Time Series Analysis:
Monthly Electric Power Generation in the United States

Author:Qifei Cui

University of California, Santa Barbara

PSTAT 274: TIME SERIES

Instructor: Dr. Raya Feldman

Fall 2023

Abstract

In this paper, I focused on analyzing the monthly electric power generation in the United States from January 1985 to January 2018, using the SARIMA model. I addressed challenges like non-stationarity and heteroscedasticity in the dataset and successfully built a model for forecasting future electricity production. The study confirmed the effectiveness of the SARIMA model in capturing and predicting long-term trends in electricity production, providing valuable insights for policymakers and energy analysts. My work demonstrates a practical application of advanced time series analysis techniques in real-world data.

Introduction

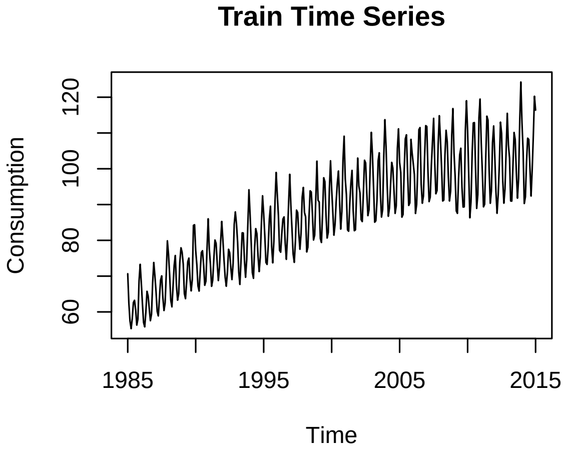
This study focuses on an analysis of a time series dataset from the Federal Reserve Economic Data (FRED), which tracks the monthly electric power generation in the United States (NAICS = 2211). This dataset spans from January 1985 to January 2018 with 397 observations, providing a vast and detailed view of electric production records over multiple decades. By examining this dataset, one could gain valuable insights into long-term patterns and fluctuations in the U.S. electric power sector, a critical component of the nation's industrial infrastructure.

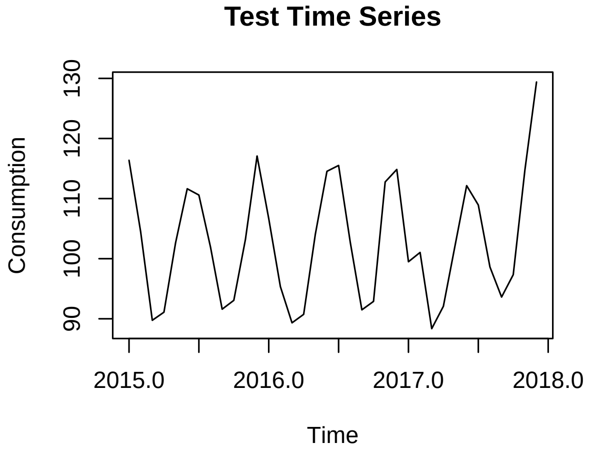
The primary objective of this study is to apply time series analysis techniques to model the monthly electric power generation and to forecast future electricity production. This forecasting is crucial for several stakeholders, including policymakers, energy producers, and environmental analysts, as it aids in planning, policy formulation, and understanding the implications of past and future trends in energy production.

To achieve this, I employ Seasonal Auto Regressive Integrated Moving Average(SARIMA) models, which is chosen for its effectiveness in modeling and predicting time series data, particularly in capturing long-term trends. After the sequence of data preparation, data transformation and differencing, model identificiation, coefficient estimation, and diagnositic checking, a model was successfully built to fit the data and was used to forecast future monthly electric power generation.

Data Preprocessing

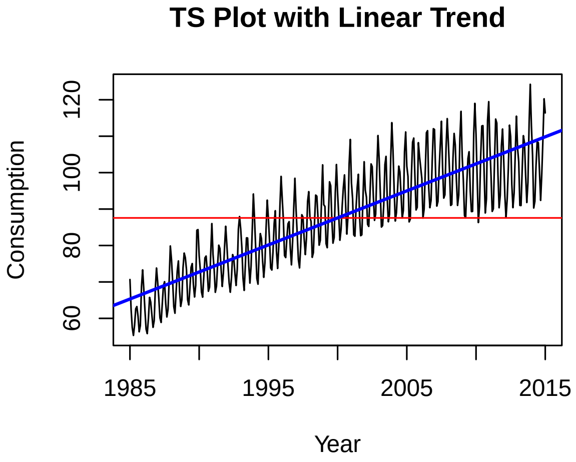
For forecasting, the dataset was partitioned into training and testing sets using a window function, with the delineation point set at January 2015. The training set encompasses 348 observations preceding 2015, while the testing set contains 36 observations from January 2015 onward.

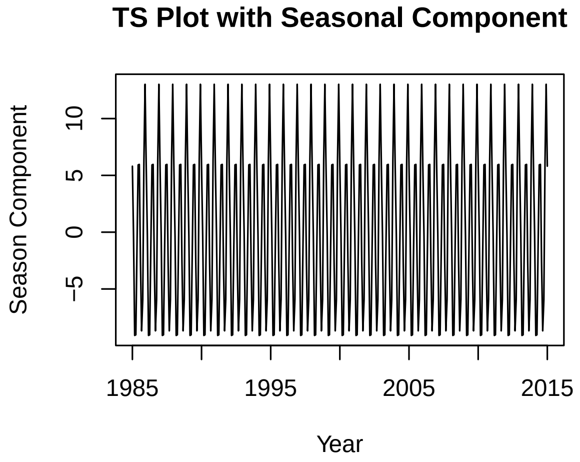

Train Time Series


Test Time Series

Plot and Analyze

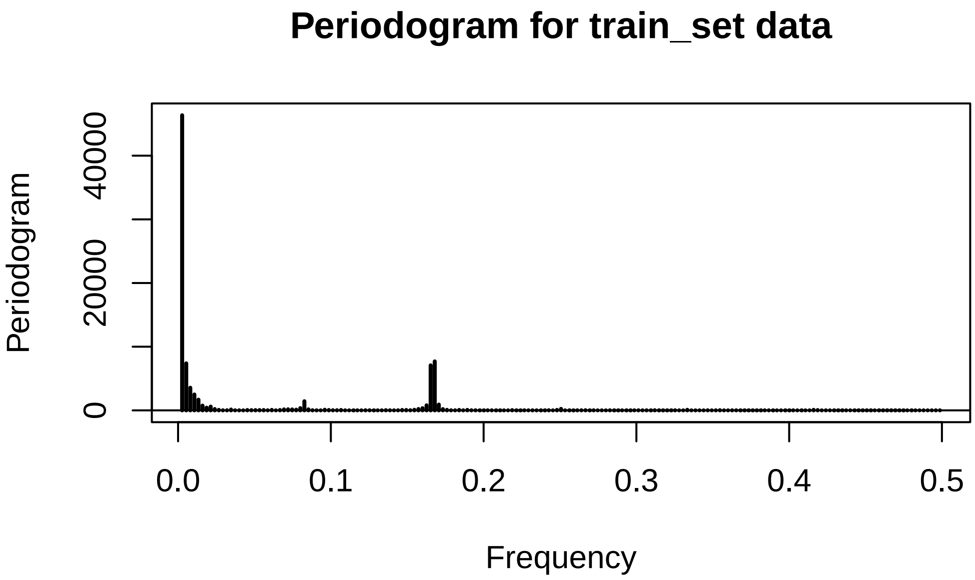
The SARIMA model fitting presupposes the time series' stationarity, where its mean, variance, and autocorrelation are uniform over time. However, the `monthly electric production` dataset exhibits trends, seasonality, and heteroscedasticity.


TS Plot with Linear Trend


TS Plot with Seasonal Component

In the left plot, the `train_set`'s linear regression line highlights a long-term trend, violating stationarity by changing the mean. The right plot displays seasonality within the `train_set`, with regular, predictable patterns altering the mean and variance at fixed intervals.

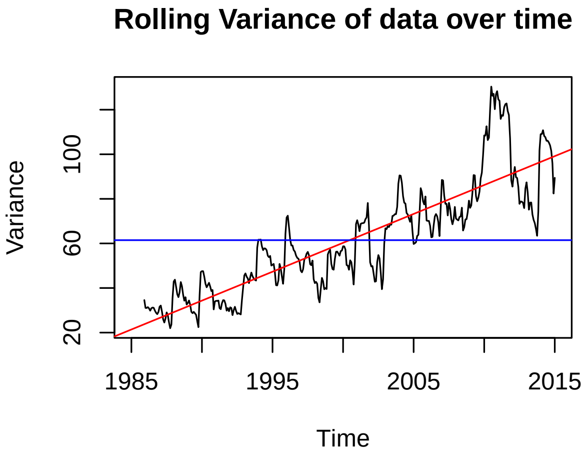
The periodogram for the `train_set` data reveals distinct spikes, which are indicative of dominant frequencies within the time series data. The most prominent spike occurs at $0.00267 \approx 1/374.5$, which suggests a long-term trend in the data. Another significant spike is observed at the $0.167 \approx 1/6$, indicating a period of 6. Need to mention that there's a less significant spike at $0.08267 \approx 1/12$, which indicates a period of 12.

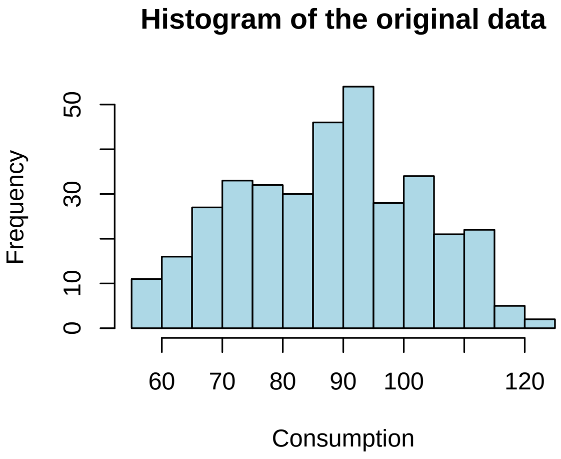


Transformation and Differencing

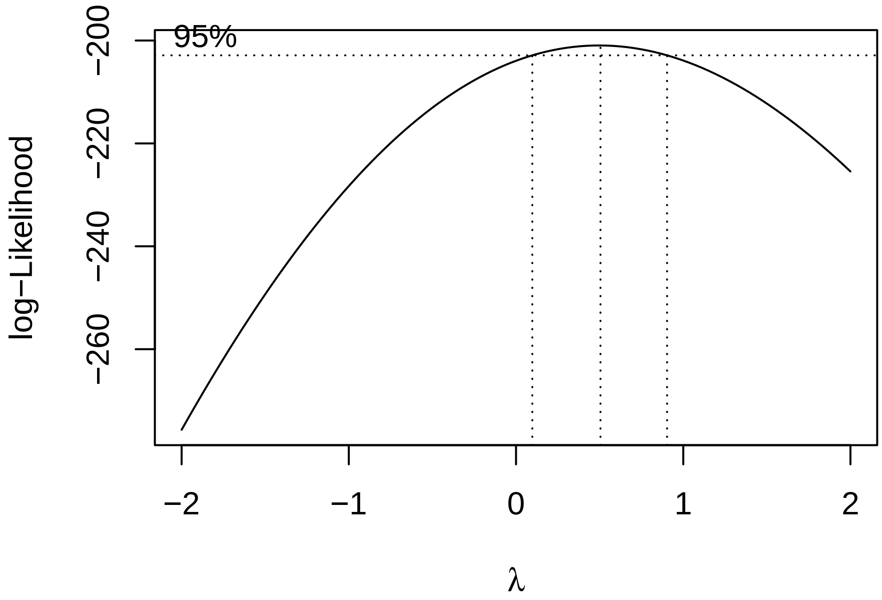
Transformation

To eliminate trends and achieve mean stabilization, differencing and transformations are employed. When I considered applying a transformation, I first check if the variance is changing with the time. Choosing a window of 12 (one period) lags, I plot the rolling Variance vs. Time plot. From the left plot, I find fluctuations and an increasing trend of rolling variance over the time suggested that the variance is not stationary (heteroskedasticity). Meanwhile, the right plot demonstrates that the original dataset exhibits a symmetrical distribution with no significant skewness. Therefore, a transformation is needed to stabilize variance and seasonal effect.

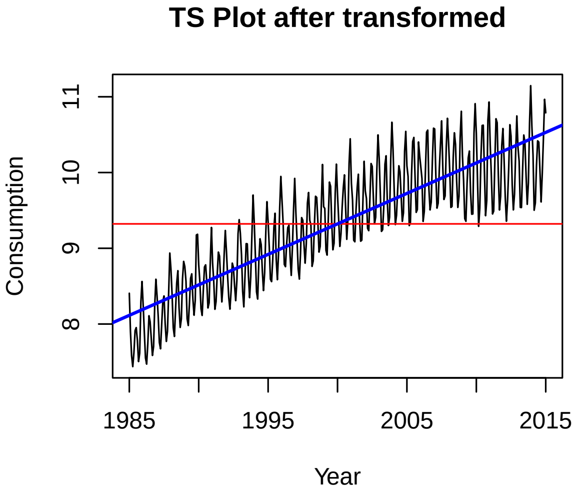


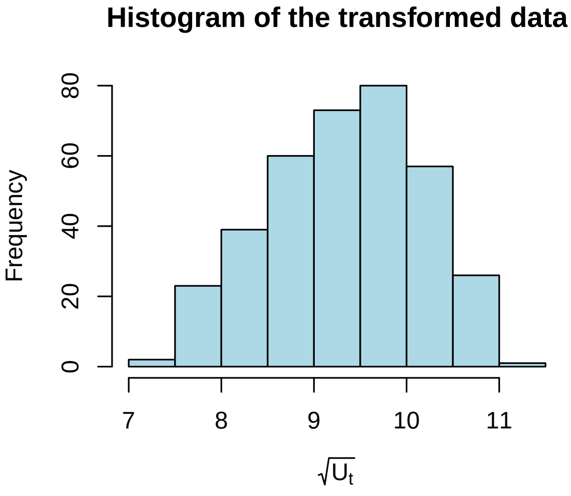


Let U_t donate the original data. The Box-Cox command gives us the maximum likelihood value of $\lambda_0 = 0.5051$. We may also notice that $\lambda = 0.5$ is also in the confidence interval and very close to λ_0. The square root transformation is selected.



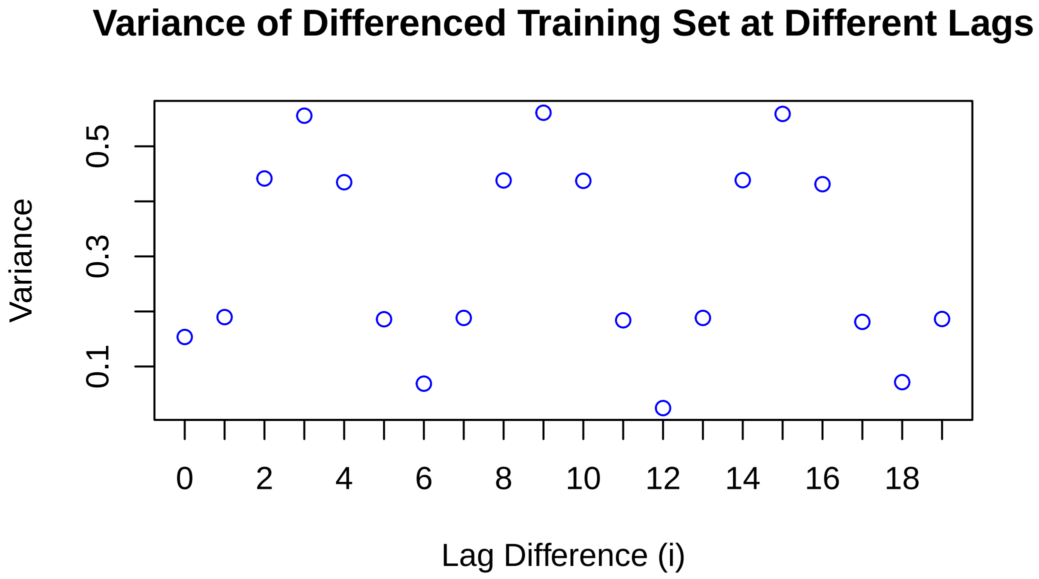
After transformation, there is a noticeable stabilization and reduction in variance across the time. Additionally, the data's distribution now approximates normal distributed which characterized by its bell-shaped appearance. Despite initial transformations, residual trends and seasonal patterns indicate that the data need additional differencing.



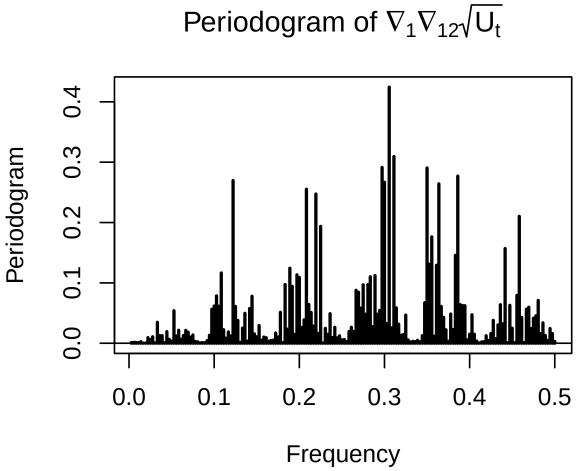


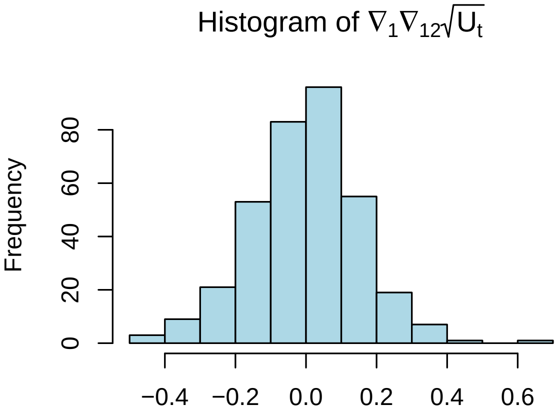
Differencing

In previous part, I identified a trend within the dataset, which was further corroborated by the periodogram's indication of a long-term trend. Hence, a first-order differencing *(lag 1)* was applied to remove this trend. The periodogram also highlighted two short-term dominant cycles with periods of 6 and 12. In order to evaluate the need for additional differencing measures and to capture the effects of these dominant periods, I have devised plots with the lag difference *(k)* on the x-axis, ranging from 1 to 20. The corresponding y-axis measures the variance of the dataset after each differencing operation.



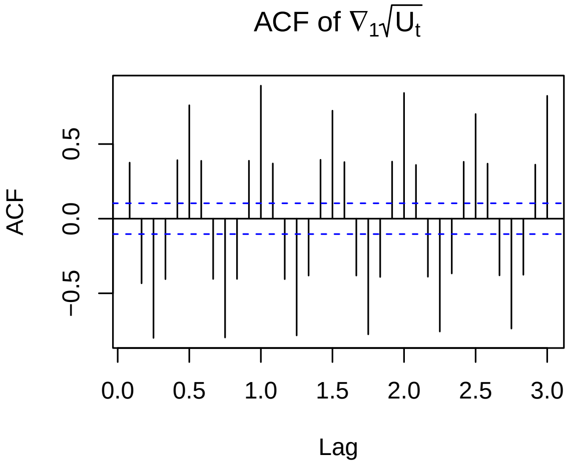
The plot analysis reveals that applying a differencing at lag 12 yields the minimum variance in the time series, suggesting that differencing at other lags could lead to over-differencing. A seasonal trend of 12 is reasonable since there are 12 months in a year, and a lag of 12 captures the seasonality that recurs yearly. Hence, I decided to difference the data once at lag one and difference twice at lag twelve.

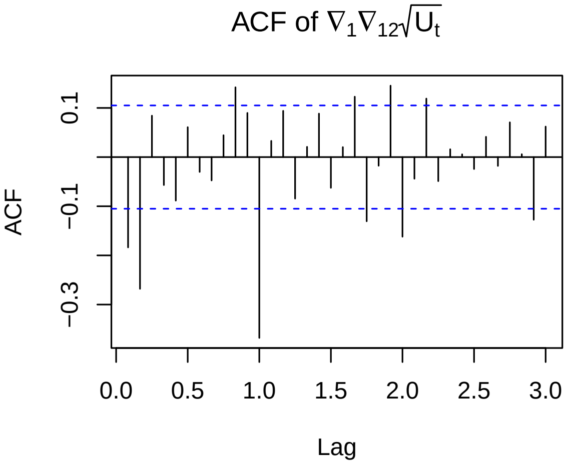




After differencing, there's no dominated frequency in the periodogram, and the histogram of $\nabla_{12}\nabla_1\sqrt{U_t}$ looks symmetric and almost Gaussian.

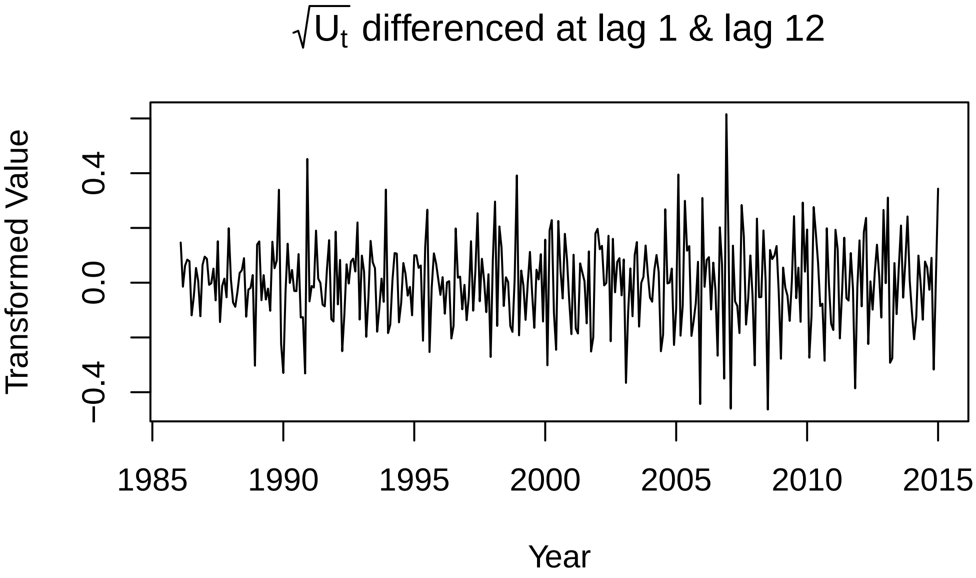
ACF and PACF



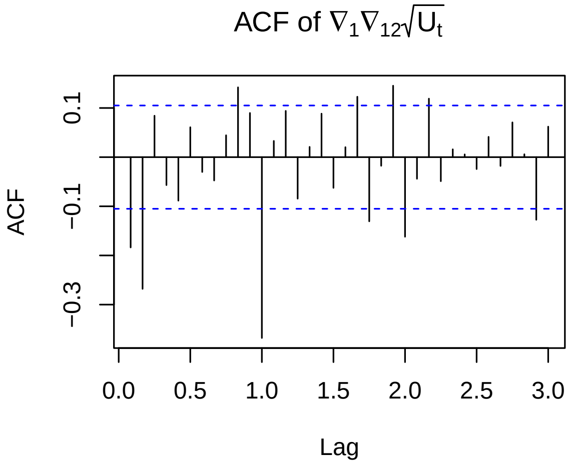


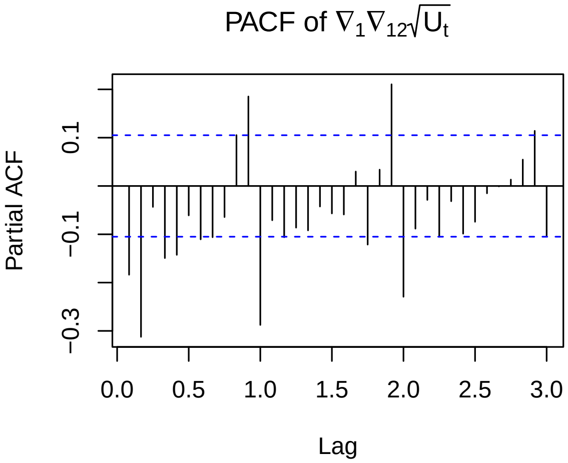
The top-left plot is the ACF of the raw data differenced at lag 1. The periodic ACF shows that there are still seasonal components. The top right plot is the ACF values of data after second differenced at lag=12, which decay corresponds to a stationary process. One may notice that the seasonal trend has been eliminated and the data(below) looks stable.

The plot below is the data after differencing. One may notice that there is no significant trend and seasonal component the variance looks stable over time, which satisfied the stationary requirement for the ARIMA model.



Preliminary Model Identification



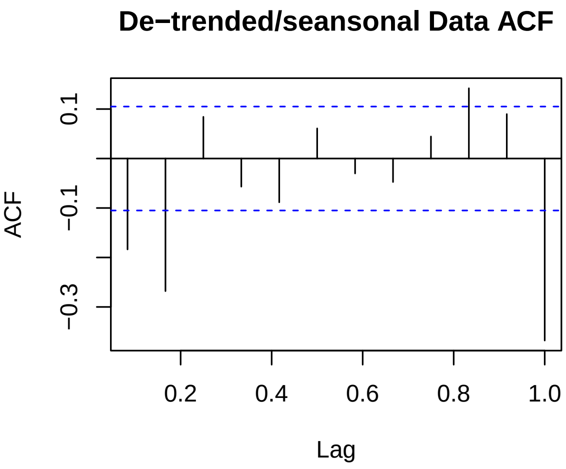


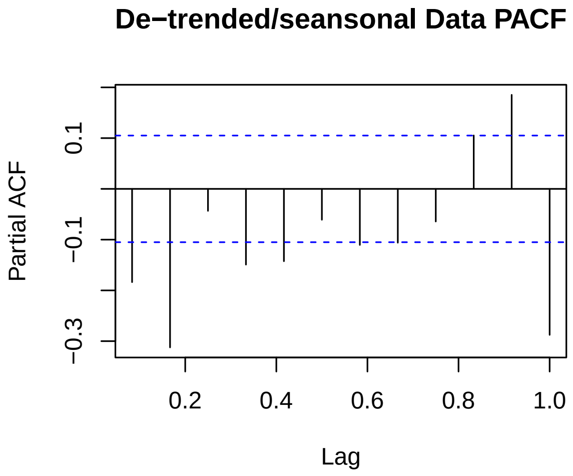
The above are the ACF and PACF in three periods. One may notice that the ACF is significant in one period at lag 0,12,24 and the PACF is significant at lag 12, 24, 36. In this case, a seasonal ARIMA model will be appropriate for model fitting and prediction.

From the plot I could notice the following on the Seasonal Components:

ACF: The significant lags in seasonal lags suggested that there might be a seasonal component Q. I choose $Q = 1, 2$ to be the candidate values for Q. The reason why I don't choose 3 is because acf at lag 36 already sits on the border of the 95%CI. Using Bartlett's formula, one may notice that that these borders are too conservative for MA models.

PACF: The significant lags in seasonal lags suggested that there might be a seasonal component P. However, for a pure $SMA(*)$ model would also have a ACF plot which its seasonal lag is significant with exponential decay. In this case, I choose $0, 1, 2$ to be the candidate values for P.





The above are the ACF and PACF in one period (s=12). One may notice that the ACF is significant in one period at lag 0,1,2,10 and the PACF is significat at 1,2,4,5,10,11. From the plot I could notice the following on Non-Seasonal Components:

ACF: The significant ACF at lags 1 and 2 suggests that there is a short-term correlation in the data, which indicated that the values are dependent on the immediate previous ones. The significant autocorrelations at higher lags like 10 might indicative of a seasonal pattern, suggested that Q should at least greater than 1. With these information, I choose $q = 2$ to be the candidate values for q.

PACF: Same as ACF, the significant PACF at lag 1,2,4,5 suggests that there is a short-term correlation. The significat PACF values at higher lags 10, 11, may comes from both the Seasonal Autoregressive (SAR) part and the Seasonal Moving Average (SMA) part. Notice that the PACF has a strong peak at lag 2 and a weak peak at lag 5, I will choose $p = 2, 5$ to be the candidate values for p.

Result

With the ascending sort of AICc, I have the following models:

- SARIMA(2,1,2)(1,1,2)[12] with an AICc of -498.6206
- SARIMA(2,1,1)(0,1,1)[12] with an AICc of -498.5167
- SARIMA(5,1,2)(0,1,1)[12] with an AICc of -498.4168
- SARIMA(5,1,2)(0,1,2)[12] with an AICc of -497.6795
- SARIMA(5,1,2)(1,1,2)[12] with an AICc of -497.6725

Coefficients Estimation and Diagnostic Checking

Coefficient Estimation

Let X_t denotes our transformed data ($\sqrt{U_t}$), B denotes the the backshift operator.

Model 1: $SARIMA(2,1,2)(1,1,2)[12]$ with the coefficient table and $AICc = -498.62$

	ar1	ar2	ma1	ma2	sar1	sma1	sma2
	0.0831	0.1778	-0.4593	-0.4479	-0.4276	-0.2511	-0.4386
s.e.	0.3138	0.1841	0.3039	0.2887	0.2283	0.2144	0.1596

Looking at this table, one may find that coefficient of ϕ_1 are closed to 0 within one standard deviations and hence not significant. By setting $\phi_1 = 0$, I get the following refined model with $AICc = -500.54$.

	ar2	ma1	ma2	sar1	sma1	sma2
	0.2236	-0.3813	-0.5215	-0.4245	-0.2533	-0.4353
s.e.	0.0718	0.0501	0.0591	0.2297	0.2162	0.1602

Then we find that 0 falls in the 1.96 standard deviations of Θ_1. By setting $\Theta_1 = 0$, the model is further refined with a lower $AICc = -500.7721$

	ar2	ma1	ma2	sar1	sma2
	0.2224	-0.3876	-0.5164	-0.7011	-0.5993
s.e.	0.0719	0.0500	0.0595	0.0547	0.0577

After refined, one could find that all coefficients are significantly different form 0, and AICc is also decrease from the originial one. Hence I choose this as the final refined *model1* with formula expression:

$$(1 - 0.2224B^2)(1 + 0.7011B^{12})\nabla_1\nabla_{12}X_t = (1 - 0.3876B - 0.5164B^2)(1 - 0.5993B^{12})Z_t$$

Model 2: $SARIMA(5,1,2)(0,1,1)_{12}$ with the coefficient table and $AICc = -498.42$

	ar1	ar2	ar3	ar4	ar5	ma1	ma2	sma1
	-0.3847	0.4001	-0.0174	0.0095	-0.1283	0.0106	-0.8503	-0.7784
s.e.	0.0750	0.0767	0.0699	0.0678	0.0594	0.0564	0.0547	0.0382

With same process, I first set the parameter $\phi_3, \phi_4, \theta_1 = 0$ since they are closed to 0 within one standard deviation. By doing this I get the following refined model $SARIMA(5,1,2) \times (0,1,1)_{12}$ with the following coefficient table and a decreased $AICc = -504.2583$.

	ar1	ar2	ar5	ma2	sma1
	-0.3814	0.4080	-0.1390	-0.8463	-0.7756
s.e.	0.0499	0.0724	0.0486	0.0492	0.0362

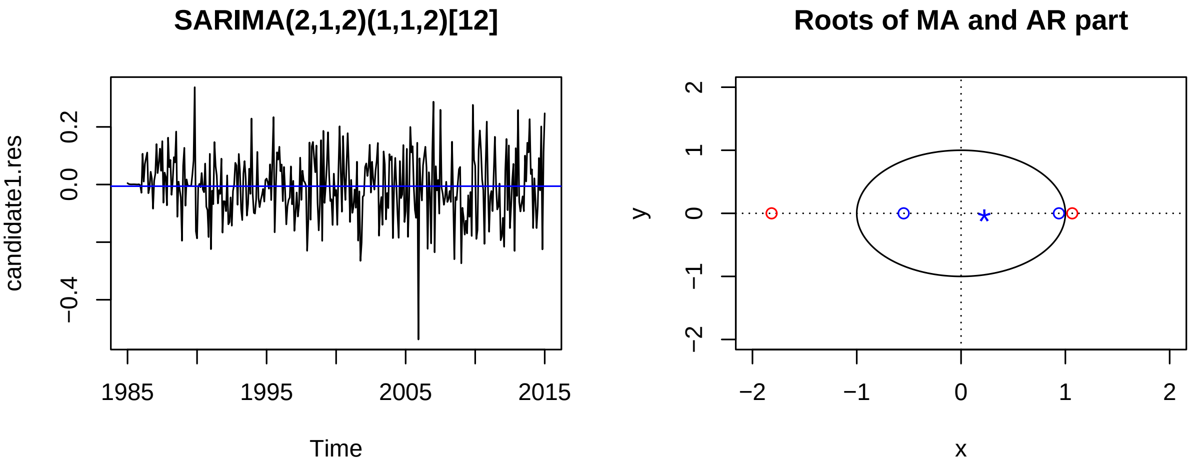
After refined, one could find that all coefficients are significantly different form 0, and AICc is also decrease from the originial one. Hence I choose this as the final refined *model1* with formula expression:

$$(1 + 0.3814B - 0.408B^2 + 0.139B^5)\nabla_1\nabla_{12}X_t = (1 - 0.8463B^2)(1 - 0.7756B^{12})Z_t$$

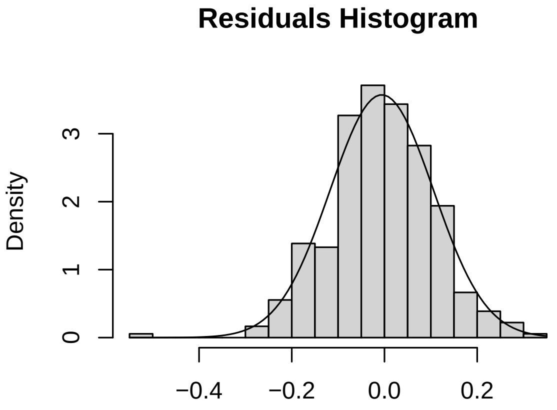
Rest of the Models: For the rest of the models one may notice that they have more parameters but have higher AICCs. This suggests that these additional parameters are not efficiently contributing to the model's performance. According to the principle of parsimony, these model are not the best choice and I should select the two simpliest with lowest AICcs. Thus, I decided to only take $SARIMA(2,1,2) \times (1,1,2)_{12}$ and $SARIMA(5,1,2) \times (0,1,1)_{12}$ to diagnostic checking.

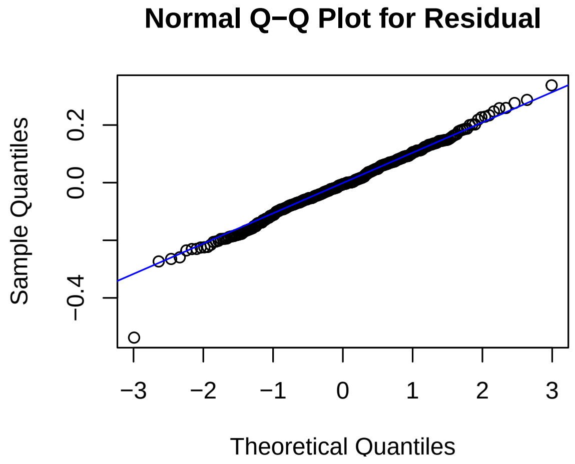
Diagnostic Checking

Model1: For model $SARIMA(2,1,2)(1,1,2)_{12}$,

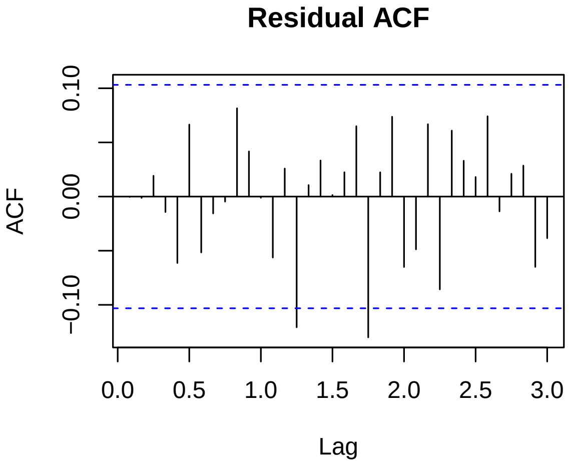


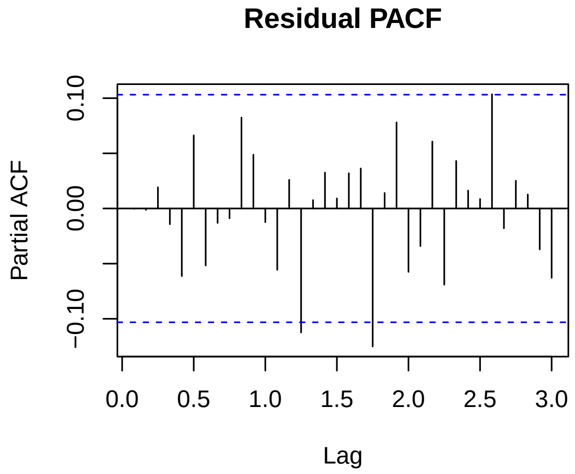
The top-left plot is residual of the model. One may notice that these residuals are fluctuate randomly around the *mean* zero showing a patterns of White Noise, which indicated that the model has captured all the patterns in the data. The top-right plot is the roots of Auto Regressive characteristic polynomial and Moving Average characteristic polynomial on the complex plane. One may notice that all the roots are outside of the unit circle. Also the absolute value of parameter of seasonal Moving Average component $|\Theta_1| < 1$, and seasonal Auto Regressive component $|\Phi_2|$ is less than 1 when $|\Phi_1 = 0|$. These evidence indicates that this model is stationary and invertible.


Residuals Histogram


Normal Q−Q Plot for Residual

The histogram above indicates that the residuals follows a normal distribution since it has a bell shape and follows the theoretical density curve of normal distribution. In the Q-Q plot, more than 95% of the points are between -2 and 2 and the points lie approximately along the 45-degree reference line. This also indicates that the residuals follow normal distribution, which is a property of White Noise.


Residual ACF

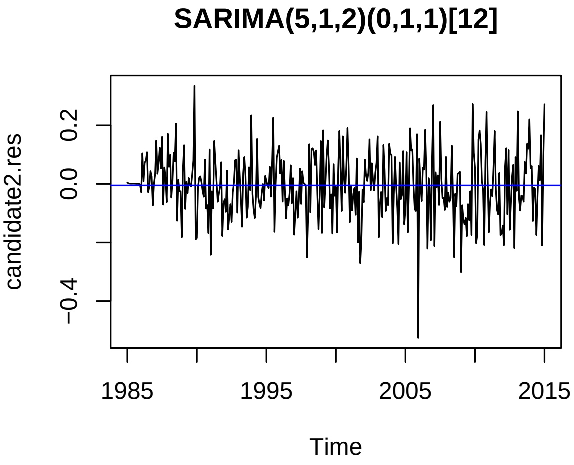

Residual PACF

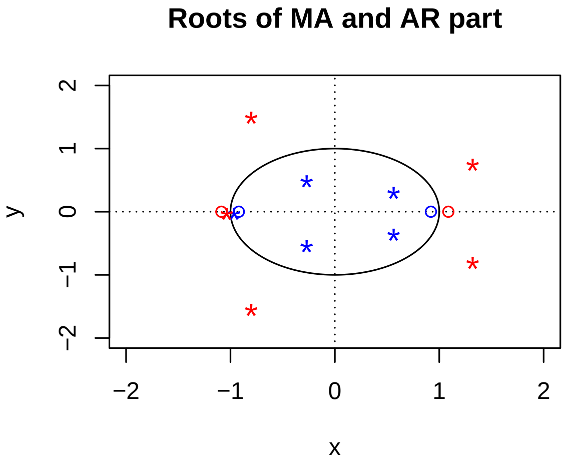
The ACF plot above shows that all the ACF of residuals are within the confidence interval, suggested that the residual is a $MA(0)$ process. One may notice that there's one little peak in the PACF plot, hence *Box-Pierce Test*, *Ljung-Box Test*, and *Mcleod-Li Test* are employed with the following result

	Test Result
Shapiro-Wilk normality test	0.01567
Box-Pierce Test	0.4123
Ljung-Box Test	0.3737
Mcleod-Li Test	0.3258

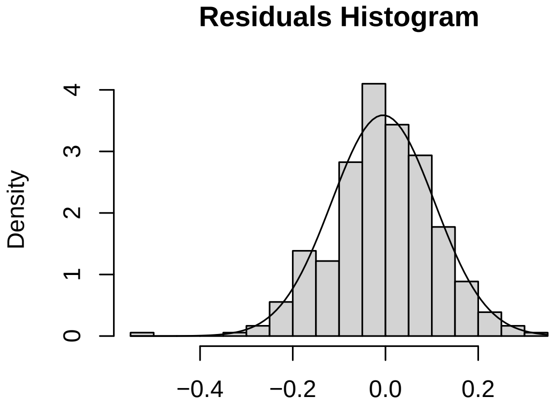
Notice that the residual pass all the tests except for Shapiro-Wilk normality test. Moreover, by plugged the residuals from into the Yule-Walker method, and the function automatically selected 0 for the Auto Regressive part. These evidence suggested that there's no autocorrelation in the process generated by the residual of the model, and hence suggested that this process is a White Noise. However, even though the distribution plot and Q-Q plot indicates that the residuals follow normal distribution, the residual didn't pass the Shapiro-Wilk normality test. This may because there's a significant outlier in 2005.

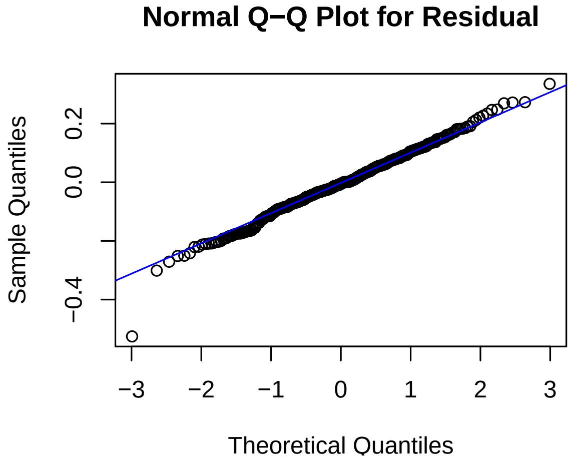
Model2: For model $SARIMA(5,1,2)(0,1,1)_{12}$,



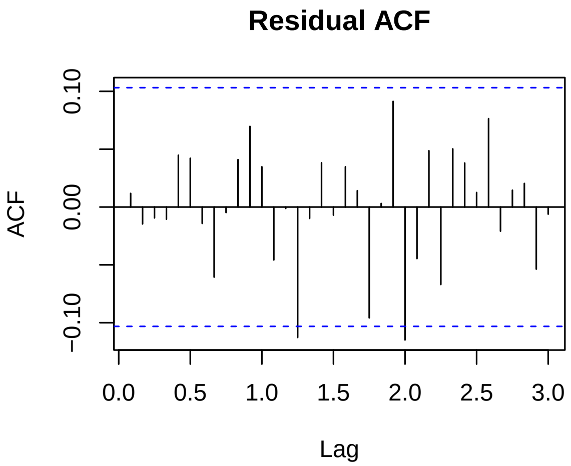


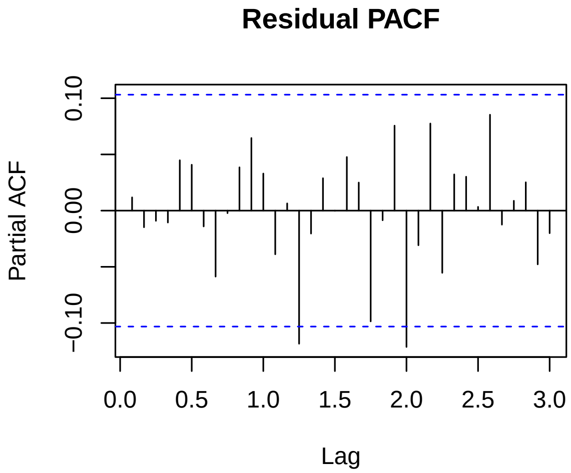
The top-left plot is residual of the model. It follows the same pattern as the residual of model1 that these residuals are fluctuate randomly around the *mean* zero showing a patterns of White Noise. This indicated that the model has captured all the patterns in the data. The top-right plot is the roots of Auto Regressive characteristic polynomial and Moving Average characteristic polynomial on the complex plane. One may notice that all the roots are outside of the unit circle. Also the absolute value of parameter of seasonal Moving Average component $|\Theta_1|$ and the seasonal Auto Regressive component $|\Phi_1|$ is less than 1. These evidence indicates that this model is stationary and invertible.





The histogram above indicates that the residuals follows a normal distribution since it has a bell shape and follows the theoretical density curve of normal distribution. In the Q-Q plot, more than 95% of the points are between -2 and 2 and the points lie approximately along the 45-degree reference line. This also indicates that the residuals follow normal distribution, which is a property of White Noise.





One may notice that there's one little peak in both ACF and PACF plot, hence *Box-Pierce Test*, *Ljung-Box Test*, and *Mcleod-Li Test* are employed with the following result

	Test Result
Shapiro-Wilk normality test	0.01913
Box-Pierce Test	0.5956
Ljung-Box Test	0.5563
Mcleod-Li Test	0.5317

Notice that the residual pass all the tests except for Shapiro-Wilk normality test. Moreover, by plugged the residuals from into the Yule-Walker method, and the function automatically selected 0 for the Auto Regressive part. These evidence suggested that there's no autocorrelation in the process generated by the residual of the model, and hence suggested that this process is a White Noise. Similar to model1, even though the distribution plot and Q-Q plot indicates that the residuals follow normal distribution, the residual didn't pass the Shapiro-Wilk normality test. This may because of the same reason there's a significant outlier in 2005.

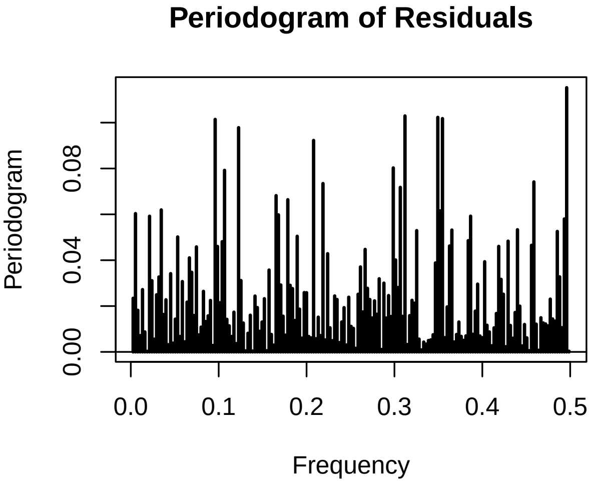
The empirical rule indicates that 99.7% of observations are within 3 standard deviations of the mean. To test our assumption that the outliers in residuals affect the Shapiro-Wilk normality test, I eliminate outliers that are more than 3 standard deviations away from the mean and test if the rest of residual could pass the Shapiro-Wilk normality test.

	Shapiro-Wilk normality test
Model1 Residuals Without Outliers	0.5323
Model2 Residuals Without Outliers	0.5256

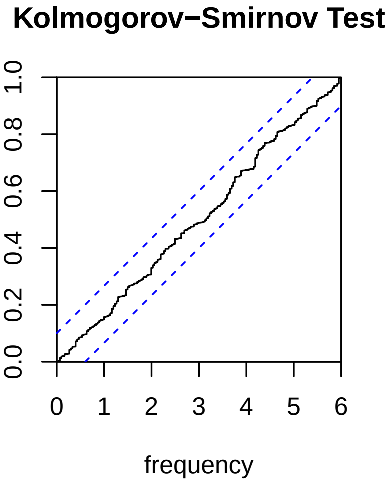
The result shows that outliers in residual did affect the normality test, by removing the outliers both model pass the Shapiro-Wilk test.

Spectral Analysis

To further examine if the residual of model1 and model2 are White Noise, I employed *Periodogram*, *Kolmogorov-Smirnov Test*, and *Fisher's test*.



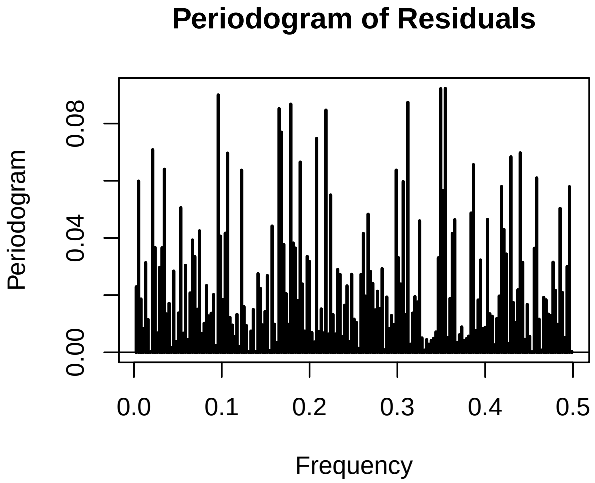
Periodogram of Residuals

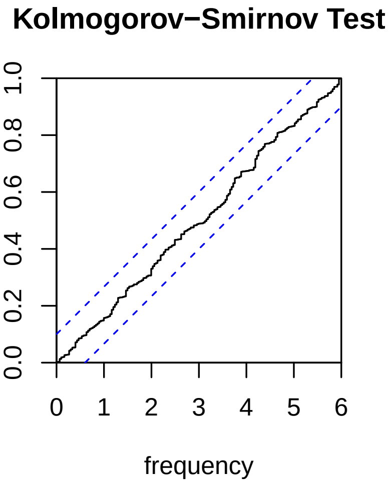


Kolmogorov−Smirnov Test

The left plot is the Periodogram of $SARIMA(2,1,2)(1,1,2)[12]$. Notice that there is no frequency dominates, which suggests that I have the White Noise. The right plot is the Kolmogorov-Smirnov Test where the function $C(x)$ never exits boundaries, which also suggested that there's no sufficient evidence to reject the null hypothesis that $H_0 : X_t$ is a Gaussian White Noise.

From the Fisher's test with the same null hypothesis. Notice that the $P(\xi_q \geq x) = 0.8242 > \alpha = 0.05$, I accept the null hypothesis and hence I concludes that there's no sufficient evidence that the residual is statistically different from a Gaussian White Noise.





Same spectral analysis are applied on the residual of model2 with $P(\xi_q \geq x) = 0.8317 > \alpha = 0.05$ from Fisher's test and I get the same conclusion that there's no sufficient evidence that the residual is statistically different from a Gaussian White Noise.

Final Selection

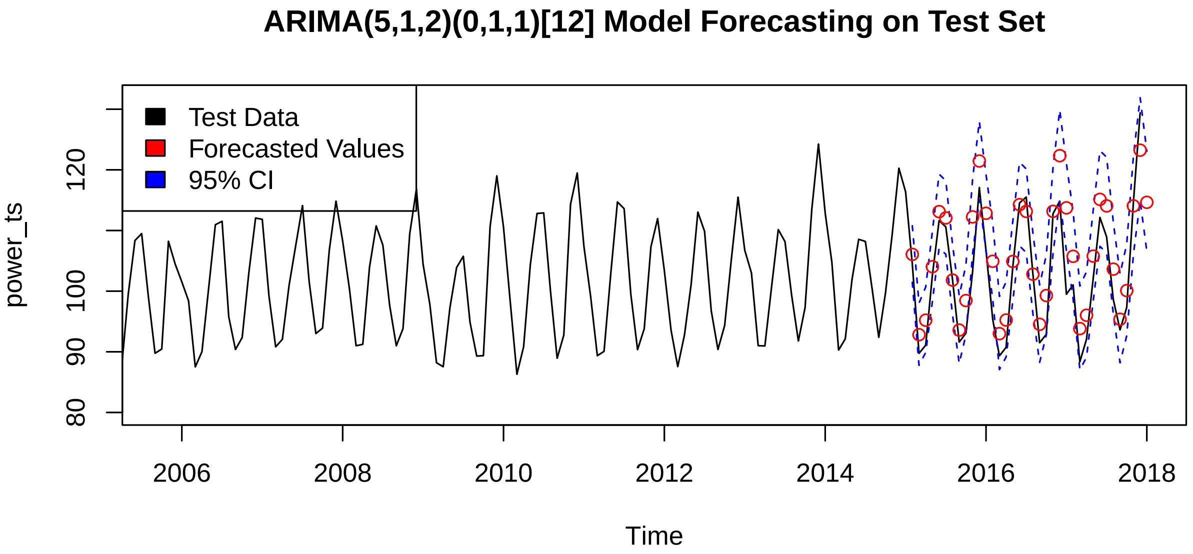
Since I've checked that the residuals of both models are White Noise and both models are stationary invertible models with same number of coeficients. By the principle of minimize AICc, model2 with a lower AICc wins. And thereby I select model2 with expression:

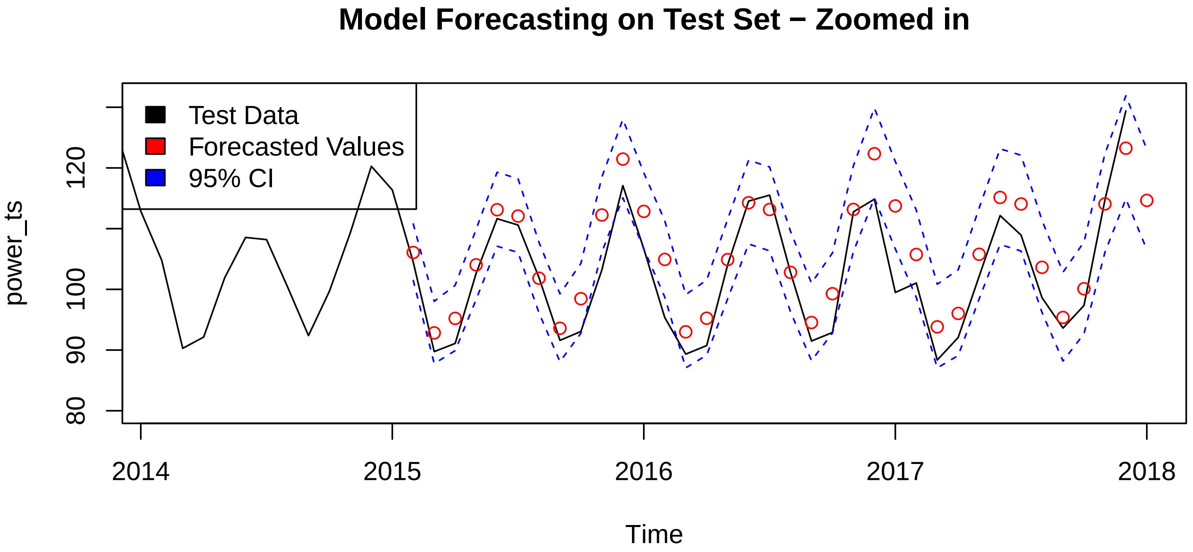
$$(1 + 0.3814B - 0.408B^2 + 0.139B^5)\nabla_1\nabla_{12}X_t = (1 - 0.8463B^2)(1 - 0.7756B^{12})Z_t$$

as the final model.

Forcasting

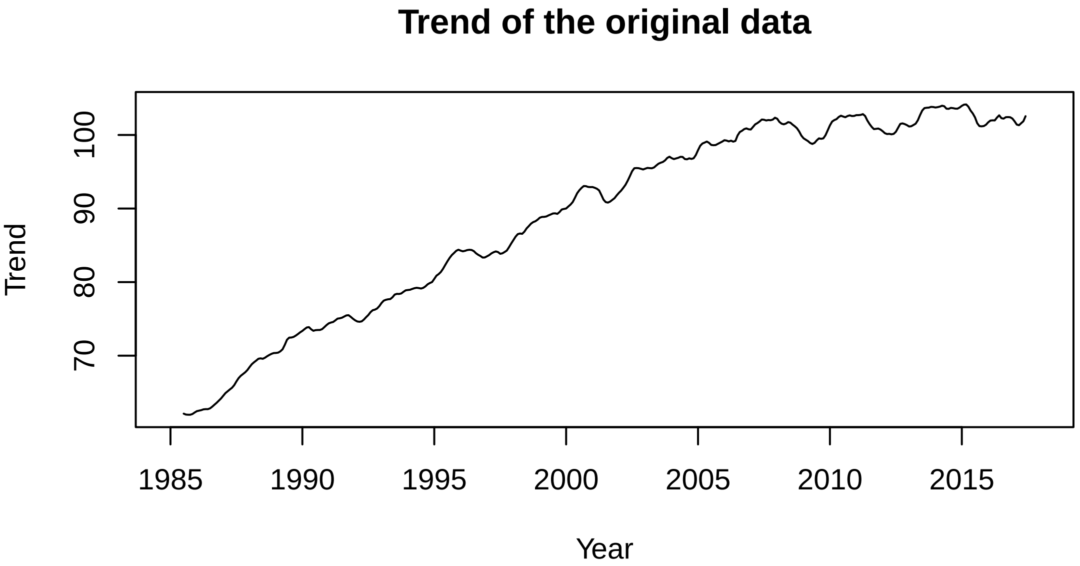
For forcasting, the predicted values was transformed back by an inverse Box-Cox transformation and an exponential transformation. The results are as following plot, where the black line is the real values from testing set, blue dash lines are the upper and lower bound of the 95% confidence interval, and the red points are the predicted values.





The first plot is the forecasted value using final model with the 95% confidence interval for the forecasted values with the real values in the testing set. The second plot is the zoom-in plots of the first plot so that I could see the details. From the plots, one could notice that most of the test sets are within the confidence intervals and the predicted values align with the observed value in the testing set.

However, the model has a trend of overestimate the real data. This could be explained that the original data has a increasing trend on mean which rate is getting slower after 2010. This change wasn't fully captured by the model because of lacking data.



Conclusion

The goal of this project is achieved. In this project, I successfully overcome challenges like non-stationarity and heteroscedasticity in the data, trained a SARIMA model, and used it to forecast the industrial production (IP) index. The model with the formula form

$$(1 + 0.3814B - 0.408B^2 + 0.139B^5)\nabla_1\nabla_{12}X_t = (1 - 0.8463B^2)(1 - 0.7756B^{12})Z_t$$

performs well on the test set. The residuals of this model has been tested with various analysis stratagy to be consisted with White Noise and implies that the model has capture the pattern of the time series.

Finally, I would like to thanks to Professor Feldman for her excellent lecture recordings, rich and organized class materials, and patience during the office hour. I am grateful for her help when I apply the theoretical knowledge from the classroom to real-world data. Additionally, I would like to thanks to Ms. Sandy Gao, who helped me with formatting the content in the paper using quarto.

Reference

Board of Governors of the Federal Reserve System (US), Industrial Production: Utilities: Electric and Gas Utilities (NAICS = 2211,2) [IPG2211A2N], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/IPG2211A2N, December 15, 2023.

Feldman, Raya. "Lecture Notes for Weeks 1-9." 2023, University of California: Santa Barbara, https://ucsb.instructure.com/courses/13750/modules/items/961165.

Feldman, Raya. "Summary of ACF and PACF Patterns for (S)ARIMA-2 pages" 2023, University of California: Santa Barbara, https://ucsb.instructure.com/courses/13750/modules/items/935878

Feldman, Raya. "Lab 5 Materials" 2023, University of California: Santa Barbara, https://ucsb.instructure.com/files/1643566/download?download_frd=1

Appendix

Code:

```r
## Required Library and Source Code##
library(astsa)
library(TSA)
library(GeneCycle)
library(xts)
library(zoo)
library(MASS)
library(forecast)
library(qpcR)
require(TSA)
library(tidyverse)
library(dplyr)
library(readr)
source("./scripts/rawdata2.R")
source("./scripts/plot.roots.R")

## rawdata2.R Raw Data Downloader##
# install.packages(c("devtools"))
# check file structure
root_dir <- rprojroot::find_rstudio_root_file()
setwd(root_dir)
if (!dir.exists("./data")){dir.create("./data")} # create "./data" is not exist and set up
if (!dir.exists("./result")){dir.create("./result")} # create "./data" is not exist and se
data_dir <- file.path(root_dir, "data")
scripts_dir <- file.path(root_dir, "scripts")
result_dir <- file.path(root_dir, "result")

url <- "https://fred.stlouisfed.org/plot/fredplot.csv?bgcolor=%23e1e9f0&chart_type=line&dr
file <- "./data/IPG2211A2N.csv"

download.file(url, destfile = file, mode = "wb")
data_file <- "./data/Electric_Production.csv"
data <- read.csv(data_file, header = TRUE)


## Setting Up Environment and Variable Carriers ##

rm(list = ls())
```

```r
root_dir <- rprojroot::find_rstudio_root_file()
setwd(root_dir)
refine_models <- list()
aiccs <- list()
model_saver <- list()

## Data Processing ##
data$DATE <- as.Date(data$DATE, format="%m/%d/%Y")
data_1985 <- data[data$DATE > as.Date("1985-01-01"), ]
power_ts <- ts(data_1985$IPG2211A2N, start=c(1985, 1), frequency=12)

op <- par(mfrow=c(1,2),cex=0.9)
split_point <- c(2015, 1)
ts_start <- start(power_ts)
ts_end <- end(power_ts)

# Create the training and testing set
train_set <- window(power_ts, start=ts_start, end=split_point)
test_set <- window(power_ts, start=split_point, end=ts_end)

# Plotting the training and test sets
plot(train_set, main="Train Time Series", xlab="Time", ylab="Consumption")
plot(test_set, main="Test Time Series", xlab="Time", ylab="Consumption")
par(op)

## Plot and Analyze ##
# Trend and Seasonal Component
op <- par(mfrow=c(1,2),cex=0.9)
fit <- lm(train_set ~ time(train_set))
ts.plot(train_set,gpars=list(xlab="Year",ylab="Consumption"), main='TS Plot with Linear Tr
abline(h=mean(train_set), col="red")
abline(fit, col="blue", lwd=2)
plot(decompose(train_set)$seasonal, xlab="Year",ylab="Season Component", main='TS Plot wit
par(op)

# Periodogram
require(TSA)
op <- par(mfrow=c(1,1))
TSA::periodogram(train_set, main="Periodogram for train_set data")
abline(h=0)
par(op)
```

```
periodogram_result <- TSA::periodogram(train_set)

# Extract the frequencies and corresponding spike values
frequencies <- periodogram_result$freq
spike_values <- periodogram_result$spec

# Combine frequencies and spike values into a data frame
periodogram_data <- data.frame(Frequency = frequencies, Spike = spike_values)

# Sort the data frame by spike values in descending order
sorted_periodogram_data <- periodogram_data[order(-periodogram_data$Spike), ]

# Rolling variance
op <- par(mfrow=c(1,2),cex=0.9)
# Calculate rolling variance
rolling_variance <- rollapply(train_set, width = 12, FUN = var, align = 'right', fill = NA
fit <- lm(rolling_variance~ time(rolling_variance))
plot(rolling_variance, type = 'l', xlab = 'Time', ylab = 'Variance', main = 'Rolling Varia
abline(fit, col="red")
abline(h=mean(rolling_variance, na.rm = TRUE), col="blue")
hist(train_set, col="light blue", xlab="", main="Histogram of the original data")
par(op)

## Transformation and Differencing##
# Box-Cox Plot
library(MASS)
t = 1:length(train_set)
fit = lm(train_set ~ t)
bcTransform = boxcox(train_set ~ t,plotit = TRUE)
train_set.sqrt = sqrt(train_set)

# Data Transformation
op <- par(mfrow=c(1,2),cex=0.9)
fit <- lm(train_set.sqrt ~ time(train_set.sqrt))
ts.plot(train_set.sqrt,gpars=list(xlab="Year",ylab="Consumption"), main='TS Plot after tra
abline(h=mean(train_set.sqrt), col="red")
abline(fit, col="blue", lwd=2)
hist(train_set.sqrt, col="light blue", xlab="", main="Histogram of the transformed data")
par(op)

# Difference Variance with Different Lag
```

```r
variances <- numeric(20)
variances[1] <- var(diff(train_set.sqrt))
# Loop over the lags from 1 to 10
for (i in 1:19) {
  # Calculate the differenced data at lag i
  diff_data <- diff(diff(train_set.sqrt), lag = i)

  # Calculate and store the variance of the differenced data
  variances[i+1] <- var(diff_data)
}

# Create a scatter plot of the variances
plot(1:20, variances, type = "p", col = "blue",
     main = "Variance of Differenced Training Set at Different Lags",
     xlab = "Lag Difference (i)", ylab = "Variance", xaxt = "n")
axis(1, at = 1:20, labels = 0:19)
par(mfrow=c(1, 1))

# Transformed Value Plot
op <- par(mfrow=c(1,1))
plot(diff(diff(train_set.sqrt), lag=12),xlab="Year", ylab="Transformed Value")
par(op)

# Transformed Periodogram
op <- par(mfrow=c(1,2),cex=0.9)
TSA::periodogram(diff(diff(train_set.sqrt), lag=12), main=TeX(r'(Periodogram of $\nabla_1\
abline(h=0)
hist(diff(diff(train_set.sqrt), lag=12), col="light blue", xlab="", main=TeX(r'(Histogram
par(op)

# ACF and PACF
op <- par(mfrow=c(1,2),cex=0.9)
acf(diff(train_set), lag.max = 12*3, main = TeX(r'(ACF of $\nabla_1\sqrt{U_t}$)')) # ACF
acf(diff(diff(train_set),lag=12), lag.max = 12*3, main = TeX(r'(ACF of $\nabla_1\nabla_{12
par(op)

# Plot Transformed data
op <- par(mfrow=c(1,1))
plot(diff(diff(train_set.sqrt), lag=12),xlab="Year", ylab="Transformed Value", main=TeX(r'
par(op)
```

```
## Preliminary Model Identification ##
# ACF and PACF
op <- par(mfrow=c(1,2),cex=0.9)
acf(train_data, lag.max = 12*3, main = 'De-trended/seansonal Data ACF') # ACF
pacf(train_data, lag.max = 12*3, main = 'De-trended/seansonal Data PACF') # PACF
par(op)

op <- par(mfrow=c(1,2),cex=0.9)
acf(train_data, lag.max = 12, main = 'De-trended/seansonal Data ACF') # ACF
pacf(train_data, lag.max = 12, main = 'De-trended/seansonal Data PACF') # PACF
par(op)


# Lop of Model Fitting
# Given values
p_vals <- c(2,5)
q_vals <- c(0,1,2)
P_vals <- c(0,1)
Q_vals <- c(0,1)
d_val <- 1
D_val <- 1
s_val <- 12

library(forecast)
library(qpcR)

# Initialize an empty list to store AICc values and model specifications
aiccs <- list()
model_saver <- list()

# Given values
p_vals <- c(2,5)
q_vals <- c(2)
P_vals <- c(0,1)
Q_vals <- c(1,2)
d_val <- 1
D_val <- 1
s_val <- 12

# Loop over the candidate variables for p, q, P, Q
for(p in p_vals) {
  for(P in P_vals) {
    for(q in q_vals) {
      for(Q in Q_vals) {
        # Fit the SARIMA model with the given parameters
        model <- Arima(train_set.sqrt, order=c(p, d_val, q), seasonal=c(P, D_val, Q), meth
        model_name <- paste("SARIMA(", p, ",", d_val, ",", q, ")(", P, ",", D_val, ",", Q,
        model_saver[[model_name]] <- model
        # Calculate the AICc and store in the list with the corresponding model specificat
        aiccs[[model_name]] <- AICc(model)
      }
    }
  }
}

# AICC Sorting
# Convert the list to a named vector
aiccs_vector <- unlist(aiccs)
# Sort the vector in ascending order of AICc values
sorted_aiccs <- sort(aiccs_vector)
top_five_names <- names(head(sorted_aiccs, 5))
top_five_values <- head(sorted_aiccs, 5)


## Coeficient Estimation ##

# Model 1: SARIMA(2,1,2)(1,1,2)[12]
refine_models[['SARIMA(2,1,2)(1,1,2)[12]']] <- arima(train_set.sqrt, order=c(2,1,2), seaso
summary(refine_models[['SARIMA(2,1,2)(1,1,2)[12]']])
AICc(refine_models[['SARIMA(2,1,2)(1,1,2)[12]']])
refine_models[['SARIMA(2,1,2)(1,1,2)[12]']] <- arima(train_set.sqrt, order=c(2,1,2), seaso
summary(refine_models[['SARIMA(2,1,2)(1,1,2)[12]']])
AICc(refine_models[['SARIMA(2,1,2)(1,1,2)[12]']])
refine_models[['SARIMA(2,1,2)(1,1,2)[12]']] <- arima(train_set.sqrt, order=c(2,1,2), seaso
summary(refine_models[['SARIMA(2,1,2)(1,1,2)[12]']])
AICc(refine_models[['SARIMA(2,1,2)(1,1,2)[12]']])

# Model 2: SARIMA(5,1,2)(0,1,1)[12]
refine_models[['SARIMA(5,1,2)(0,1,1)[12]']] <- arima(train_set.sqrt, order=c(5,1,2), seaso
summary(refine_models[['SARIMA(5,1,2)(0,1,1)[12]']])
AICc(refine_models[['SARIMA(5,1,2)(0,1,1)[12]']])
refine_models[['SARIMA(5,1,2)(0,1,1)[12]']] <- arima(train_set.sqrt, order=c(5,1,2), seaso
```

```r
summary(refine_models[['SARIMA(5,1,2)(0,1,1)[12]']])
AICc(refine_models[['SARIMA(5,1,2)(0,1,1)[12]']])


## Diagnostic Checking ##
candidate_model <- list()
candidates = c("SARIMA(2,1,2)(1,1,2)[12]","SARIMA(5,1,2)(0,1,1)[12]")
for (model_name in candidates) {
  candidate_model[[model_name]] <- refine_models[[model_name]]
}
candidate1 <- candidate_model[[1]]
candidate2 <- candidate_model[[2]]

# Residual and Root Plot of Candidate 1
op <- par(mfrow=c(1,2),cex=0.9)
candidate1.res <- residuals(candidate1)
candidate2.res <- residuals(candidate2)
plot.ts(candidate1.res, main = 'SARIMA(2,1,2)(1,1,2)[12]')
abline(h=mean(candidate1.res), col="blue")
plot.roots(polyroot(c(1,-0.3876,-0.5164)),polyroot(c(1, -0.22246)), main = 'Roots of MA an
par(op)

# Normal Distribution Check
op <- par(mfrow = c(1,2),cex=0.9)
hist(candidate1.res, breaks = 40, xlab="", prob=TRUE,
     main = 'Residuals Histogram')
m.candidate1 <- mean(candidate1.res)
std.candidate1 <- sqrt(var(candidate1.res))
curve(dnorm(x,m.candidate2,std.candidate1), add=TRUE )
qqnorm(candidate1.res,main= "Normal Q-Q Plot for Residual")
qqline(candidate1.res,col="blue")
par(op)

# Residual ACF and PACF
op <- par(mfrow = c(1,2),cex=0.9)
acf(candidate1.res, lag.max=12*3,main='Residual ACF')
pacf(candidate1.res, lag.max=12*3,main='Residual PACF')
par(op)

# Tests
shapiro.test(candidate1.res)
```

```r
Box.test(candidate1.res, type=c("Box-Pierce"), lag = sqrt(length(train_set.sqrt)), fitdf =
Box.test(candidate1.res, type=c("Ljung-Box"), lag = sqrt(length(train_set.sqrt)), fitdf =
Box.test((candidate1.res)^2, type=c("Ljung-Box"), lag = sqrt(length(train_set.sqrt)), fitd
ar(candidate1.res, aic = TRUE, order.max = NULL, method = c("yule-walker"))

# Residual and Root Plot of Candidate 2
op <- par(mfrow=c(1,2),cex=0.9)
candidate2.res <- residuals(candidate2)
plot.ts(candidate2.res, main = 'SARIMA(5,1,2)(0,1,1)[12]')
abline(h=mean(candidate2.res), col="blue")
plot.roots(polyroot(c(1,0,-0.8463)),polyroot(c(1, 0.3814, -0.4080, 0, 0, 0.1390)), main =
par(op)

# Normal Distribution Check
op <- par(mfrow = c(1,2),cex=0.9)
hist(candidate2.res, breaks = 40, xlab="", prob=TRUE,
     main = 'Residuals Histogram')
m.candidate2 <- mean(candidate2.res)
std.candidate2 <- sqrt(var(candidate2.res))
curve(dnorm(x,m.candidate2,std.candidate2), add=TRUE )
qqnorm(candidate2.res,main= "Normal Q-Q Plot for Residual")
qqline(candidate2.res,col="blue")
par(op)

# Residual ACF and PACF
op <- par(mfrow = c(1,2),cex=0.9)
acf(candidate1.res, lag.max=12*3,main='Residual ACF')
pacf(candidate1.res, lag.max=12*3,main='Residual PACF')
par(op)

# Tests
shapiro.test(candidate2.res)
Box.test(candidate2.res, type=c("Box-Pierce"), lag = sqrt(length(train_set.sqrt)), fitdf =
Box.test(candidate2.res, type=c("Ljung-Box"), lag = sqrt(length(train_set.sqrt)), fitdf =
Box.test((candidate2.res)^2, type=c("Ljung-Box"), lag = sqrt(length(train_set.sqrt)), fitd
ar(candidate1.res, aic = TRUE, order.max = NULL, method = c("yule-walker"))

# Shapiro Test on Outliers Removed Residuals
# model1
mean_val1 <- mean(candidate1.res)
sd_val1 <- sd(candidate1.res)
```

```
threshold1 <- 3 * sd_val1

# Remove outliers
cleaned_data1 <- candidate1.res[abs(candidate1.res - mean_val1) <= threshold1]
shapiro.test(cleaned_data1)

# model2
mean_val2 <- mean(candidate2.res)
sd_val2 <- sd(candidate2.res)
threshold2 <- 3 * sd_val2

# Remove outliers
cleaned_data2 <- candidate2.res[abs(candidate2.res - mean_val2) <= threshold2]
shapiro.test(cleaned_data2)

## Spectral Analysis
# Periodogram and Kolmogorov-Smirnov Test
op <- par(mfrow=c(1,2),cex=0.9)
TSA::periodogram(candidate1.res, main="Periodogram of Residuals")
abline(h=0)
cpgram(candidate1.res, main = "Kolmogorov-Smirnov Test")
par(op)

# Fisher's Test
fisher.g.test(candidate2.res)

# Periodogram and Kolmogorov-Smirnov Test
op <- par(mfrow=c(1,2),cex=0.9)
TSA::periodogram(candidate2.res, main="Periodogram of Residuals")
abline(h=0)
cpgram(candidate1.res, main = expression("Kolmogorov-Smirnov Test"))
par(op)

# Fisher's Test
fisher.g.test(candidate2.res)

## Forcasting ##
library(forecast)
pred<- predict(candidate2, n.ahead = 36) # To produce plot with 10 forecast on data
U <- pred$pred + 2*pred$se # Upper bound of prediction interval
L <- pred$pred - 2*pred$se # Lower bound of prediction interval
```

```
n_forecast <- 36

forecast_start <- as.numeric(tail(time(train_set), 1)) + 1/12
forecast_end <- forecast_start + (n_forecast-1)/12
forecast_time_points <- seq(from = forecast_start, to = forecast_end, by = 1/12)

orig_pred <- (pred$pred **2)
orig_U <- (U **2)
orig_L <- (L **2)

ts.plot(power_ts, xlim=c(time(power_ts)[250], forecast_end), ylim = c(min(power_ts), max(o
        main = 'ARIMA(5,1,2)(0,1,1)[12] Model Forecasting on Test Set')

lines(forecast_time_points, y = orig_U, col = "blue", lty = 2)
lines(forecast_time_points, y = orig_L, col = "blue", lty = 2)
points(forecast_time_points, orig_pred, col = "red")

legend("topleft",
       legend = c('Test Data', 'Forecasted Values', '95% CI'),
       fill = c('black','red','blue'),
       border = "black")

ts.plot(power_ts, xlim=c(time(power_ts)[350], forecast_end), ylim = c(min(power_ts), max(o
        main = 'Model Forecasting on Test Set - Zoomed in')

# Plot zoom in
lines(forecast_time_points, y = orig_U, col = "blue", lty = 2)
lines(forecast_time_points, y = orig_L, col = "blue", lty = 2)
points(forecast_time_points, orig_pred, col = "red")

legend("topleft",
       legend = c('Test Data', 'Forecasted Values', '95% CI'),
       fill = c('black','red','blue'),
       border = "black")

plot(decompose(power_ts)$trend, xlab="Year",ylab="Trend", main='Trend of the original data
```