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

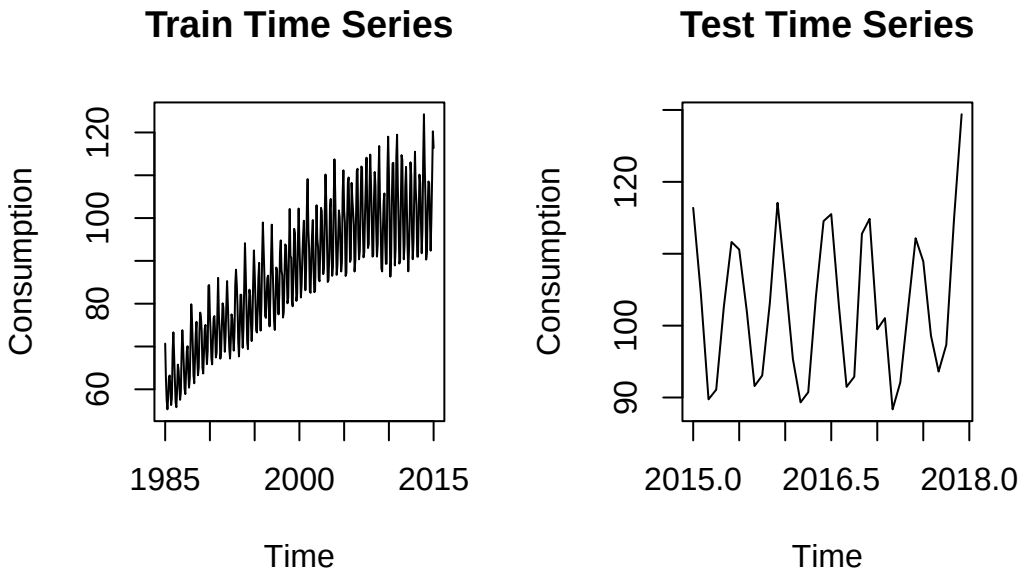
This study focuses on an analysis of a time series dataset from the Federal Reserve Economic Data (FRED), which tracks the monthly electric power generation in the United States (NAICS = 2211). The dataset spans from January 1985 to January 2018 with 397 observations, providing a vast and detailed view of electric production records over multiple decades. By examining this dataset, I 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 advanced time series analysis techniques to capture the patterns under the data 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 make forecasting in future monthly electric power generatio.

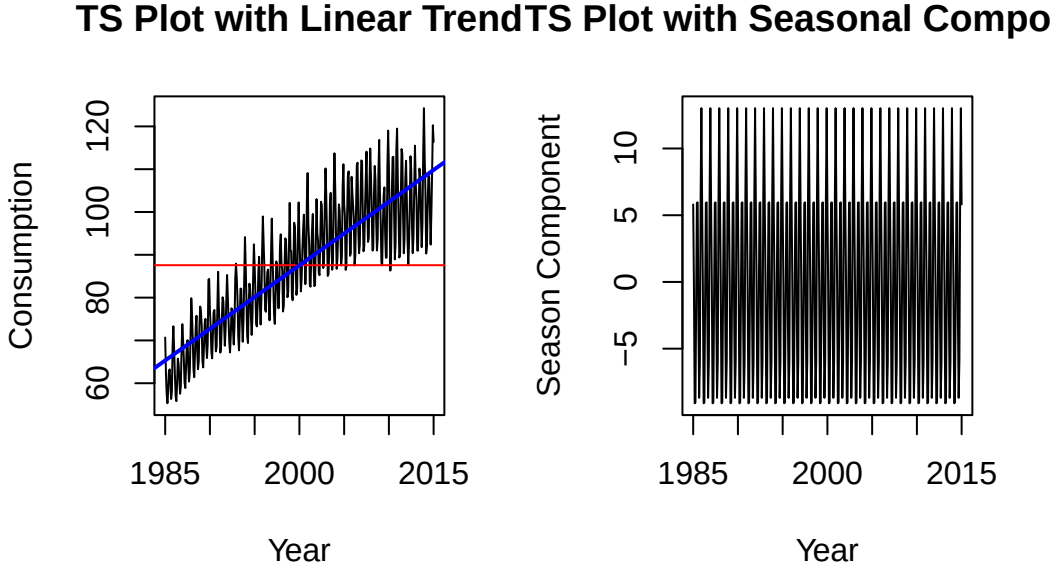
Data Preprocessing

In this part of preparation for the analysis, I'm setting up my time series data for forecasting. I've decided to split my dataset into two: a training set and a testing set. The split point I've chosen is January 2015. Using the window function, I separate my time series data into these two sets. The training set includes data from the beginning up to the start of 2015 with 348 observations, and the testing set comprises data from 2015 onwards includes 36 observations. The following two plots are the time series data in training set and testing set.



Plot and Analyze

When fitting an SARIMA model, we are holding the assumption that the time series is stationary. However, real-world data often exhibit trends and non-constant variance. Stationarity implies that the statistical properties of the time series—such as mean, variance, and autocorrelation—are constant over time. However, this assumption is violated with the `monthly electric production` dataset, which can exhibit trends, seasonality, and heteroscedasticity, where the variance changes over time.



The left plot depicts the `train_set` with a linear regression line. The slope of the blue line indicating that there's an underlying trend, which are systematic long-tern movements in the time series. A trend violates the stationarity assumption because the mean of the series changes over time.

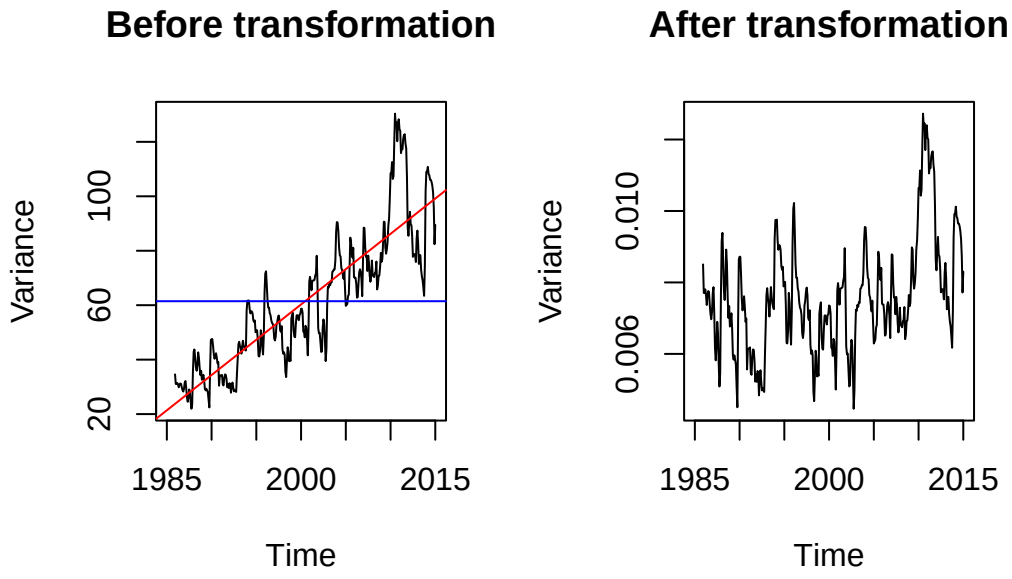
The right plot depicts the seasonal component in the `train_set` which clearly indicates a seasonal pattern. Seasonality refers to regular and predictable patterns or movements that recur over comparable periods. This can impact stationarity because it introduces systematic changes in both the mean and the variance at regular intervals.

Transformation and Differencing

To remove trends and stabilize the mean, we are going to use regular differencing and seasonal differencing, and applied transformation when needed.

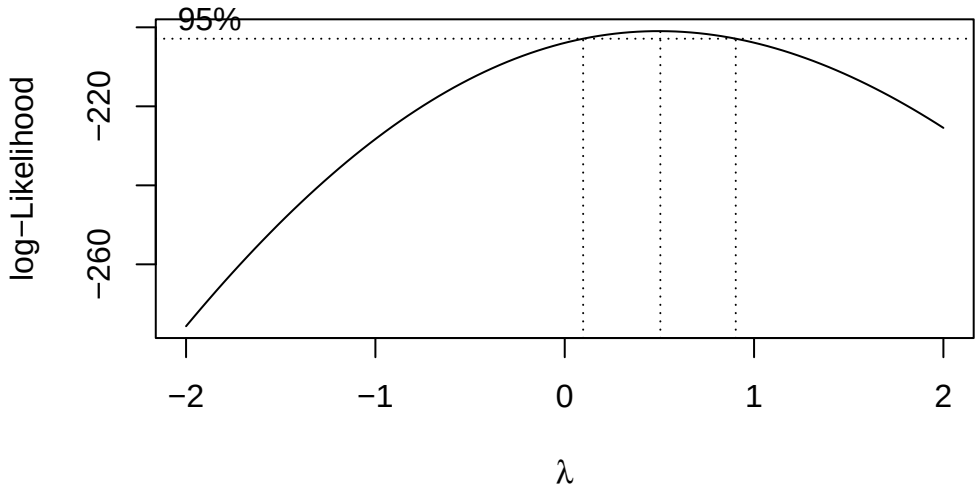
Transformation

When we considered applying a transformation, we first check if the variance is changing with the time. Choosing a window of 12 (one period) lags, we plot the rolling Variance vs. Time plot. From the left plot below, we find fluctuations and an increasing trend of variance over the time suggested that the variance is not stationary (heteroskedasticity). In this case a **Log-transformation** is considered to stabilized the variance.

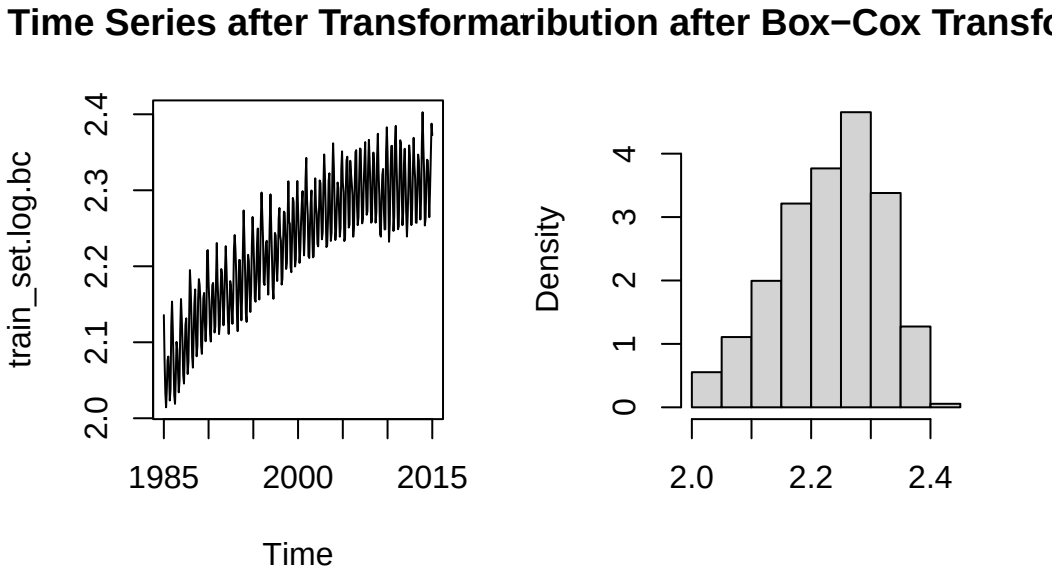


From the right plot above we could find that the variance of the preivous time-series data is stabled. Then we are going to normalize the variance of the data, we employed Box-Cox Plot

to identified the transformation we are going to use. The Box-Cox plot with 95% confidence interval of Box-Cox parameter λ is as follows.

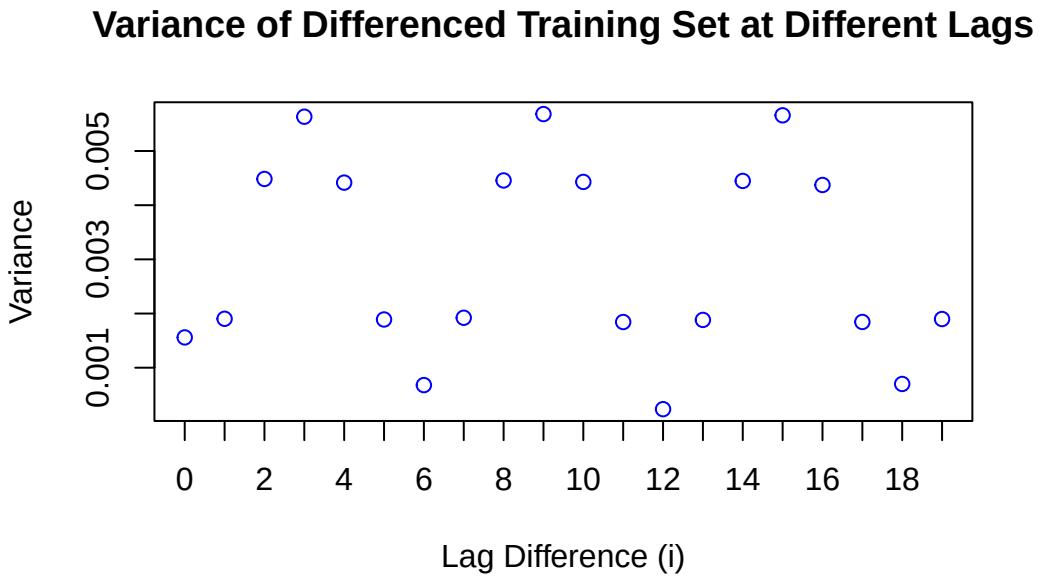


In plot, one may notice that 1 is **NOT** within the 95% Confidence Interval for the value of λ. In this case, it suggested that a transformation is needed to normalized the data to satisfy the assumption that the data is normally distributed for forecasting. After transformation the data has a a bell shape implies that the distribution of data is closed to the normal distribution and further transformation is not need.

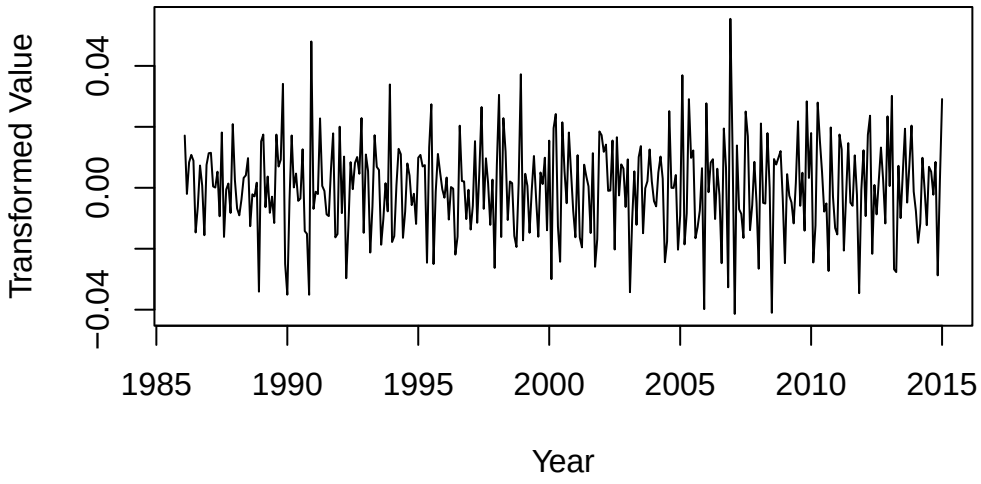


Differencing

Since we observed a trend in the data, a difference with lag 1 is applied to eliminate the trend. After this, we need to decided if further difference is needed. In the following plots, the x-axis represents the lag difference (k), ranging from 1 to 15, and the y-axis represents the variance of the differenced data.

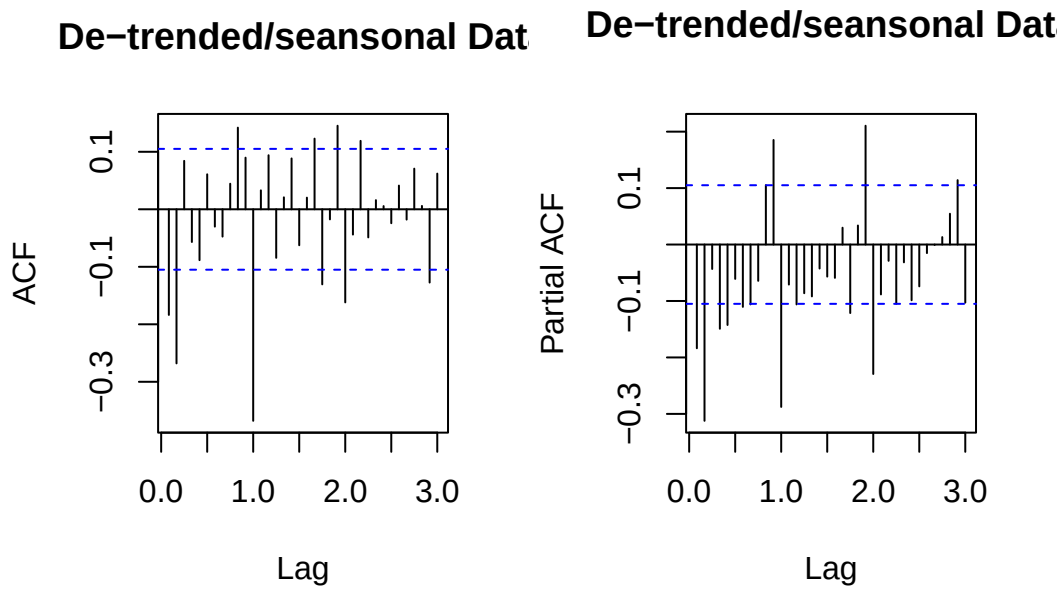


From the plot above, one may notice that after differencing at lags of 12 the times series has the lowest variance, indicating that difference at other lags might conclude to a over differencing. A seasonal trend of 12 is reasonable since there are 12 months in a year, and a lag of 12 captures the seasonality that occurs at the same time each year. Hence, I decided to difference the data once at lag one and difference twice at lag twelve.



The plot above 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



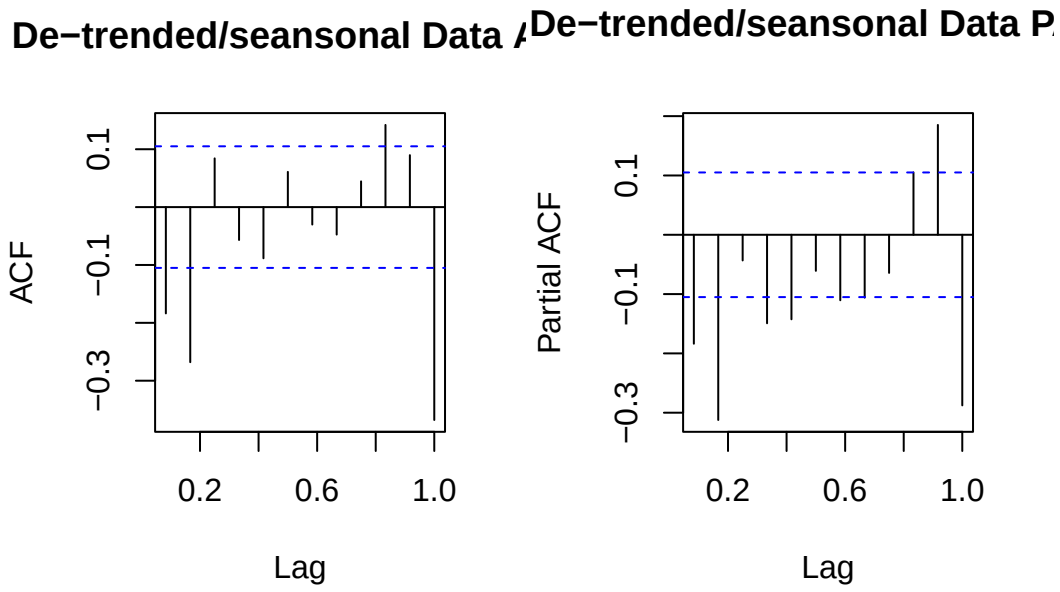
The above are the ACF and PACF in three periods. One may notice that the ACF is significant in one period at lag 0,12,24 and the PACF is significat at lag 12, 24, 36. In this case, a seasonal ARIMA model will be appropriate for model fitting and prediction.

From the plot we 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 $0, 1$ to be the candidate values for P. The reason why I don't choose 2 is because acf at lag 24 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 we 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. Since the ACF values at lag 1 and 2 are not that significant, I will choose $0, 1, 2$ 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 $2, 5$ to be the candidate values for p.

With these potential values of the parameters, I use for loop to fit different models and sorted the fited model using AICc:

Result

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

- Model1: SARIMA(5,1,2)(0,1,1)[12] with an AICc of -2116.004
- Model2: SARIMA(2,1,1)(1,1,1)[12] with an AICc of -2115.702
- Model3: SARIMA(2,1,1)(0,1,1)[12] with an AICc of -2115.337
- Model4: SARIMA(5,1,2)(1,1,1)[12] with an AICc of -2115.175
- Model5: SARIMA(2,1,2)(1,1,1)[12] with an AICc of -2114.613

Coefficients Estimation and Diagnostic Checking

Coefficient Estimation

Let X_t denotes our original data, B denotes the the backshift operator.

Model 1: $SARIMA(5,1,2)(0,1,1)[12]$ with the coefficient table and $AICc = -2118.52$

	ar1	ar2	ar3	ar4	ar5	ma1	ma2	sma1
	-0.3660	0.3982	-0.0307	0.0222	-0.1175	-0.0102	-0.8679	-0.7359
s.e.	0.0747	0.0724	0.0694	0.0650	0.0586	0.0555	0.0538	0.0402

Looking at this table, one may find that some coeficients are closed to 0 within one standard deviations and hence not significant. By setting them to zero, I get the following refined model with $AICc = -2124.06$.

	ar1	ar2	ar5	ma2	sma1
	-0.3637	0.4015	-0.1406	-0.8095	-0.8206
s.e.	0.0504	0.0773	0.0485	0.0545	0.0372

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:

$$(1 + 0.3637B - 0.4015B^2 + 0.1405B^5)(1 - B)(1 - B^{12})X_t = (1 - 0.8095B^2)(1 - 0.8206B^{12})Z_t$$

Model 2: $SARIMA(2,1,1) \times (1,1,1)_{12}$ with the following coefficient table and $AICc = -2117.94$

	ar1	ar2	ma1	sar1	sma1
	0.5436	-0.0408	-0.9162	0.1043	-0.8495
s.e.	0.0611	0.0581	0.0297	0.0672	0.0417

With same process, I first set the parameter $\phi_2 = 0$ since it is closed to 0. By doing this I get the following refined model $SARIMA(1,1,1) \times (1,1,1)_{12}$ with the following coefficient table and a decreasing $AICc = -2119.45$.

	ar1	ma1	sar1	sma1
	0.5312	-0.9232	0.1028	-0.8478
s.e.	0.0574	0.0256	0.0670	0.0414

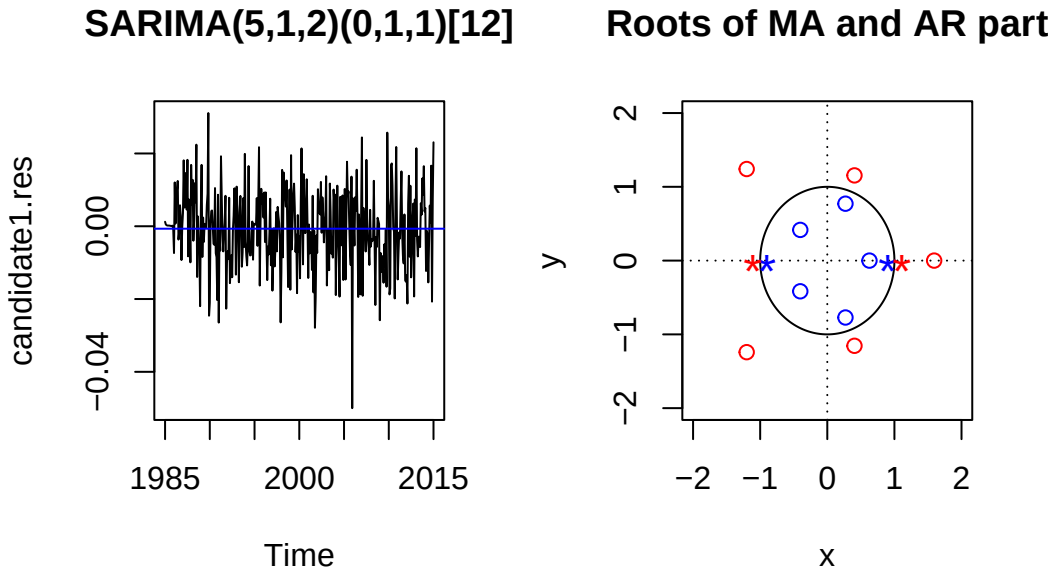
Notice that 0 falls in the 95% interval of Φ_1, I set $\Phi = 0$ and refined the model to $SARIMA(1,1,1) \times (0,1,1)_{12}$. However, by doing this the $AICc$ increse from -2119.45 to -2119.07, implies that these parameters didn't maximize the likelihood L_Y for all other samples Y in the process and thus decreasing the model performance of forcasting. Hence we choose the previous $SARIMA(1,1,1) \times (1,1,1)_{12}$ as the final *model2* with formula expression:

$$(1-0.5312B)(1-0.1028B^{12})(1-B)(1-B^{12})X_t = (1-0.9232B)(1-0.8478B^{12})Z_t$$

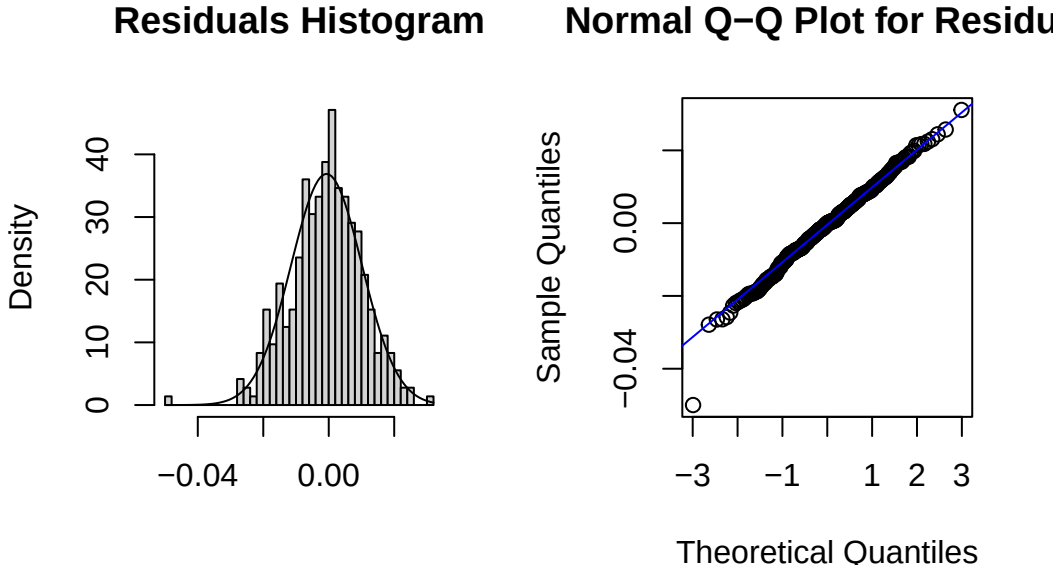
Rest of the Model: For the rest of the models one may notice that they have more parameters but have higher AICCs. This suggests that these additional parameters are not efficiently contributing to the model's performance. According to the principle of parsimony, these model are not the best choice and I should select the two simpliest *model1* and *model2*. Thus, I decided to only take $SARIMA(5,1,2)(0,1,1)[12]$ and $SARIMA(1,1,1) \times (0,1,1)_{12}$ to diagnostic checking.

Diagnostic Checking

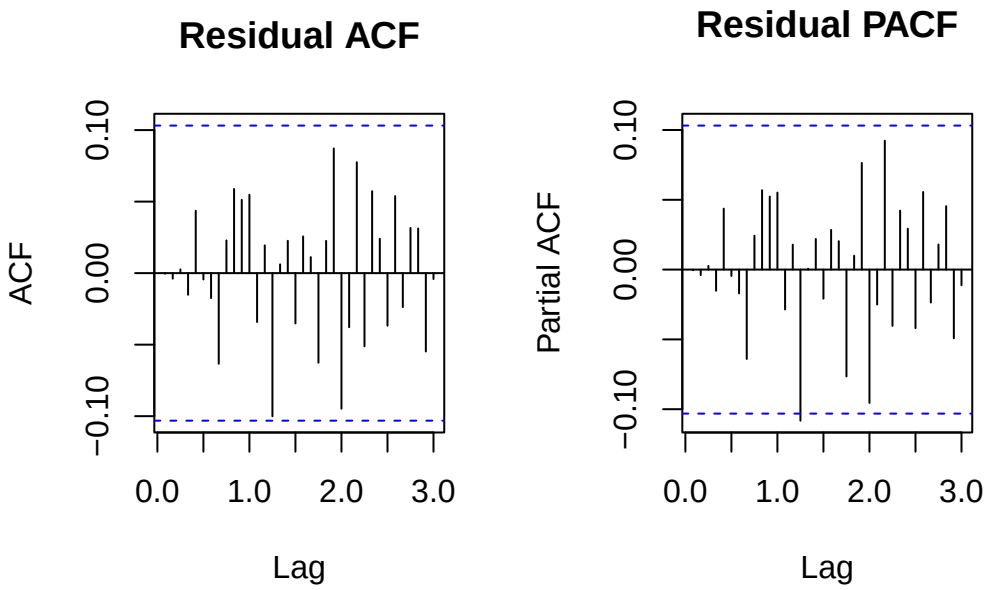
Model1: For model $SARIMA(5,1,2)(0,1,1)[12]$,



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 $|\Theta_1|$ 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.

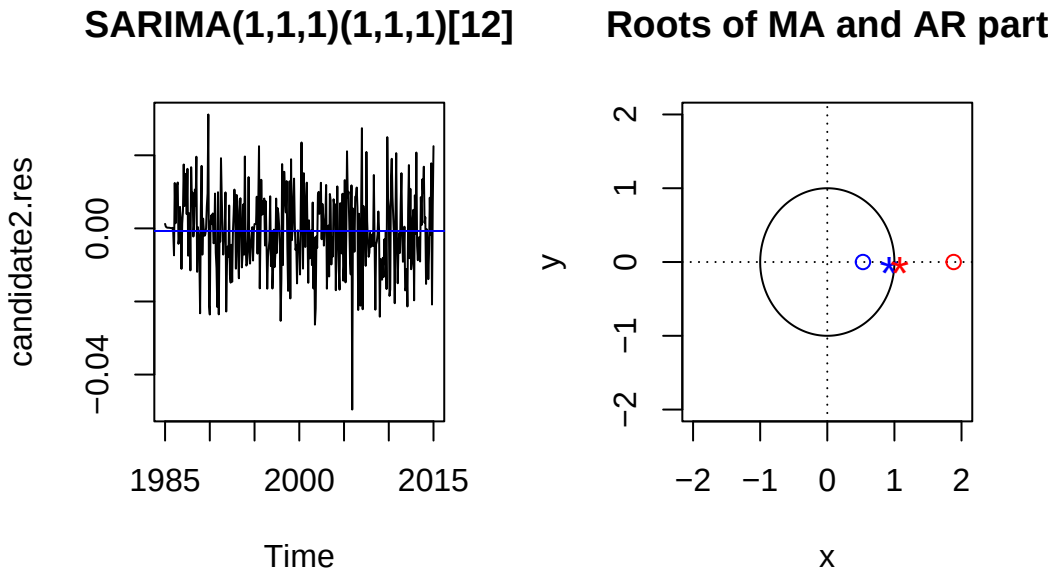


The ACF plot above shows that all the ACF of residuals are within the confidence interval, suggested that the residual is a $MA(0)$ process. One may notice that there's one little peak in the PACF plot, hence *Box-Pierce Test*, *Ljung-Box Test*, and *Mcleod-Li Test* are employed with the following result

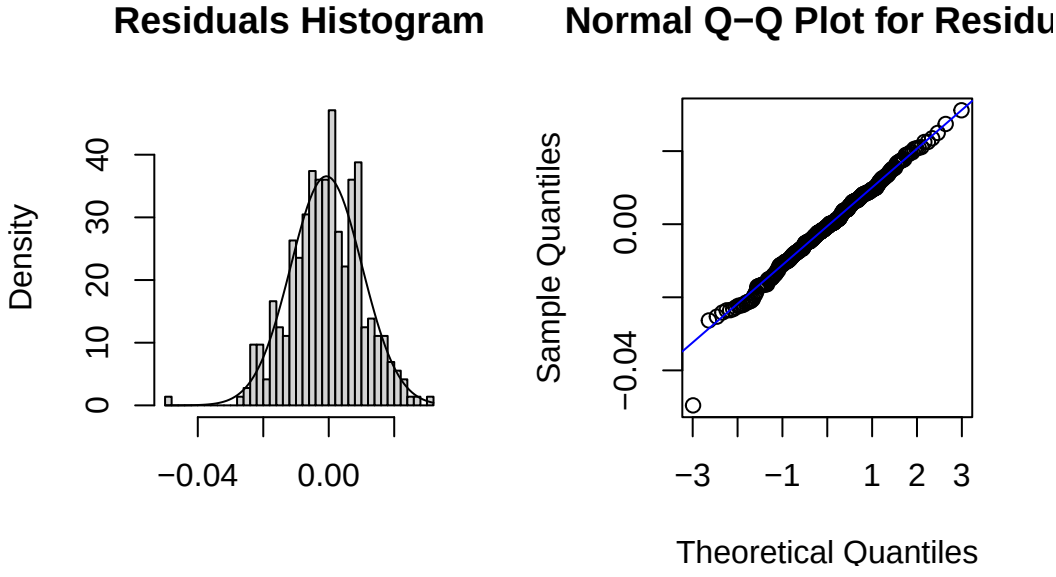
	Test Result
Shapiro-Wilk normality test	0.04671
Box-Pierce Test	0.696
Ljung-Box Test	0.6613
Mcleod-Li Test	0.8399

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 destribution 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.

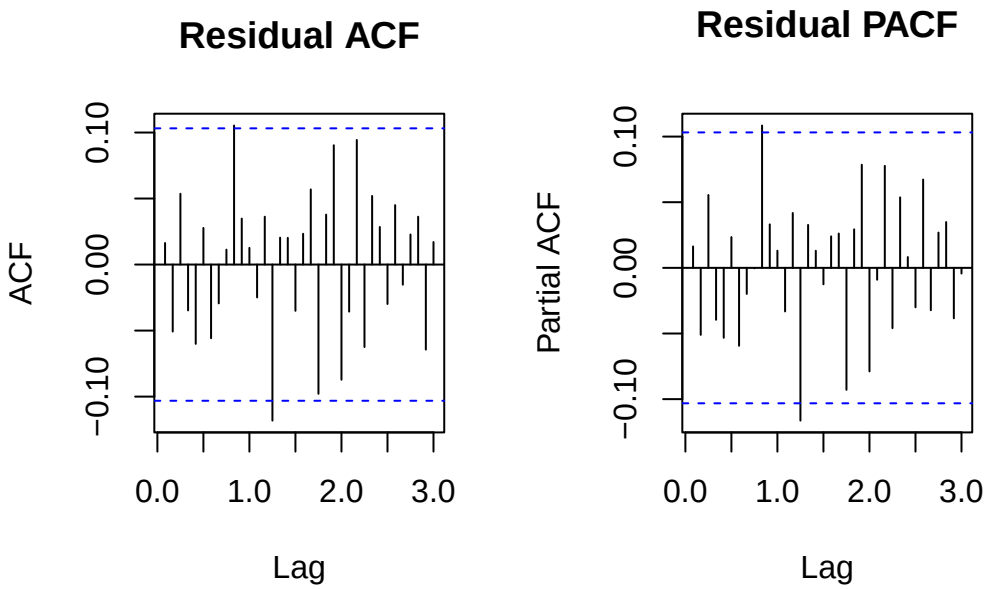
Model2: For model $SARIMA(1,1,1)(1,1,1)[12]$,



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.



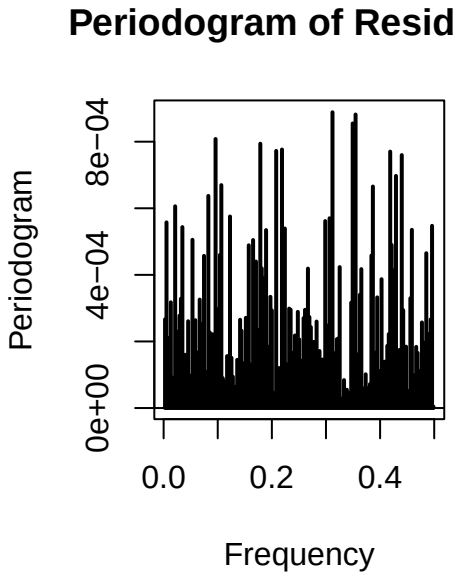
One may notice that there's one little peak in both ACF and PACF plot, hence *Box-Pierce Test*, *Ljung-Box Test*, and *Mcleod-Li Test* are employed with the following result

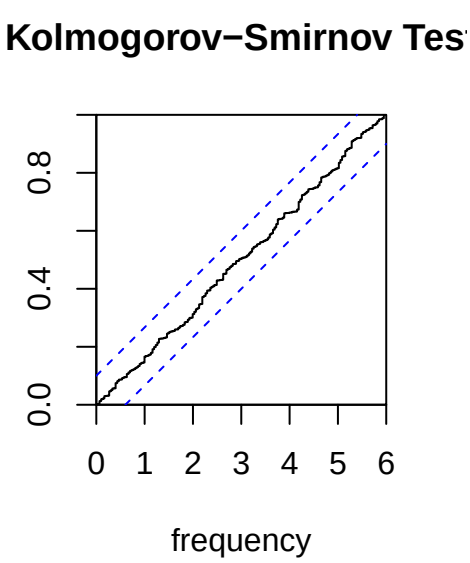
	Test Result
Shapiro-Wilk normality test	0.06738
Box-Pierce Test	0.3349
Ljung-Box Test	0.2996
Mcleod-Li Test	0.5893

Notice that the residual pass all the tests with a $p \geq 0.05$. 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.

Spectral Analysis

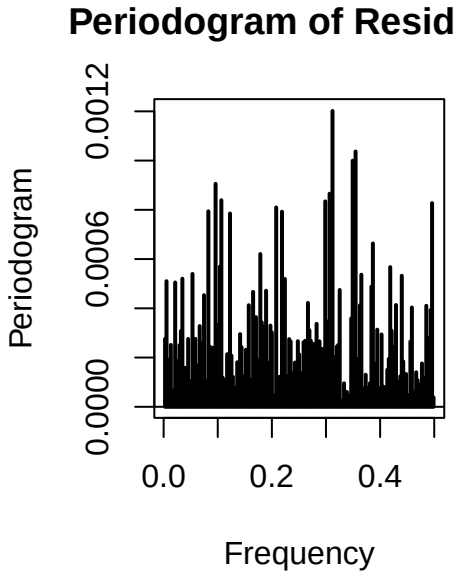
To further examine if the residual of model1 and model2 are White Noise, I employed *Periodogram*, *Kolmogorov-Smirnov Test*, and *Fisher's test*.

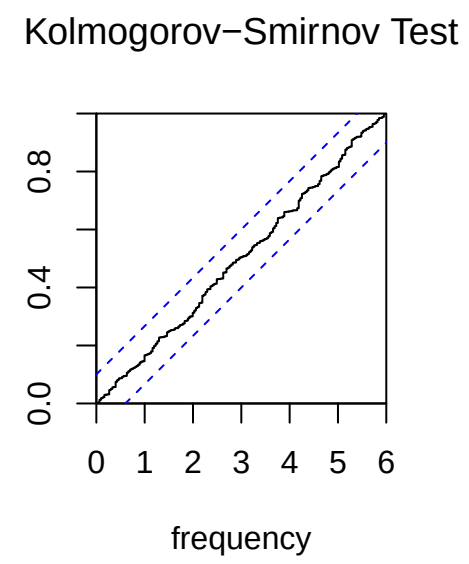




The left plot is the Periodogram of $SARIMA(5,1,2)(0,1,1)[12]$. Notice that there is no frequency dominates, which suggests that we have the White Noise. The right plot is the Kolmogorov-Smirnov Test where the function $C(x)$ never exits boundaries, which also suggested that there's no sufficient evidence 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 $P(\xi_q \geq x) = 0.9841 > \alpha = 0.05$, we accept the null hypothesis and hence we 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 $P(\xi_q \geq x) = 0.9285 > \alpha =$

0.05 from Fisher's test and we get the same conclusion that there's no sufficient evidence that the residual is statistically different from a Gaussian White Noise.

Final Selection

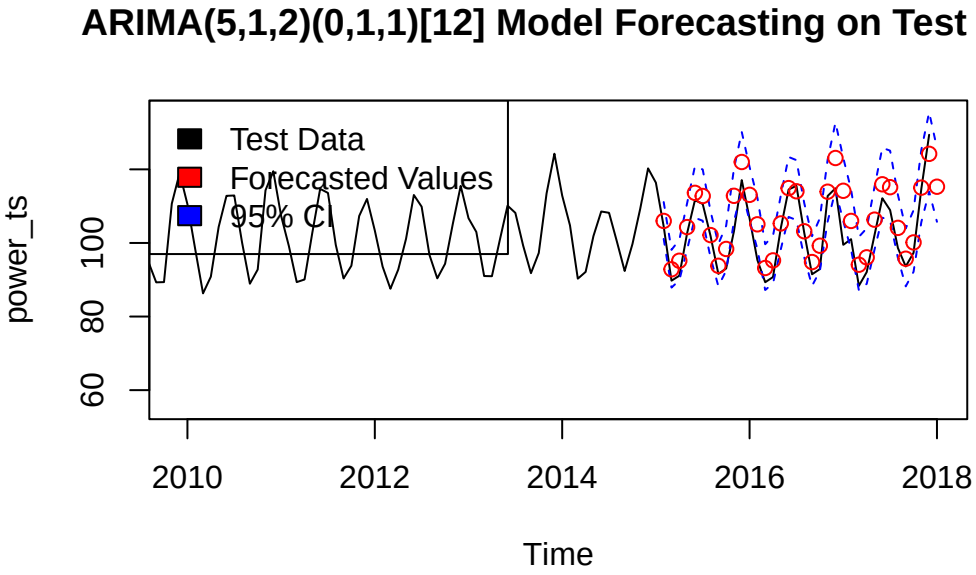
Since we've checked that the residuals of both models are White Noise and both models are stationary invertible models. By the principle of minimize AICC, model1 with a lower AICC wins. And thereby I select model1 with expression

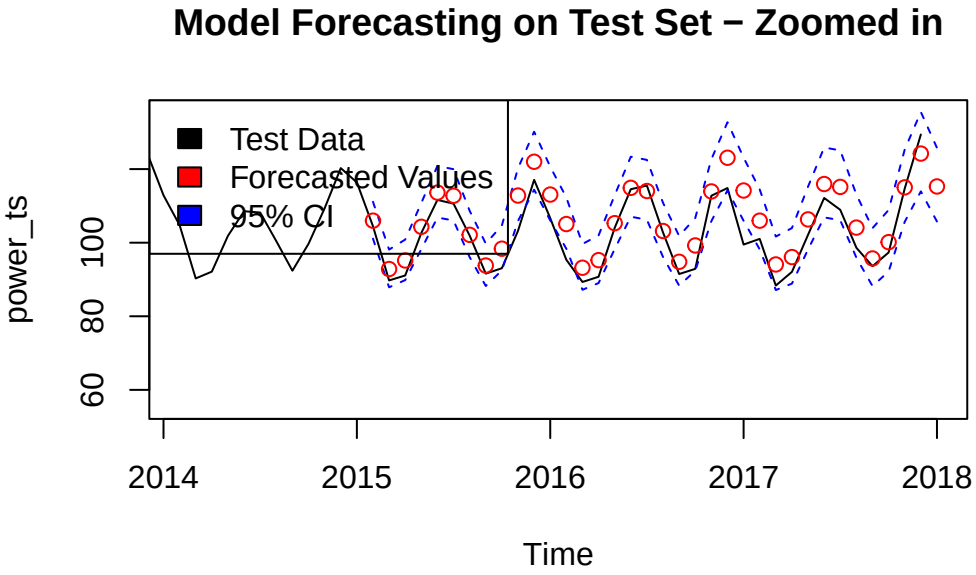
$$(1 + 0.3637B - 0.4015B^2 + 0.1405B^5)(1 - B)(1 - B^{12})X_t = (1 - 0.8095B^2)(1 - 0.8206B^{12})Z_t$$

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.

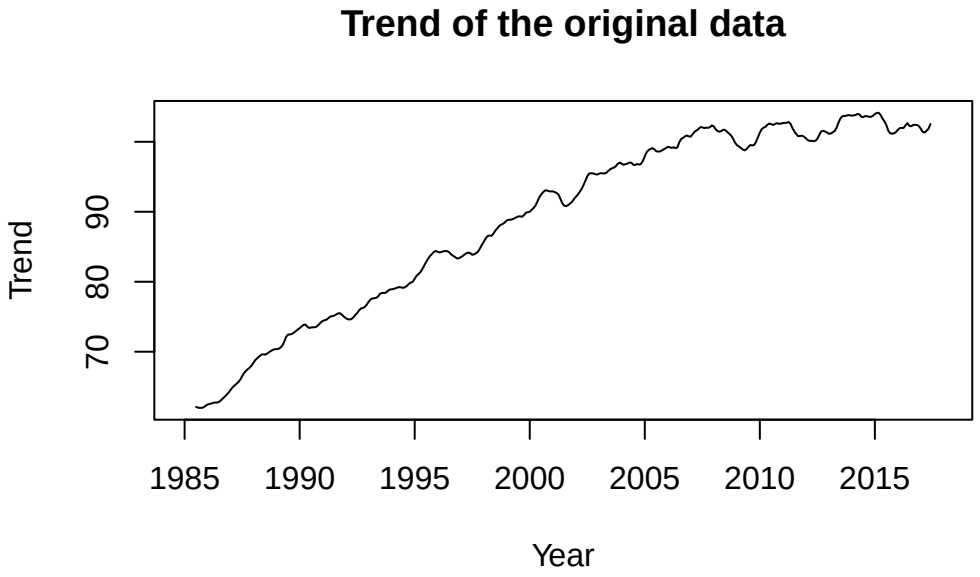




The first plot is the forecasted value using final model with the 95% confidence interval for the forecasted values with the real values in the testing set. The second plot is the zoom-in plots of the first plot so that we could see the details. From the plots, one could notice that

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

$$(1 + 0.3637B - 0.4015B^2 + 0.1405B^5)(1 - B)(1 - B^{12})X_t = (1 - 0.8095B^2)(1 - 0.8206B^{12})Z_t$$

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.

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

Feldmen, 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

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

Appendix

Code:

```
## 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/graph/fredgraph.csv?bgcolor=%23e1e9f0&chart_type=line&
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))
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))
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)

# Rolling variance
op <- par(mfrow=c(1,2))
# 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 = 'Before transf
```

```r
abline(fit, col="red")
abline(h=mean(rolling_variance, na.rm = TRUE), col="blue")
train_set.log=log(train_set)
rolling_variance_log <- rollapply(train_set.log, width = 12, FUN = var, align = 'right', f
plot(rolling_variance_log, type = 'l', xlab = 'Time', ylab = 'Variance', main = 'After tra
fit_log <- lm(rolling_variance_log~ time(rolling_variance_log))
par(op)

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

# Data Transformation
op <- par(mfrow=c(1,2))
lambda = bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
train_set.log.bc = (1/lambda)*(train_set.log^lambda-1)
plot(train_set.log.bc)
hist(train_set.log.bc, breaks = 6, xlab="", prob=TRUE,
     main = 'Time Series Histogram after Box-Cox Trans')

# Difference Variance with Different Lag
variances <- numeric(20)
variances[1] <- var(diff(train_set.log.bc))
# 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.log.bc), 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))
```

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

## Preliminary Model Identification ##
# ACF and PACF
op <- par(mfrow=c(1,2))
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))
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

# 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.log.bc, order=c(p, d_val, q), seasonal=c(P, D_val, Q), xr
        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(5,1,2)(0,1,2)[12]
refine_models[['SARIMA(5,1,2)(0,1,1)[12]']] <- arima(train_set.log.bc, order=c(5,1,2), sea
summary(refine_models[['SARIMA(5,1,2)(0,1,1)[12]']])
refine_models[['SARIMA(5,1,2)(0,1,1)[12]']] <- arima(train_set.log.bc, order=c(5,1,2), sea
summary(refine_models[['SARIMA(5,1,2)(0,1,1)[12]']])

# Model 2: SARIMA(2,1,1)(1,1,1)[12]
refine_models[['SARIMA(2,1,1)(1,1,1)[12]']] <- arima(train_set.log.bc, order=c(2,1,1), sea
summary(refine_models[['SARIMA(2,1,1)(1,1,1)[12]']])
refine_models[['SARIMA(1,1,1)(1,1,1)[12]']] <- arima(train_set.log.bc, order=c(1,1,1), sea
summary(refine_models[['SARIMA(1,1,1)(1,1,1)[12]']])
refine_models[['SARIMA(1,1,1)(0,1,1)[12]']] <- arima(train_set.log.bc, order=c(1,1,1), sea
summary(refine_models[['SARIMA(1,1,1)(0,1,1)[12]']])


## Diagnostic Checking ##
# candidate saver
candidates = c("SARIMA(5,1,2)(0,1,1)[12]","SARIMA(1,1,1)(1,1,1)[12]")
candidate_model <- list()
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))
```

```r
candidate1.res <- residuals(candidate1)
plot.ts(candidate1.res, main = 'SARIMA(5,1,2)(0,1,1)[12]')
abline(h=mean(candidate1.res), col="blue")
plot.roots(polyroot(c(1, -0.3637, 0.4015, 0, 0, -0.1406)),polyroot(c(1,0,-0.8095)), main =
par(op)

# Normal Distribution Check
op <- par(mfrow = c(1,2))
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))
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)
Box.test(candidate1.res, type=c("Box-Pierce"), lag = sqrt(length(train_set.log.bc)), fitdf
Box.test(candidate1.res, type=c("Ljung-Box"), lag = sqrt(length(train_set.log.bc)), fitdf
Box.test((candidate1.res)^2, type=c("Ljung-Box"), lag = sqrt(length(train_set.log.bc)), fi
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))
candidate2.res <- residuals(candidate2)
plot.ts(candidate2.res, main = 'SARIMA(1,1,1)(1,1,1)[12]')
abline(h=mean(candidate2.res), col="blue")
plot.roots(polyroot(c(1,-0.5312)),polyroot(c(1,-0.9232)), main = 'Roots of MA and AR part'
par(op)

# Normal Distribution Check
op <- par(mfrow = c(1,2))
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))
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.log.bc)), fitdf
Box.test(candidate2.res, type=c("Ljung-Box"), lag = sqrt(length(train_set.log.bc)), fitdf
Box.test((candidate2.res)^2, type=c("Ljung-Box"), lag = sqrt(length(train_set.log.bc)), fi
ar(candidate1.res, aic = TRUE, order.max = NULL, method = c("yule-walker"))

## Spectral Analysis
# Periodogram and Kolmogorov-Smirnov Test
op <- par(mfrow=c(1,2))
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))
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)
```

```r
## Forcasting ##
forecast(candidate1)
pred<- predict(candidate1, 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

# setting up forcast slice
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)

# invert transformation
orig_pred <- (pred$pred * lambda +1)^(1/lambda)
orig_pred <-exp(orig_pred)  # Convert predicted values back to original scale
orig_U <- (U * lambda +1)^(1/lambda)
orig_L <- (L * lambda +1)^(1/lambda)
orig_U <- exp(orig_U)  # Convert upper bound back to original scale
orig_L <- exp(orig_L)  # Convert lower bound back to original scale

ts.plot(power_ts, xlim=c(time(power_ts)[300], forecast_end), ylim = c(min(power_ts), max(o
main = 'ARIMA(5,1,2)(0,1,1)[12] Model Forecasting on Test Set')

# Plot
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
```