r
#set graphical parameters
op <- par(mfrow = c(1,2))
#acf of y1
acf(y1, lag.max = 60, main = expression("ACF of " ~ Delta[1](X)))
#pacf of y1
pacf(y1, lag.max = 60, main = expression("PACF of " ~ Delta[1](X)))
```

Differencing at Lag 12

```r
#differencing at lag 12
y2 <- diff(y1, 12)

#plotting the de-seasonalized time series
plot(y2, main = "De-seasonalized Time Series", type = "l")

#printing the de-seasonalized time series
paste0("Variance after differencing at lag 12: ", var(y2))

#fitting y2 to 1 to the length of y2
fit_y2 <- lm(y2 ~ as.numeric(1:length(y2)))

#adding trend line
abline(fit_y2, col="red")

#adding mean line
abline(h=mean(y2), col="blue")
```

Differencing at Lag 4

```r
#differencing at lag 4
y3 <- diff(y1, 4)
#variance of y3
var(y3)
```

ACF and PACF of De-seasonalized Time Series

```r
#set graphical parameters
op <- par(mfrow = c(1,2))
#plotting acf of y2 to check seasonal patterns
acf(y2, lag.max = 60, main = expression("ACF of " ~ Delta[1]*Delta[12](X)))
```

```
#plotting pacf of y2 to check seasonal patterns
pacf(y2, lag.max = 60, main = expression("PACF of " ~ Delta[1]*Delta[12](X)))
```

ACF and PACF of De-seasonalized Time Series Within A Seasonality

```
#set graphical parameters
op <- par(mfrow = c(1,2))
#plotting acf of y2 to check within one seasonality
acf(y2, lag.max = 12, main = expression("ACF of " ~ Delta[1]*Delta[12](X)))

#plotting pacf of y2 to check within one seasonality
pacf(y2, lag.max = 12, main = expression("PACF of " ~ Delta[1]*Delta[12](X)))
```

Histogram of De-seasonalized Time Series

```
#histogram of de-seasonalized time series
hist(y2, col = 'mediumorchid3', main = expression("Histogram of " ~ Delta[1]*Delta[12](X)))
```

Models

```
#SARIMA(2,1,1)(1,1,1)
#trying SARIMA(2,1,1)(1,1,1)_12
#AICc = -131.1191
arima(electricity.bc, order = c(2,1,1), seasonal = list(order = c(1,1,1), period = 12),
      method = "ML")
#checking AICc for SARIMA(2,1,1)(1,1,1)_12
AICc(arima(electricity.bc, order = c(2,1,1), seasonal = list(order = c(1,1,1), period = 12),
           method = "ML"))
#trying SARIMA(2,1,1)(1,1,1)_12 with fixed coefficients
#AICc = -133.2847
arima(electricity.bc, order = c(2,1,1), seasonal = list(order = c(1,1,1), period = 12),
      fixed = c(NA, 0, NA, 0, NA), method = "ML")
#checking AICc for SARIMA(2,1,1)(1,1,1)_12 with fixed coefficients
AICc(arima(electricity.bc, order = c(2,1,1), seasonal = list(order = c(1,1,1), period = 12),
           fixed = c(NA, 0, NA, 0, NA), method = "ML"))
```

```
#SARIMA(2,1,1)(1,1,2)
#trying SARIMA(2,1,1)(1,1,2)_12
#AICc = -131.7136
arima(electricity.bc, order = c(2,1,1), seasonal = list(order = c(1,1,2), period = 12),
      method = "ML")
#checking AICc for SARIMA(2,1,1)(1,1,2)_12
AICc(arima(electricity.bc, order = c(2,1,1), seasonal = list(order = c(1,1,2), period = 12),
           method = "ML"))
#trying SARIMA(2,1,1)(1,1,2)_12 with fixed coefficients
#AICc = -133.3393
```

```r
arima(electricity.bc, order = c(2,1,1), seasonal = list(order = c(1,1,2), period = 12),
      fixed = c(NA, 0, NA, NA, 0, NA), method = "ML")
#checking AICc for SARIMA(2,1,1)(1,1,2)_12 with fixed coefficients
AICc(arima(electricity.bc, order = c(2,1,1), seasonal = list(order = c(1,1,2), period = 12),
           fixed = c(NA, 0, NA, NA, 0, NA), method = "ML"))
```

```r
#SARIMA(2,1,1)(2,1,1) Model A
#trying SARIMA(2,1,1)(2,1,1)_12
#AICc = -136.6063
arima(electricity.bc, order = c(2,1,1), seasonal = list(order = c(2,1,1), period = 12),
      method = "ML")
#checking AICc for SARIMA(2,1,1)(2,1,1)_12
AICc(arima(electricity.bc, order = c(2,1,1), seasonal = list(order = c(2,1,1), period = 12),
           method = "ML"))
#trying SARIMA(2,1,1)(2,1,1)_12 with fixed coefficients
#AICc = -139.3177
arima(electricity.bc, order = c(2,1,1), seasonal = list(order = c(2,1,1), period = 12),
      fixed = c(NA, 0, NA, 0, NA, NA))
#checking AICc for SARIMA(2,1,1)(2,1,1)_12 with fixed coefficients
AICc(arima(electricity.bc, order = c(2,1,1), seasonal = list(order = c(2,1,1), period = 12),
           fixed = c(NA, 0, NA, 0, NA, NA)))
```

```r
#SARIMA(2,1,1)(2,1,2)
#trying SARIMA(2,1,1)(2,1,2)_12
#AICc = -135.7247
arima(electricity.bc, order = c(2,1,1), seasonal = list(order = c(2,1,2), period = 12),
      method = "ML")
#checking AICc for SARIMA(2,1,1)(2,1,2)_12
AICc(arima(electricity.bc, order = c(2,1,1), seasonal = list(order = c(2,1,2), period = 12),
           method = "ML"))
#trying SARIMA(2,1,1)(2,1,2)_12 with fixed coefficients
#AICc = -137.0059
arima(electricity.bc, order = c(2,1,1), seasonal = list(order = c(2,1,2), period = 12),
      fixed = c(NA, 0, NA, NA, NA, NA, NA))
#checking AICc for SARIMA(2,1,1)(2,1,2)_12 with fixed coefficients
AICc(arima(electricity.bc, order = c(2,1,1), seasonal = list(order = c(2,1,2), period = 12),
           fixed = c(NA, 0, NA, NA, NA, NA, NA)))
#trying SARIMA(2,1,1)(2,1,2)_12 with fixed coefficients
#AICc = -83.62477
arima(electricity.bc, order = c(2,1,1), seasonal = list(order = c(2,1,2), period = 12),
      fixed = c(NA, 0, NA, 0, NA, NA, NA))
#checking AICc for SARIMA(2,1,1)(2,1,2)_12 with fixed coefficients
AICc(arima(electricity.bc, order = c(2,1,1), seasonal = list(order = c(2,1,2), period = 12),
           fixed = c(NA, 0, 0, NA, NA, NA, NA)))
```

```r
#SARIMA(2,1,2)(1,1,1)
#trying SARIMA(2,1,2)(1,1,1)_12
#AICc = -130.1487
arima(electricity.bc, order = c(2,1,2), seasonal = list(order = c(1,1,1), period = 12),
      method = "ML")
#checking AICc for SARIMA(2,1,2)(1,1,1)_12
AICc(arima(electricity.bc, order = c(2,1,2), seasonal = list(order = c(1,1,1), period = 12),
           method = "ML"))
```

```r
#trying SARIMA(2,1,2)(1,1,1)_12 with fixed coefficients
#AICc = -66.48683
arima(electricity.bc, order = c(2,1,2), seasonal = list(order = c(1,1,1), period = 12),
      fixed = c(0, 0, 0, 0, 0, NA), method = "ML")
#checking AICc for SARIMA(2,1,2)(1,1,1)_12 with fixed coefficients
AICc(arima(electricity.bc, order = c(2,1,2), seasonal = list(order = c(1,1,1), period = 12),
           fixed = c(0, 0, 0, 0, 0, NA), method = "ML"))
```

```r
#SARIMA(2,1,2)(1,1,2)
#trying SARIMA(2,1,2)(1,1,2)_12
#AICc = -130.5209
arima(electricity.bc, order = c(2,1,2), seasonal = list(order = c(1,1,2), period = 12),
      method = "ML")
#checking AICc for SARIMA(2,1,2)(1,1,2)_12
AICc(arima(electricity.bc, order = c(2,1,2), seasonal = list(order = c(1,1,2), period = 12),
           method = "ML"))
#trying SARIMA(2,1,2)(1,1,2)_12 with fixed coefficients
#AICc = 58.8537
arima(electricity.bc, order = c(2,1,2), seasonal = list(order = c(1,1,2), period = 12),
      fixed = c(0, 0, 0, 0, 0, 0, NA), method = "ML")
#checking AICc for SARIMA(2,1,2)(1,1,2)_12 with fixed coefficients
AICc(arima(electricity.bc, order = c(2,1,2), seasonal = list(order = c(1,1,2), period = 12),
           fixed = c(0, 0, 0, 0, 0, 0, NA), method = "ML"))
```

```r
#SARIMA(2,1,2)(2,1,1)
#trying SARIMA(2,1,2)(2,1,1)_12
#AICc = -135.0825
arima(electricity.bc, order = c(2,1,2), seasonal = list(order = c(2,1,1), period = 12),
      method = "ML")
#checking AICc for SARIMA(2,1,2)(2,1,1)_12
AICc(arima(electricity.bc, order = c(2,1,2), seasonal = list(order = c(2,1,1), period = 12),
           method = "ML"))
#trying SARIMA(2,1,2)(2,1,1)_12 with fixed coefficients
#AICc = -73.48194
arima(electricity.bc, order = c(2,1,2), seasonal = list(order = c(2,1,1), period = 12),
      fixed = c(0, 0, 0, 0, 0, NA, NA), method = "ML")
#checking AICc for SARIMA(2,1,2)(2,1,1)_12 with fixed coefficients
AICc(arima(electricity.bc, order = c(2,1,2), seasonal = list(order = c(2,1,1), period = 12),
           fixed = c(0, 0, 0, 0, 0, NA, NA), method = "ML"))
```

```r
#SARIMA(2,1,2)(2,1,2)
#trying SARIMA(2,1,2)(2,1,2)_12
#AICc = -134.106
arima(electricity.bc, order = c(2,1,2), seasonal = list(order = c(2,1,2), period = 12))
#checking AICc for SARIMA(2,1,2)(2,1,2)_12
AICc(arima(electricity.bc, order = c(2,1,2), seasonal = list(order = c(2,1,2), period = 12)))
#trying SARIMA(2,1,2)(2,1,2)_12 with fixed coefficients
#AICc = -73.38657
arima(electricity.bc, order = c(2,1,2), seasonal = list(order = c(2,1,2), period = 12),
      fixed = c(0, 0, 0, 0, 0, NA, NA, 0), method = "ML")
#checking AICc for SARIMA(2,1,2)(2,1,2)_12 with fixed coefficients
AICc(arima(electricity.bc, order = c(2,1,2), seasonal = list(order = c(2,1,2), period = 12),
           fixed = c(0, 0, 0, 0, 0, NA, NA, 0), method = "ML"))
```

```
#SARIMA(4,1,1)(1,1,1)
#trying SARIMA(4,1,1)(1,1,1)_12
#AICc = -129.3006
arima(electricity.bc, order = c(4,1,1), seasonal = list(order = c(1,1,1), period = 12),
      method = "ML")
#checking AICc for SARIMA(4,1,1)(1,1,1)_12
AICc(arima(electricity.bc, order = c(4,1,1), seasonal = list(order = c(1,1,1), period = 12),
           method = "ML"))
#trying SARIMA(4,1,1)(1,1,1)_12 with fixed coefficients
#AICc = -133.1308
arima(electricity.bc, order = c(4,1,1), seasonal = list(order = c(1,1,1), period = 12), fixed = c(NA, 0
      method = "ML")
#checking AICc for SARIMA(4,1,1)(1,1,1)_12 with fixed coefficients
AICc(arima(electricity.bc, order = c(4,1,1), seasonal = list(order = c(1,1,1), period = 12), fixed = c(
           method = "ML"))
```

```
#SARIMA(4,1,1)(1,1,2)
#trying SARIMA(4,1,1)(1,1,2)_12
#AICc = -129.4245
arima(electricity.bc, order = c(4,1,1), seasonal = list(order = c(1,1,2), period = 12),
      method = "ML")
#checking AICc for SARIMA(4,1,1)(1,1,2)_12
AICc(arima(electricity.bc, order = c(4,1,1), seasonal = list(order = c(1,1,2), period = 12),
           method = "ML"))
#trying SARIMA(4,1,1)(1,1,2)_12 with fixed coefficients
#AICc = -12.62668
arima(electricity.bc, order = c(4,1,1), seasonal = list(order = c(1,1,2), period = 12),
      fixed = c(NA, 0, 0, 0, NA, 0, 0, NA), method = "ML")
#checking AICc for SARIMA(4,1,1)(1,1,2)_12 with fixed coefficients
AICc(arima(electricity.bc, order = c(4,1,1), seasonal = list(order = c(1,1,2), period = 12),
           fixed = c(NA, 0, 0, 0, NA, 0, 0, NA), method = "ML"))
```

```
#SARIMA(4,1,1)(2,1,1) Model B
#trying model SARIMA(4,1,1)(2,1,1)_12
#AICc = -133.7444
arima(electricity.bc, order = c(4,1,1), seasonal = list(order = c(2,1,1), period = 12),
      method = "ML")
#checking AICc for model SARIMA(4,1,1)(2,1,1)_12
AICc(arima(electricity.bc, order = c(4,1,1), seasonal = list(order = c(2,1,1), period = 12),
           method = "ML"))
#trying model SARIMA(4,1,1)(2,1,1)_12 with fixed coefficients
#AICc = -139.1392
arima(electricity.bc, order = c(4,1,1), seasonal = list(order = c(2,1,1), period = 12),
      fixed = c(NA, 0, 0, 0, NA, 0, NA, NA))
#checking AICc for model SARIMA(4,1,1)(2,1,1)_12 with fixed coefficients
AICc(arima(electricity.bc, order = c(4,1,1), seasonal = list(order = c(2,1,1), period = 12),
           fixed = c(NA, 0, 0, 0, NA, 0, NA, NA)))
```

```
#SARIMA(4,1,1)(2,1,2)
#trying model SARIMA(4,1,1)(2,1,2)_12
#AICc = -132.6952
arima(electricity.bc, order = c(4,1,1), seasonal = list(order = c(2,1,2), period = 12),
      method = "ML")
```

```r
#checking AICc for model SARIMA(4,1,1)(2,1,2)_12
AICc(arima(electricity.bc, order = c(4,1,1), seasonal = list(order = c(2,1,2), period = 12),
           method = "ML"))
#trying model SARIMA(4,1,1)(2,1,2)_12 with fixed coefficients
#AICc = -139.0315
arima(electricity.bc, order = c(4,1,1), seasonal = list(order = c(2,1,2), period = 12),
      fixed = c(NA, 0, 0, 0, NA, 0, NA, NA, 0))
#checking AICc for model SARIMA(4,1,1)(2,1,2)_12 with fixed coefficients
AICc(arima(electricity.bc, order = c(4,1,1), seasonal = list(order = c(2,1,2), period = 12),
           fixed = c(NA, 0, 0, 0, NA, 0, NA, NA, 0)))
```

```r
#SARIMA(4,1,2)(1,1,1)
#trying model SARIMA(4,1,2)(1,1,1)_12
#AICc = -127.2347
arima(electricity.bc, order = c(4,1,2), seasonal = list(order = c(1,1,1), period = 12),
      method = "ML")
#checking AICc for model SARIMA(4,1,2)(1,1,1)_12
AICc(arima(electricity.bc, order = c(4,1,2), seasonal = list(order = c(1,1,1), period = 12),
           method = "ML"))
#trying model SARIMA(4,1,2)(1,1,1)_12 with fixed coefficients
#AICc = -92.09458
arima(electricity.bc, order = c(4,1,2), seasonal = list(order = c(1,1,1), period = 12),
      fixed = c(0, 0, 0, 0, NA, 0, 0, NA), method = "ML")
#checking AICc for model SARIMA(4,1,2)(1,1,1)_12 with fixed coefficients
AICc(arima(electricity.bc, order = c(4,1,2), seasonal = list(order = c(1,1,1), period = 12),
           fixed = c(0, 0, 0, 0, NA, 0, 0, NA), method = "ML"))
```

```r
#SARIMA(4,1,2)(1,1,2)
#trying model SARIMA(4,1,2)(1,1,2)_12
#AICc = -127.3381
arima(electricity.bc, order = c(4,1,2), seasonal = list(order = c(1,1,2), period = 12),
      method = "ML")
#checking AICc for model SARIMA(4,1,2)(1,1,2)_12
AICc(arima(electricity.bc, order = c(4,1,2), seasonal = list(order = c(1,1,2), period = 12),
           method = "ML"))
#trying model SARIMA(4,1,2)(1,1,2)_12 with fixed coefficients
#AICc = 59.05684
arima(electricity.bc, order = c(4,1,2), seasonal = list(order = c(1,1,2), period = 12),
      fixed = c(0, 0, 0, 0, 0, 0, 0, 0, NA), method = "ML")
#checking AICc for model SARIMA(4,1,2)(1,1,2)_12 with fixed coefficients
AICc(arima(electricity.bc, order = c(4,1,2), seasonal = list(order = c(1,1,2), period = 12),
           fixed = c(0, 0, 0, 0, 0, 0, 0, 0, NA), method = "ML"))
```

```r
#SARIMA(4,1,2)(2,1,1)
#trying model SARIMA(4,1,2)(2,1,1)_12
#AICc = -131.6398
arima(electricity.bc, order = c(4,1,2), seasonal = list(order = c(2,1,1), period = 12),
      method = "ML")
#checking AICc for model SARIMA(4,1,2)(2,1,1)_12
AICc(arima(electricity.bc, order = c(4,1,2), seasonal = list(order = c(2,1,1), period = 12),
           method = "ML"))
#trying model SARIMA(4,1,2)(2,1,1)_12 with fixed coefficients
#AICc = -73.2788
```

```r
arima(electricity.bc, order = c(4,1,2), seasonal = list(order = c(2,1,1), period = 12),
      fixed = c(0, 0, 0, 0, 0, 0, 0, NA, NA), method = "ML")
#checking AICc for model SARIMA(4,1,2)(2,1,1)_12 with fixed coefficients
AICc(arima(electricity.bc, order = c(4,1,2), seasonal = list(order = c(2,1,1), period = 12),
           fixed = c(0, 0, 0, 0, 0, 0, 0, NA, NA), method = "ML"))
```

```r
#SARIMA(4,1,2)(2,1,2)
#trying model SARIMA(4,1,2)(2,1,2)_12
#AICc = -130.5750
arima(electricity.bc, order = c(4,1,2), seasonal = list(order = c(2,1,2), period = 12),
      method = "ML")
#checking AICc for model SARIMA(4,1,2)(2,1,2)_12
AICc(arima(electricity.bc, order = c(4,1,2), seasonal = list(order = c(2,1,2), period = 12),
           method = "ML"))
#trying model SARIMA(4,1,2)(2,1,2)_12 with fixed coefficients
#AICc = -73.15853
arima(electricity.bc, order = c(4,1,2), seasonal = list(order = c(2,1,2), period = 12),
      fixed = c(0, 0, 0, 0, 0, 0, 0, NA, NA, 0), method = "ML")
#checking AICc for model SARIMA(4,1,2)(2,1,2)_12 with fixed coefficients
AICc(arima(electricity.bc, order = c(4,1,2), seasonal = list(order = c(2,1,2), period = 12),
           fixed = c(0, 0, 0, 0, 0, 0, 0, NA, NA, 0), method = "ML"))
```

```r
#SARIMA(5,1,1)(1,1,1)
#trying model SARIMA(5,1,1)(1,1,1)_12
#AICc = -127.3053
arima(electricity.bc, order = c(5,1,1), seasonal = list(order = c(1,1,1), period = 12),
      method = "ML")
#checking AICc for model SARIMA(5,1,1)(1,1,1)_12
AICc(arima(electricity.bc, order = c(5,1,1), seasonal = list(order = c(1,1,1), period = 12),
           method = "ML"))
#trying model SARIMA(5,1,1)(1,1,1)_12 with fixed coefficients
#AICc = -133.0354
arima(electricity.bc, order = c(5,1,1), seasonal = list(order = c(1,1,1), period = 12),
      fixed = c(NA, 0, 0, 0, 0, NA, 0, NA), method = "ML")
#checking AICc for model SARIMA(5,1,1)(1,1,1)_12 with fixed coefficients
AICc(arima(electricity.bc, order = c(5,1,1), seasonal = list(order = c(1,1,1), period = 12),
           fixed = c(NA, 0, 0, 0, 0, NA, 0, NA), method = "ML"))
```

```r
#SARIMA(5,1,1)(1,1,2)
#trying model SARIMA(5,1,1)(1,1,2)_12
#AICc = -127.3955
arima(electricity.bc, order = c(5,1,1), seasonal = list(order = c(1,1,2), period = 12),
      method = "ML")
#checking AICc for model SARIMA(5,1,1)(1,1,2)_12
AICc(arima(electricity.bc, order = c(5,1,1), seasonal = list(order = c(1,1,2), period = 12),
           method = "ML"))
#trying model SARIMA(5,1,1)(1,1,2)_12 with fixed coefficients
#AICc = -12.51892
arima(electricity.bc, order = c(5,1,1), seasonal = list(order = c(1,1,2), period = 12),
      fixed = c(NA, 0, 0, 0, 0, NA, 0, 0, NA), method = "ML")
#checking AICc for model SARIMA(5,1,1)(1,1,2)_12 with fixed coefficients
AICc(arima(electricity.bc, order = c(5,1,1), seasonal = list(order = c(1,1,2), period = 12),
           fixed = c(NA, 0, 0, 0, 0, NA, 0, 0, NA), method = "ML"))
```

```r
#SARIMA(5,1,1)(2,1,1) Model C
#trying model SARIMA(5,1,1)(2,1,1)_12
#AICc = -131.6495
arima(electricity.bc, order = c(5,1,1), seasonal = list(order = c(2,1,1), period = 12),
      method = "ML")
#checking AICc for SARIMA(5,1,1)(2,1,1)_12
AICc(arima(electricity.bc, order = c(5,1,1), seasonal = list(order = c(2,1,1), period = 12),
           method = "ML"))
#trying model SARIMA(5,1,1)(2,1,1)_12 with fixed coefficients
#AICc = -139.0316
arima(electricity.bc, order = c(5,1,1), seasonal = list(order = c(2,1,1), period = 12),
      fixed = c(NA, 0, 0, 0, 0, NA, 0, NA, NA), method = "ML")
#checking AICc for SARIMA(5,1,1)(2,1,1)_12 with fixed coefficients
AICc(arima(electricity.bc, order = c(5,1,1), seasonal = list(order = c(2,1,1), period = 12),
           fixed = c(NA, 0, 0, 0, 0, NA, 0, NA, NA), method = "ML"))
```

```r
#SARIMA(5,1,1)(2,1,2)
# Error non-finite finite-difference value [7]

#trying model SARIMA(5,1,1)(2,1,2)_12
# arima(electricity.bc, order = c(5,1,1), seasonal = list(order = c(2,1,2), period = 12))
# #checking AICc for SARIMA(5,1,1)(2,1,2)_12
# AICc(arima(electricity.bc, order = c(5,1,1), seasonal = list(order = c(2,1,2), period = 12)))
# #trying model SARIMA(5,1,1)(2,1,2)_12 with fixed coefficients
# arima(electricity.bc, order = c(5,1,1), seasonal = list(order = c(2,1,2), period = 12),
#       fixed = c(NA, 0, 0, 0, 0, NA, 0, 0, NA), method = "ML")
# #checking AICc for SARIMA(5,1,1)(2,1,2)_12 with fixed coefficients
# AICc(arima(electricity.bc, order = c(5,1,1), seasonal = list(order = c(2,1,2), period = 12),
#            fixed = c(NA, 0, 0, 0, 0, NA, 0, 0, NA), method = "ML"))
```

```r
#SARIMA(5,1,2)(1,1,1)
#trying model SARIMA(5,1,2)(1,1,1)_12
#AICc = -127.7628
arima(electricity.bc, order = c(5,1,2), seasonal = list(order = c(1,1,1), period = 12),
      method = "ML")
#checking AICc for SARIMA(5,1,2)(1,1,1)_12
AICc(arima(electricity.bc, order = c(5,1,2), seasonal = list(order = c(1,1,1), period = 12),
           method = "ML"))
#trying model SARIMA(5,1,2)(1,1,1)_12 with fixed coefficients
#AICc = -131.0093
arima(electricity.bc, order = c(5,1,2), seasonal = list(order = c(1,1,1), period = 12),
      fixed = c(NA, NA, 0, 0, 0, 0, NA, 0, NA), method = "ML")
#checking AICc for SARIMA(5,1,2)(1,1,1)_12 with fixed coefficients
AICc(arima(electricity.bc, order = c(5,1,2), seasonal = list(order = c(1,1,1), period = 12),
           fixed = c(NA, NA, 0, 0, 0, 0, NA, 0, NA), method = "ML"))
```

```r
#SARIMA(5,1,2)(1,1,2)
#trying model SARIMA(5,1,2)(1,1,2)_12
#AICc = -127.67324
arima(electricity.bc, order = c(5,1,2), seasonal = list(order = c(1,1,2), period = 12),
      method = "ML")
#checking AICc for SARIMA(5,1,2)(1,1,2)_12
AICc(arima(electricity.bc, order = c(5,1,2), seasonal = list(order = c(1,1,2), period = 12),
```

```
           method = "ML"))
#trying model SARIMA(5,1,2)(1,1,2)_12 with fixed coefficients
#AICc = -17.67324
arima(electricity.bc, order = c(5,1,2), seasonal = list(order = c(1,1,2), period = 12),
      fixed = c(NA, NA, 0, 0, 0, 0, NA, 0, 0, NA), method = "ML")
#checking AICc for SARIMA(5,1,2)(1,1,2)_12 with fixed coefficients
AICc(arima(electricity.bc, order = c(5,1,2), seasonal = list(order = c(1,1,2), period = 12),
           fixed = c(NA, NA, 0, 0, 0, 0, NA, 0, 0, NA), method = "ML"))
```

```
#SARIMA(5,1,2)(2,1,1)
#trying model SARIMA(5,1,2)(2,1,1)_12
#AICc = -133.1973
arima(electricity.bc, order = c(5,1,2), seasonal = list(order = c(2,1,1), period = 12),
      method = "ML")
#checking AICc for SARIMA(5,1,2)(2,1,1)_12
AICc(arima(electricity.bc, order = c(5,1,2), seasonal = list(order = c(2,1,1), period = 12),
           method = "ML"))
#trying model SARIMA(5,1,2)(2,1,1)_12 with fixed coefficients
#AICc = -137.2604
arima(electricity.bc, order = c(5,1,2), seasonal = list(order = c(2,1,1), period = 12),
      fixed = c(NA, NA, 0, 0, 0, 0, NA, 0, NA, NA), method = "ML")
#checking AICc for SARIMA(5,1,2)(2,1,1)_12 with fixed coefficients
AICc(arima(electricity.bc, order = c(5,1,2), seasonal = list(order = c(2,1,1), period = 12),
           fixed = c(NA, NA, 0, 0, 0, 0, NA, 0, NA, NA), method = "ML"))
```

```
#SARIMA(5,1,2)(2,1,2)
#trying model SARIMA(5,1,2)(2,1,2)_12
#AICc = -133.3340
arima(electricity.bc, order = c(5,1,2), seasonal = list(order = c(2,1,2), period = 12),
      method = "ML")
#checking AICc for SARIMA(5,1,2)(2,1,2)_12
AICc(arima(electricity.bc, order = c(5,1,2), seasonal = list(order = c(2,1,2), period = 12),
           method = "ML"))
#trying model SARIMA(5,1,2)(2,1,2)_12 with fixed coefficients
#AICc = -135.6305
arima(electricity.bc, order = c(5,1,2), seasonal = list(order = c(2,1,2), period = 12),
      fixed = c(NA, NA, 0, 0, 0, 0, NA, 0, NA, NA, NA), method = "ML")
#checking AICc for SARIMA(5,1,2)(2,1,2)_12 with fixed coefficients
AICc(arima(electricity.bc, order = c(5,1,2), seasonal = list(order = c(2,1,2), period = 12),
           fixed = c(NA, NA, 0, 0, 0, 0, NA, 0, NA, NA, NA), method = "ML"))
```

```
#function for plotting roots
plot.roots <- function(ar.roots=NULL, ma.roots=NULL, size=2, angles=FALSE, special=NULL, sqecial=NULL,my
{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)
```

```r
}
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)
}
}
```

Invertible & Stationarity Check

```r
#set graphical parameters
op <- par(mfrow = c(1,2))
#plotting roots for SAR part for Model A
plot.roots(NULL, polyroot(c(1, 0, 0.1786)), main = "(A) roots of SAR part")
#plotting roots for SMA part for Model A
plot.roots(NULL, polyroot(c(1, 0, -0.7181)), main = "(A) roots of SMA part")
```

```r
#set graphical parameters
op <- par(mfrow = c(1,2))
#plotting roots for SAR part for Model B
plot.roots(NULL, polyroot(c(1, 0, 0.1787)), main = "(B) roots of SAR part")
#plotting roots for SMA part for Model B
plot.roots(NULL, polyroot(c(1, 0, -0.7181)), main = "(B) roots of SMA part")
```

Fitting Models

```r
#fitting SARIMA(2,1,1)(2,1,1)_12 and fixing coefficients
fit_A <- arima(electricity.bc, order = c(2,1,1), seasonal = list(order = c(2,1,1), period = 12),
               fixed = c(NA, 0, NA, 0, NA, NA)) #4
```

```r
#fitting SARIMA(4,1,1)(2,1,1)_12 and fixing coefficients
fit_B <- arima(electricity.bc, , order = c(4,1,1), seasonal = list(order = c(2,1,1), period = 12),
               fixed = c(NA, 0, 0, 0, NA, 0, NA, NA)) #4
```

Diagnostic Check

Model A

```r
#residuals of Model A
res_A <- residuals(fit_A)
#set graphical parameters
par(mfrow = c(2,2))
#histogram of the residuals
hist(res_A, density = 20, breaks = 20, col = "blue", prob = TRUE,
     main = "Histogram of Residuals of Model A")

#mean of residuals of Model A
m_A <- mean(res_A)
#standard deviation of residuals of Model A
std_A <- sqrt(var(res_A))
#adding curve to the histogram plot
curve(dnorm(x, m_A, std_A), add = TRUE)

#plotting the time series of residuals of Model A
plot.ts(res_A, ylab = "Residuals of Model A",
        main = "Residuals Plot of Model A")
#fitting linear regression
fitt <- lm(res_A ~ as.numeric(1:length(res_A)))

#adding linear regression line
abline(fitt, col = 'red')
#adding mean line
abline(h = mean(res_A), col = "blue")
#plotting the qqnorm of the residuals of Model A
qqnorm(res_A, main = "Normal Q-Q Plot for Model A")
#adding linear line of the residuals of Model A
qqline(res_A, col = "blue")
```

```r
#set graphical parameters
par(mfrow=c(1,2))
#plotting the ACF of residuals of Model A
acf(res_A, lag.max = 60, main = "ACF of Residuals of Model A")
#plotting the PACF of residuals of Model A
pacf(res_A, lag.max = 60, main = "PACF of Residuals of Model A")
```

```r
#shapiro test on residuals of Model A
shapiro.test(res_A)
```

```r
#Box-Pierce test on residuals of Model A
Box.test(res_A, lag = 19, type = c("Box-Pierce"), fitdf = 4)
```

```r
#Ljung-Box test on residuals of Model A
Box.test(res_A, lag = 19, type = c("Ljung-Box"), fitdf = 4)
```

```r
#Ljung-Box test on residuals of Model A
Box.test((res_A)^2, lag = 19, type = c("Ljung-Box"), fitdf = 0)
```

```r
#Yule-Walker test on residuals of Model A
ar(res_A, aic = TRUE, order.max = NULL, method = c("yule-walker"))
```

Model B

```r
#residuals of Model B
res_B <- residuals(fit_B)
#set graphical parameters
par(mfrow = c(2,2))
#histogram of the residuals
hist(res_B, density = 20, breaks = 20, col = "blue", prob = TRUE,
     main = "Histogram of Residuals of Model B")

#mean of residuals of Model B
m_B <- mean(res_B)
#standard deviation of residuals of Model B
std_B <- sqrt(var(res_B))
#adding curve to the histogram plot
curve(dnorm(x, m_B, std_B), add = TRUE)

#plotting the time series of residuals of Model B
plot.ts(res_B, ylab = "Residuals of Model B",
        main = "Residuals Plot of Model B")
#fitting linear regression
fitB <- lm(res_B ~ as.numeric(1:length(res_B)))

#adding linear regression line
abline(fitB, col = 'red')
#adding mean line
abline(h = mean(res_B), col = "blue")
#plotting the qqnorm of the residuals of Model B
qqnorm(res_B, main = "Normal Q-Q Plot for Model B")
#adding linear line of the residuals of Model B
qqline(res_B, col = "blue")
```

```r
#set graphical parameters
par(mfrow=c(1,2))
#plotting the ACF of the residuals of Model B
acf(res_B, lag.max = 60, main = "ACF of Residuals of Model B")
#plotting the PACF of the residuals of Model B
pacf(res_B, lag.max = 60, main = "PACF of Residuals of Model B")
```

```r
#shapiro test on residuals of Model B
shapiro.test(res_B)
```

```r
#Box-Pierce test on residuals of Model B
Box.test(res_B, lag = 19, type = c("Box-Pierce"), fitdf = 4)
```

```r
#Ljung-Box test on residuals of Model B
Box.test(res_B, lag = 19, type = c("Ljung-Box"), fitdf = 4)
```

```r
#Ljung-Box test on residuals of Model B
Box.test((res_B)^2, lag = 19, type = c("Ljung-Box"), fitdf = 0)
```

```r
#Yule-Walker test on residuals of Model B
ar(res_B, aic = TRUE, order.max = NULL, method = c("yule-walker"))
```

Forecasting

```r
#predicting 12 observations based on Model A
pred_A <- predict(fit_A, n.ahead = 12)

#upper bound
U_A = pred_A$pred + 2 * pred_A$se
#lower bound
L_A = pred_A$pred - 2 * pred_A$se

#plotting electricity.bc
ts.plot(electricity.bc, xlim = c(1, length(electricity.bc) + 12), ylim = c(min(electricity.bc),
                                                                          max(U_A, electricity.bc)))
#adding dashed lines for upper bound
lines(U_A, col = "blue", lty = "dashed")
#adding dashed lines for lower bound
lines(L_A, col = "blue", lty = "dashed")
#plotting predicted points
points((length(electricity.bc) + 1):(length(electricity.bc) + 12), pred_A$pred, col="red")
```

```r
#reverting to the original dataset
pred.orig <- (0.4646 * pred_A$pred + 1)^(1/0.4646)
#upper bound for original data
U = (U_A * 0.4646 + 1)^(1/0.4646)
#lower bound for original data
L = (L_A * 0.4646 + 1)^(1/0.4646)
#set graphical parameters
par(mfrow=c(2,1))

#plotting the original data
ts.plot(as.numeric(ts_electric),
        xlim = c(200, 360),
ylab = "Electricity Production", main = "Visualization of Forecasting on Testing Set")

#adding dashed lines for upper bound
lines(U, col="blue", lty="dashed")

#adding dashed lines for lower bound
```

```r
lines(L, col="blue", lty="dashed")
#adding in points
points((length(electricity.bc) + 1):(length(electricity.bc) + 12), pred.orig, col = "black")
#plotting original data from 340 to 360
ts.plot(as.numeric(ts_electric), xlim = c(340, 360), col = "red", ylab = "Electricity Production",
main = "Zoomed in Visualization of Forecasting on Testing Set")

#adding dashed lines for upper bound
lines(U, col="blue", lty="dashed", lwd =2)
#adding dashed lines for lower bound
lines(L, col="blue", lty="dashed", lwd =2)
#plotting predicted points
points((length(electricity.bc) + 1):(length(electricity.bc) + 12), pred.orig, col="black", pch = 20)
```