

Forecasting Australia Electricity Production in 1965

Final Project

Jiacong Wu

PSTAT 174

December 7, 2022

Contents

Abstract	3
Introduction	3
Data Analysis	3
Splitting Data	3
Ploting and Analysis	3
Transformation	4
Decomposition of transformed U_t	4
Differencing	5
Finding Parameters & Fit models	6
When using $\nabla_{12} \log(U_t)$, the ACF and PACF looks like:	6
.	7
When using $\nabla_{12} \log(U_t)$, the ACF and PACF looks like:	8
.	9
Diagnostics Checking	10
Checking stationarity and invertibility by checking the unit roots	10
Checking the normality of residuals	12
Portmanteau test	15
Forecasting	16
Conclusion	18
References	18
Appendix	19
All the code I used:	19

Abstract

Electricity has become increasingly important since it was ever invented. Human demand for electricity is also increasing along with its importance. This project focuses on electricity production in Australia from January 1956. This project aims to forecast the electricity production amount in 1965 and test whether the model performs well.

Introduction

In this project, I examined the monthly Australian electricity production(in million kilowatt hours) data from January 1956 from the `tsdl` package. First, I selected the Log transformation from 3 typical transformations to eliminate the change in variance. Then, I eliminated the seasonality by differencing at 12. I tried with and without differencing at lag 1. Next, I chose 2 of the best models to fit the data. Then, I performed diagnostic checkings of the models and selected the best model from the two. Finally, I used the best model, $SARIMA(2, 1, 1) \times (0, 1, 2)_{12}$, to forecast the tenth year's electricity production, and all the points are within the lines of 95% confidence interval.

Data Analysis

Splitting Data

Data are split into the training set and testing set, and the testing set is the last year of electricity production data that we want to forecast.

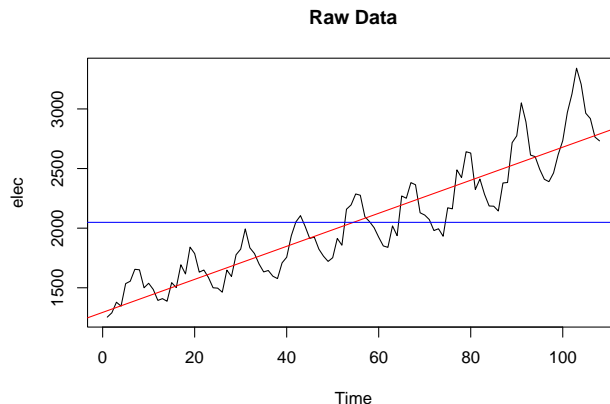
```
#dividing the training: first 9 years
elec= tsdl[[122]][1:108]
#setting the testing set: 10th year(1965)
test = ts(tsd1[[122]][109:120])
whole = tsdl[[122]][1:120]
```

Define: the monthly Australian electricity production in million kilowatt hours as

$$U_t, t = 1, 2, \dots, 120$$

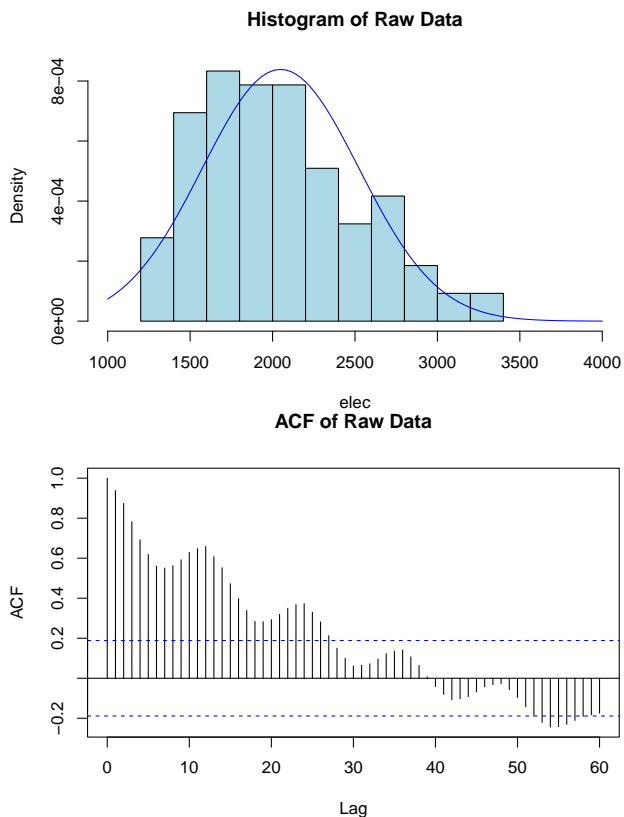
Plotting and Analysis

This is a plot of the original data:



From the graph, we can immediately observe that it is highly non-stationary. It has a trend, and there are repetitive patterns. Taking a closer look, it is not hard to find that the variance is smaller at the beginning and gets larger with time.

To further confirm the non-stationarity, here are histogram and ACF of the raw data:



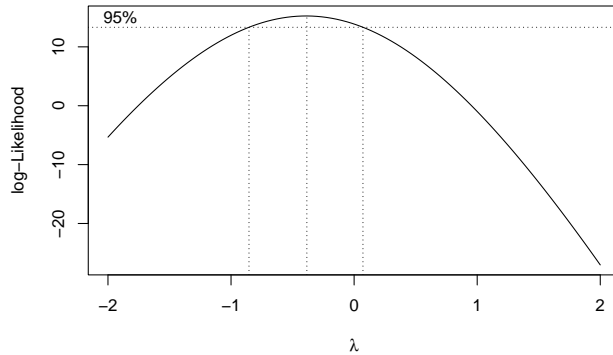
The histogram here is left-skewed. The ACF shows periodic patterns, which means there is strong seasonality. Also, the values in the ACF are still very large. As a result, it is confirmed that the raw data is not stationary.

Transformation

Since the variance change over time, the data needs to be transformed.

Box-Cox Transform can be used here with formula:

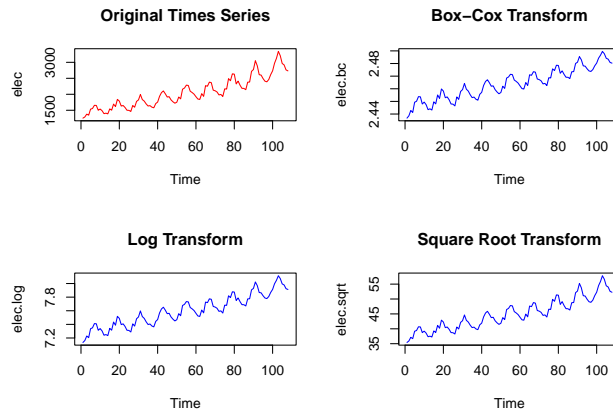
$$f_{\lambda}(U_t) = \begin{cases} \ln(U_t) & U_t > 0, \lambda = 0 \\ \lambda^{-1}(U_t^{\lambda} - 1) & U_t \geq 0, \lambda \neq 0 \end{cases}$$



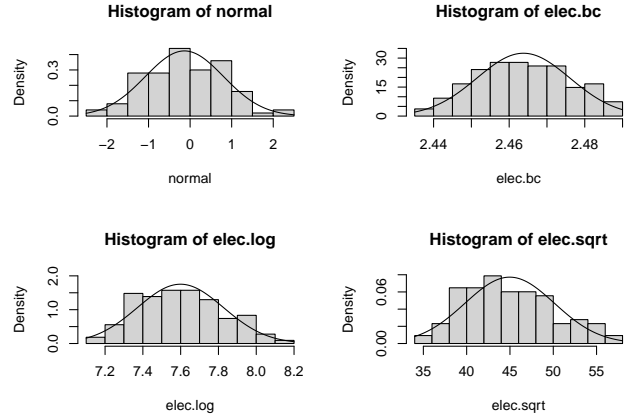
[1] -0.3838384

Using a Box-Cox transformation, the λ is calculated to be -0.3838384.

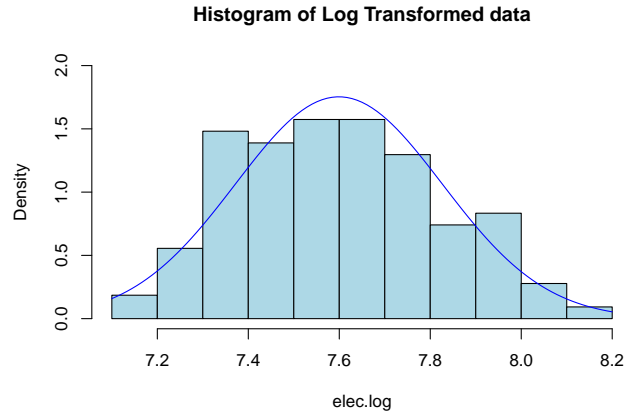
I also considered log transformation and power transformation(square root transformation here). Here are the graphs of transformed data comparing to the original data:



As the graph shows, the change in variance is eliminated in the transformations. However, it is hard to choose which transformation to use from the graphs, so I also plotted the histograms of each transformation.



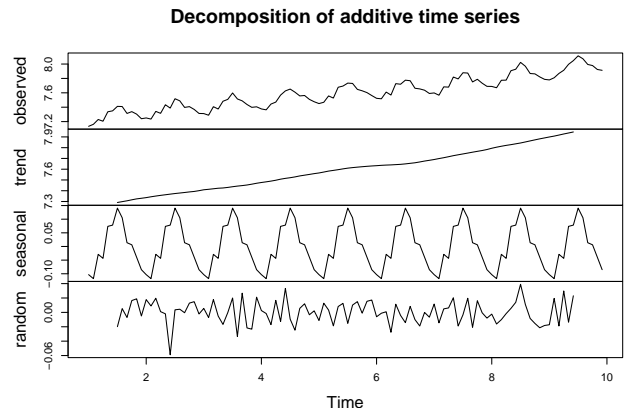
The log transformation looks the best to me, and in the plot of Box-Cox transformation, 0 is in the 95% confidence interval. Therefore, I chose to proceed with log transformation.



Here is a bigger graph of the transformation I chose. Log transformation gave a more symmetric histogram and more even variance. Transformed data is better normally distributed.

Decomposition of transformed U_t

Decomposition shows separated seasonality, linear trend and random white noise from the U_t



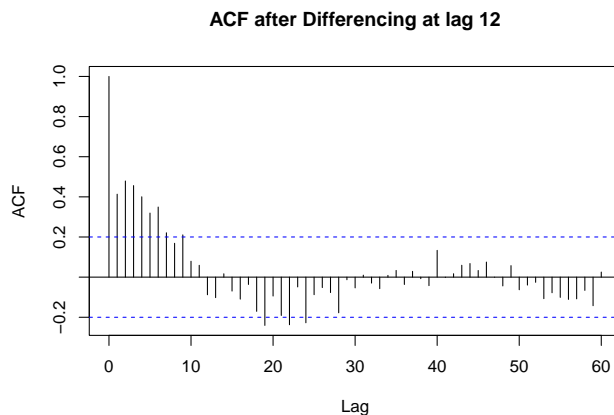
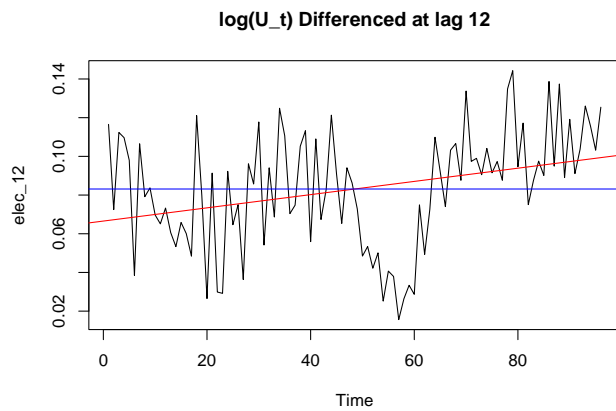
Differencing

A strong seasonality was noticed before, so I tried to difference at lag 12 to eliminate the seasonality. Making sure the differencing is helpful, variance is calculated before and after differencing at lag 12.

```
##      differenced_12 variance
## [1,] "No"              "0.0517724204680058"
## [2,] "Yes"             "0.000910931337207546"
```

It turns out that the variance did become lower after differencing at lag 12.

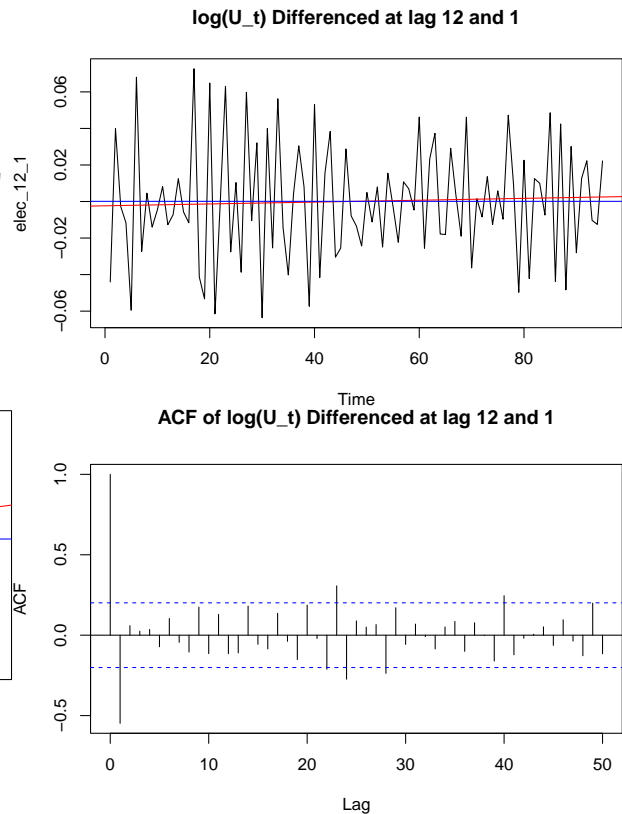
Here are two graphs after differencing at lag 12:



No periodic pattern in either of the two graphs indicates that differencing at lag 12 successfully removed the seasonality from the data. However, the red line in the first graph and the slow decay pattern in the acf show that potential non-stationarity exists.

Therefore, I tried to difference the data at lag 1 once. Variance are calculated and graphs are generated about differenced at lag 1 once after $\log(U_t)$ differenced at lag 12:

```
##      differenced_1 variance
## [1,] "No"              "0.000910931337207546"
## [2,] "Yes"             "0.00104937563670086"
```



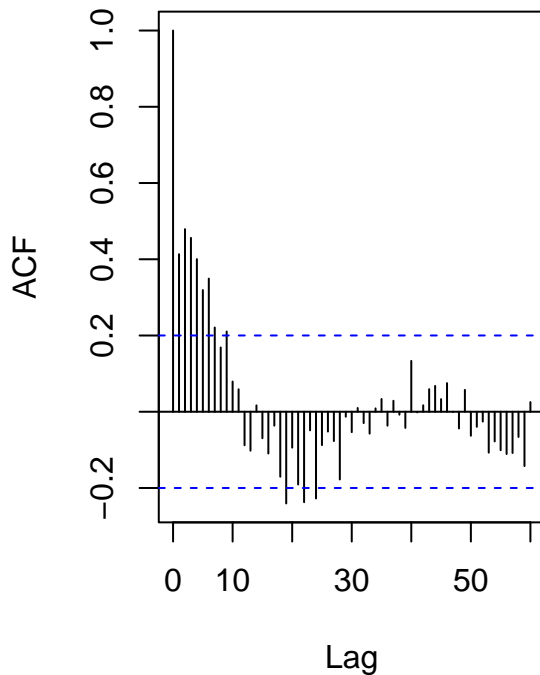
Unexpectedly, the variance became higher after difference at lag 1 once while ts plot and acf are looking better. Because differencing at lag 1 stabilized the trend and acf but increased the variance, I decided to keep both. When fitting models later, I would try $d = 0$ and $d = 1$ to see which one works better.

I will try both $\nabla_{12} \log(U_t)$ and $\nabla_1 \nabla_{12} \log(U_t)$ later.

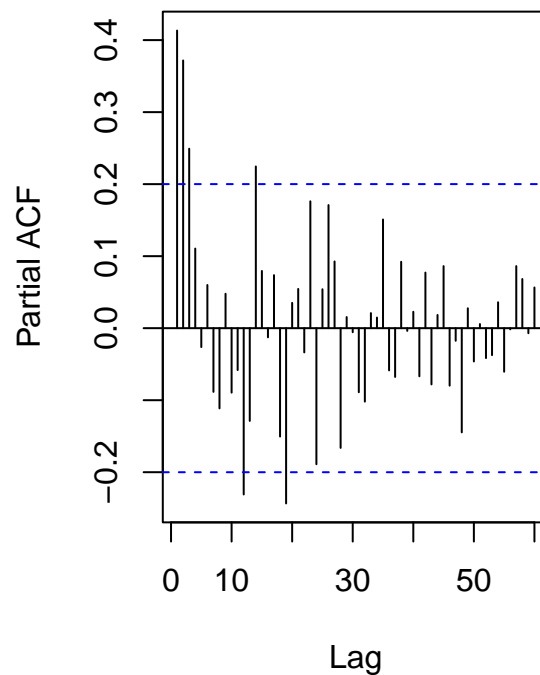
Finding Parameters & Fit models

When using $\nabla_{12} \log(U_t)$, the ACF and PACF looks like:

ACF of $\log(U_t)$ with $D = 12$



PACF of $\log(U_t)$ with $D = 12$



In ACF, lag 1, 2, 3, 4, 5, 6, maybe 7, maybe 9, 19, 22, 24 are outside the confidence interval. In PACF, lag = 1, 2, 3, 12, 14, 19 are outside the confidence interval. Based on the lags observed above, possible models are SARIMA for $\log(U_t)$:

$p = 3$ or 0 (0 if considered as tailing off)

$d = 0$ (because using data only differenced at lag 12 here)

$q = 6$ or 0 (0 if considered as tailing off)

$P = 1$

$D = 1$ (because differenced once at lag 12)

$Q = 2$

$s = 12$

```
##      model      aicc
## [1,] "mod0_0_0" "-332.884184835079"
## [2,] "mod0_0_6" "-380.945121196023"
## [3,] "mod3_0_0" "-446.224160290381"
## [4,] "mod3_0_6" "-445.599885243763"
```

When $p = 3$ and $q = 0$, the AICc are the lowest. Tried models with $p = 0$, but it increased AICc.

```
model_A = arima(elec.log, order = c(3,0,0), seasonal = list(order = c(1,1,2),period = 12),
                method = "ML")
```

```
model_A
```

```
##
```

```
## Call:
## arima(x = elec.log, order = c(3, 0, 0), seasonal = list(order = c(1, 1, 2),
##   period = 12), method = "ML")
##
## Coefficients:
##          ar1      ar2      ar3      sar1      sma1      sma2
##      0.2659  0.3905  0.3435 -0.5970  0.0495 -0.9181
## s.e.  0.1040  0.0951  0.1014   0.1487  0.7256   0.6322
##
## sigma^2 estimated as 0.0003455:  log likelihood = 230.53,  aic = -447.06
```

Since the coefficient for sma1 is 0.0495 with a standard error of 0.7256, the coefficient for sma1 is likely to be 0. So I fixed the coefficient sma1 to be 0.

```
model_A_1 = arima(elec.log, order = c(3,0,0), seasonal = list(order = c(1,1,2),period = 12),
                  fixed = c(NA,NA,NA,NA,0,NA),
                  method = "ML")
model_A_1
```

```
##
## Call:
## arima(x = elec.log, order = c(3, 0, 0), seasonal = list(order = c(1, 1, 2),
##   period = 12), fixed = c(NA, NA, NA, NA, 0, NA), method = "ML")
##
## Coefficients:
##          ar1      ar2      ar3      sar1  sma1      sma2
##      0.2673  0.3904  0.3423 -0.5876     0 -0.9031
## s.e.  0.1002  0.0952  0.0982   0.1308     0  0.4753
##
## sigma^2 estimated as 0.0003498:  log likelihood = 230.53,  aic = -449.05
```

Compared model with and without fix sma1 coefficient:

```
##      fixed_0 aicc
## [1,] "No"      "-446.224160290381"
## [2,] "Yes"     "-448.218637282883"
```

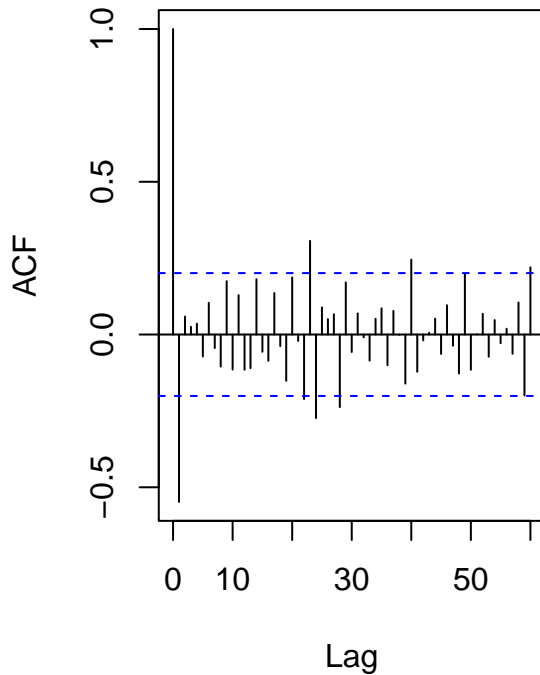
With sma1 fixed to zero, the AICc is even lower. So proceed with the model with lower AICc.

So Model A is $SARIMA(3,0,0) \times (1,1,2)_{12}$, specifically:

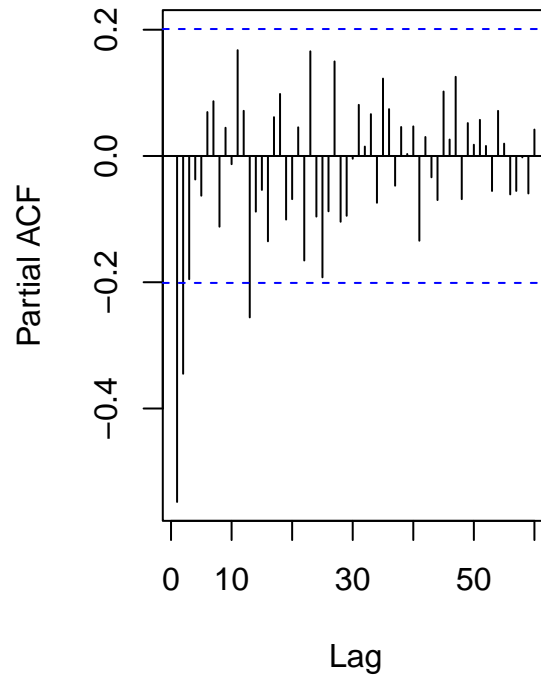
$$\begin{aligned} \nabla_{12} \log(U_t) : \\ (1 - 0.26_{(0.1002)}B - 0.3904_{(0.0952)}B^2 - 0.3423_{(0.0982)}B^3)(1 + 0.5876_{(0.1308)}B^{12})(1 - B^{12})X_t \\ = (1 - 0.9031_{(0.4753)}B^{12})Z_t \end{aligned}$$

When using $\nabla_{12} \log(U_t)$, the ACF and PACF looks like:

ACF of bc(U_t) at lag 12 and 1



PACF of bc(U_t) at lag 12 and 1



In ACF, lag 1, 23, 24 are outside confidence interval In PACF, lag 1, 2, 13 are outside the confidence interval. So possible SARIMA models are:

$p = 2$

$d = 1$ (considering data differenced at lag 1)

$q = 1$

$P = 0$

$D = 1$ (differenced once at lag 12)

$Q = 2$ or 0 (considering lag 24 may not be significant)

$s = 12$

```
##      model      aicc
## [1,] "mod_Q_2" "-444.413641600271"
## [2,] "mod_Q_0" "-425.026523346142"
```

It turns out that lag 24 in the acf is important, as the AICc with $Q = 2$ is lower.

```
model_B = arima(elec.log, order = c(2,1,1), seasonal = list(order = c(0,1,2),period = 12),
                 method = "ML")
model_B
```

```
##
## Call:
## arima(x = elec.log, order = c(2, 1, 1), seasonal = list(order = c(0, 1, 2),
```



```
##      period = 12), method = "ML")
```

```
##
```

```
## Coefficients:
```

```
##          ar1      ar2      ma1      sma1      sma2
```

```
##      -0.3666 -0.1207 -0.4110 -0.6351 -0.3643
```

```
## s.e.   0.3278   0.2246   0.3222   0.2657   0.1486
```

```
##
```

```
## sigma^2 estimated as 0.000379:  log likelihood = 228.5,  aic = -445
```

Since the coefficient for ar2 is -0.1207 with standard error of 0.22, and 0 is in the standard error, I tried to fix the ar2 coefficient.

```
model_B_1 = arima(elec.log, order = c(2,1,1), seasonal = list(order = c(0,1,2),period = 12),
                  fixed = c(NA,0,NA,NA,NA), method = "ML")
```

```
model_B_1
```

```
##
```

```
## Call:
```

```
## arima(x = elec.log, order = c(2, 1, 1), seasonal = list(order = c(0, 1, 2),
```

```
##      period = 12), fixed = c(NA, 0, NA, NA, NA), method = "ML")
```

```
##
```

```
## Coefficients:
```

```
##          ar1  ar2      ma1      sma1      sma2
```

```
##      -0.2204    0 -0.5526 -0.6288 -0.3711
```

```
## s.e.   0.1480    0  0.1253  0.2391  0.1407
```

```
##
```

```
## sigma^2 estimated as 0.0003802:  log likelihood = 228.36,  aic = -446.72
```

Compared model with and without fix sma1 coefficient:

```
##      fixed_0 aicc
```

```
## [1,] "No"      "-444.413641600271"
```

```
## [2,] "Yes"     "-446.128934115444"
```

The AICc is lower with the model fixed ar2, so proceed with the model fixed ar2 coefficient.

So Model B is $SARIMA(2, 1, 1) \times (0, 1, 2)_{12}$, specifically:

$$\nabla_1 \nabla_{12} \log(U_t) :$$

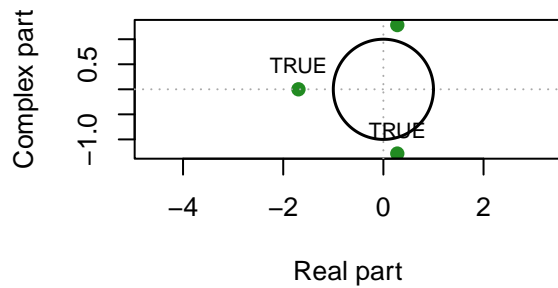
$$(1 + 0.2204_{(0.1480)}B)(1 - B)X_t = (1 - 0.6288_{(0.2391)}B^{12} - 0.3711_{(0.1407)}B^{24})Z_t$$

Diagnostics Checking

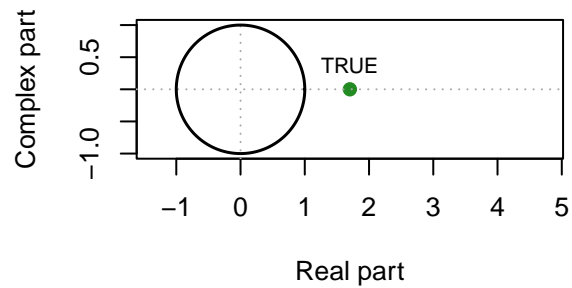
Checking stationarity and invertibility by checking the unit roots

Model A

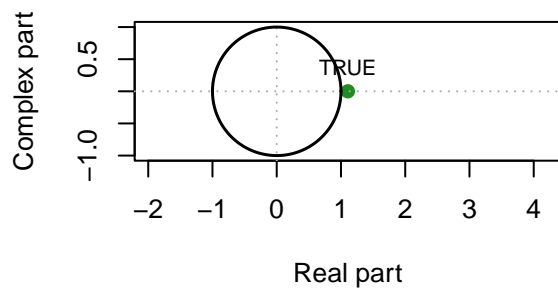
Roots outside the Unit Circle?



Roots outside the Unit Circle?

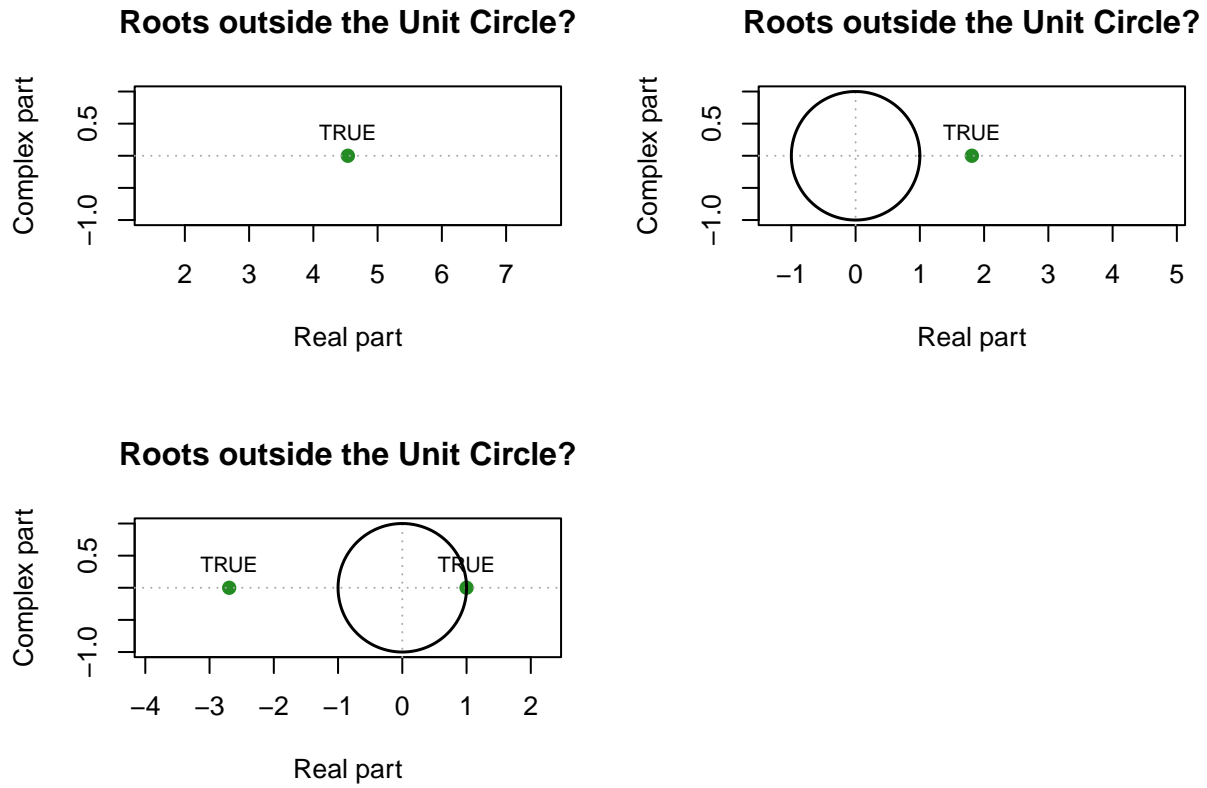


Roots outside the Unit Circle?



Model A is stationary because all roots for ϕ and Φ in the AR/SAR part are outside of the unit circle. Model A is also invertible because all roots for Θ in the MA part are outside the unit circle.

Model B



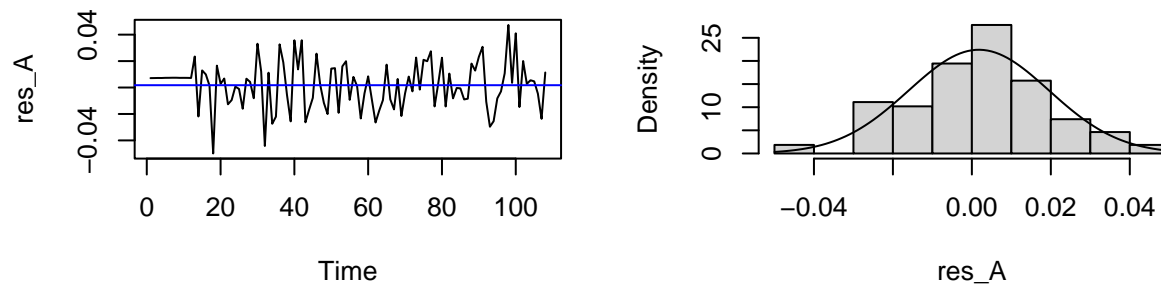
Model B is stationary because all roots for ϕ in the AR part are outside of the unit circle. Model A is also invertible because all roots for θ and Θ in the MA/SMA part are outside the unit circle.

Checking the normality of residuals

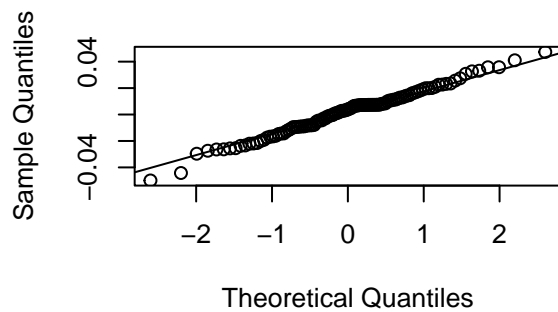
Model A

```
## [1] 0.001772426
##
## Shapiro-Wilk normality test
##
## data:  res_A
## W = 0.99116, p-value = 0.7117
```

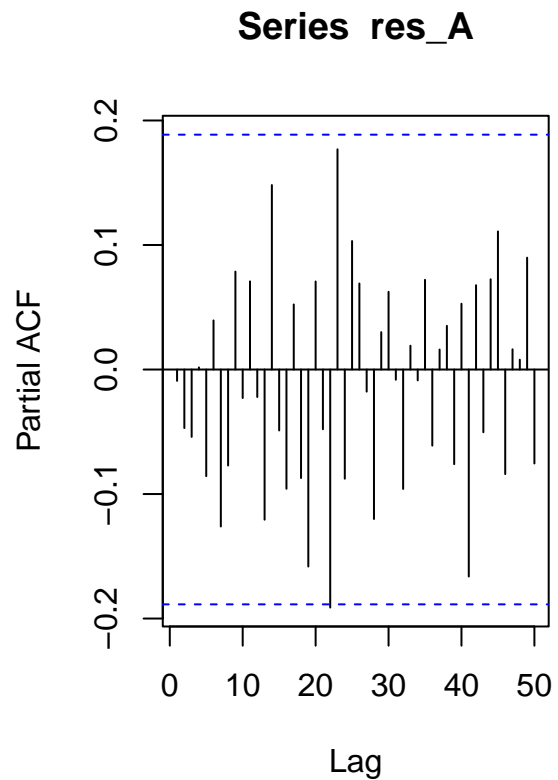
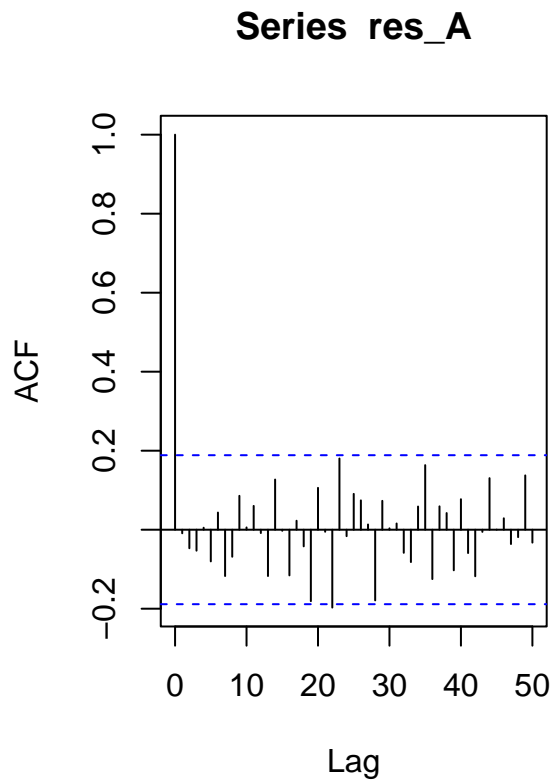
Histogram of res_A



Normal Q-Q Plot



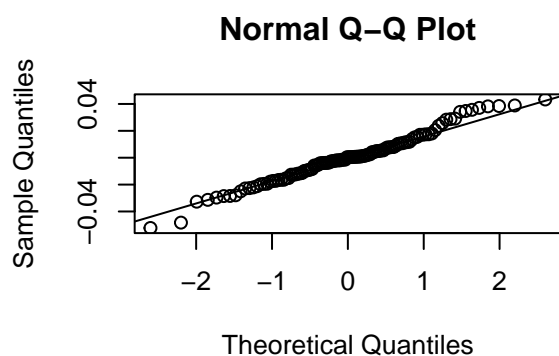
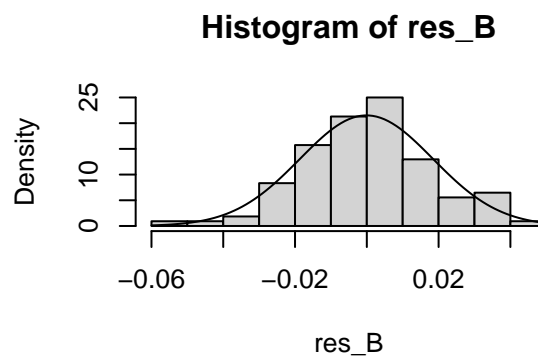
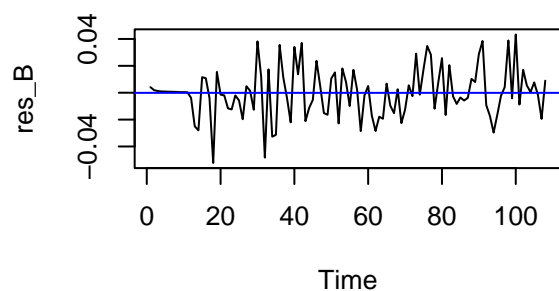
There is no trend, visible change of variance, or seasonality in the plot. The histogram is nearly normal, and the qqplot is a straight line. The sample mean is 0.001772426 which is almost zero. The p-value of Shapiro-Wilk normality test is 0.7117, greater than 0.05, so Model A passed Shapiro-Wilk normality test.



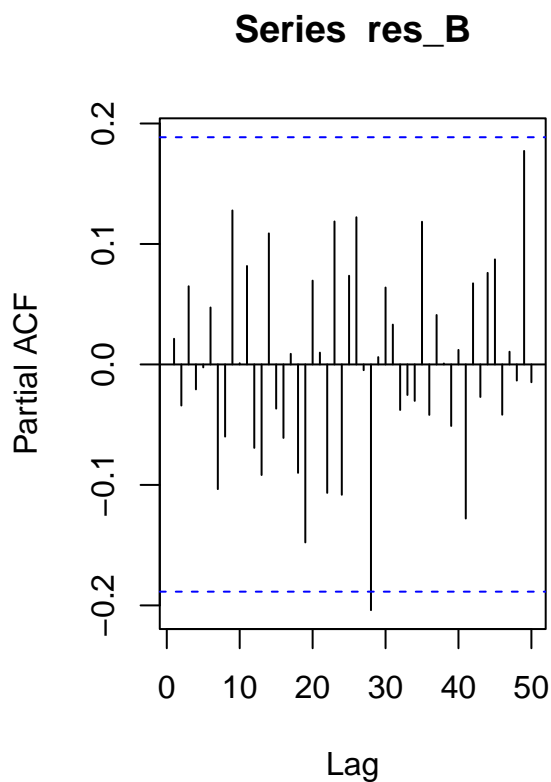
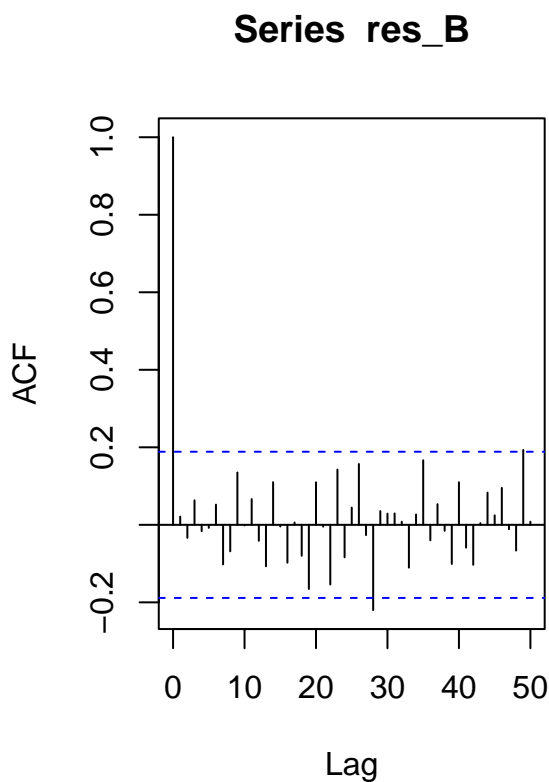
All the acf and pacf of residuals are within confidence intervals. Even if lag 22 is a little out of confidence interval, it can still be count as within the confidence interval. These can all be counted as zeros.

Model B

```
## [1] -9.864481e-05
##
## Shapiro-Wilk normality test
##
## data:  res_B
## W = 0.98335, p-value = 0.1968
```



There is no trend, visible change of variance, or seasonality in the plot. The histogram is nearly normal, and the qqplot is almost a straight line. The sample mean is $-9.864481e-05$, which is very close to zero. The p-value of Shapiro-Wilk normality test is 0.1968, greater than 0.05, so Model A passed Shapiro-Wilk normality test.



All acf and pacf of residuals are within confidence intervals or can be counted as in the confidence intervals. These can all be counted as zeros.

Portmanteau test

Model A

```
##
## Box-Pierce test
##
## data: res_A
## X-squared = 4.2445, df = 5, p-value = 0.5148
##
## Box-Ljung test
##
## data: res_A
## X-squared = 4.6035, df = 5, p-value = 0.4662
##
## Box-Ljung test
##
## data: res_A^2
## X-squared = 9.5964, df = 10, p-value = 0.4766
```

All the p-value are greater than 0.05, so model A passed all the Portmanteau tests.

```
##
## Call:
## ar(x = res_A, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0 sigma^2 estimated as 0.0003166
```

Fitted the residuals of Model A to AR(0), the residuals is white noise.

Model A passed all diagnostic checking.

Model B

```
##
## Box-Pierce test
##
## data: res_B
## X-squared = 4.5327, df = 6, p-value = 0.605
##
## Box-Ljung test
##
## data: res_B
## X-squared = 4.954, df = 6, p-value = 0.5497
##
## Box-Ljung test
##
## data: res_B^2
## X-squared = 8.0382, df = 10, p-value = 0.6251
```

All the p-value are greater than 0.05, so model B passed all the Portmanteau tests.

```
##
## Call:
## ar(x = res_B, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0   sigma^2 estimated as  0.0003438
```

Fitted the residuals of Model B to AR(0), the residuals is white noise.

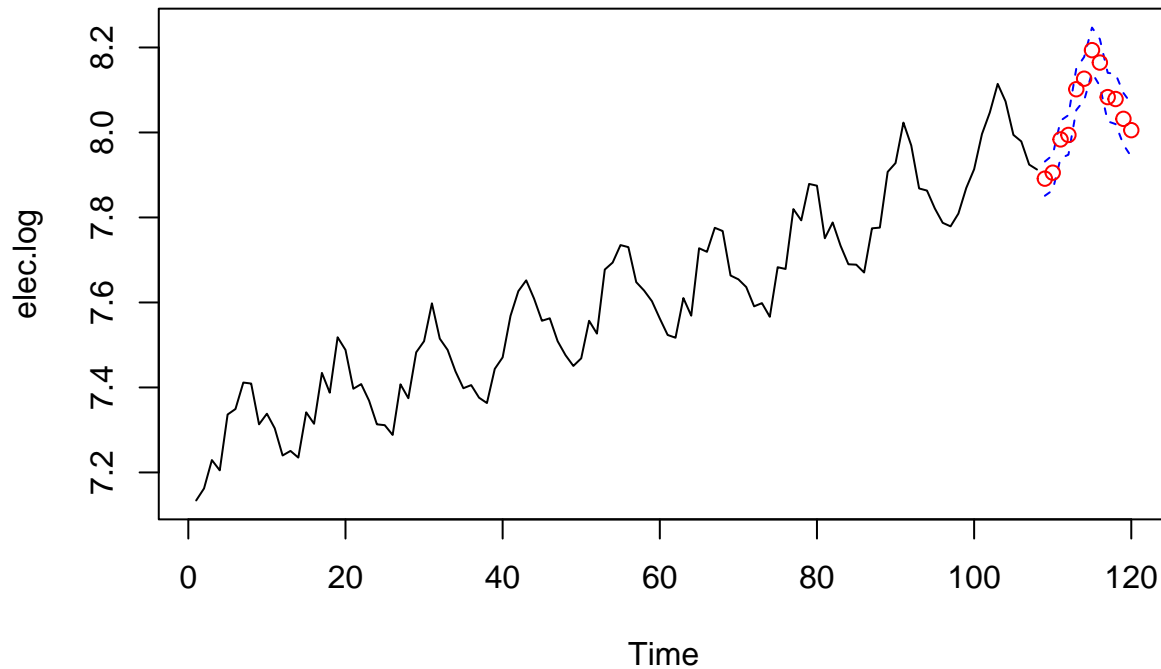
Model B passed all diagnostic checking.

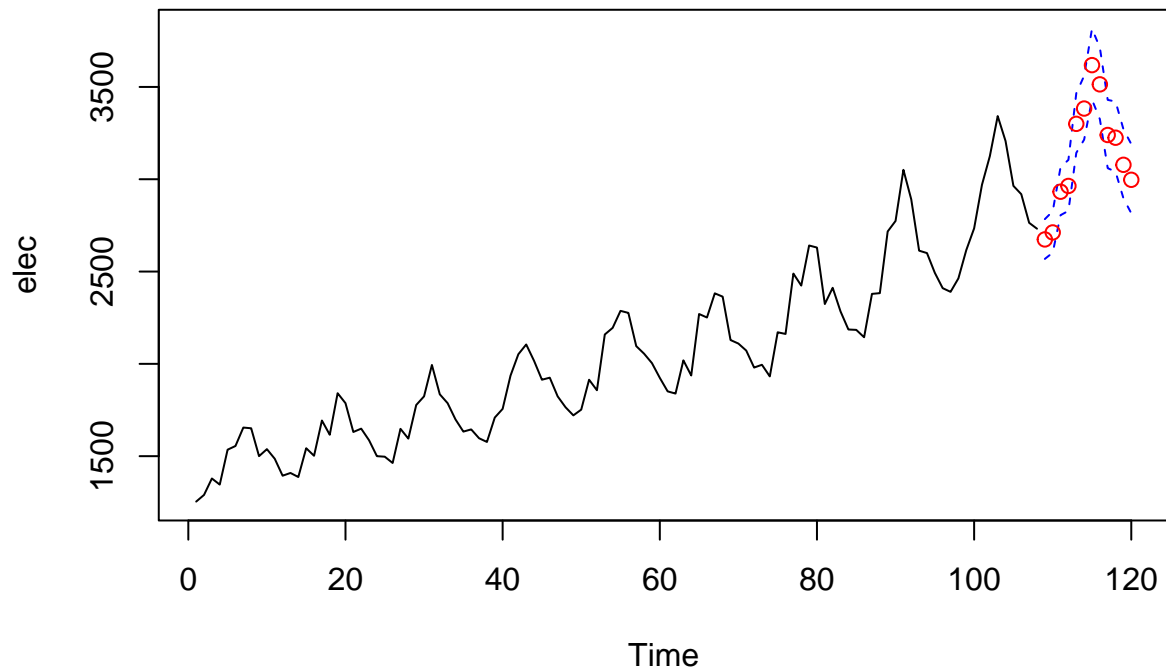
Model A $SARIMA(3,0,0) \times (1,1,2)_{12}$ **and Model B** $SARIMA(2,1,1) \times (0,1,2)_{12}$ **both passed all the diagnostic checking.**

Using the principle of parsimony, choose $SARIMA(2,1,1) \times (0,1,2)_{12}$ as the final model:

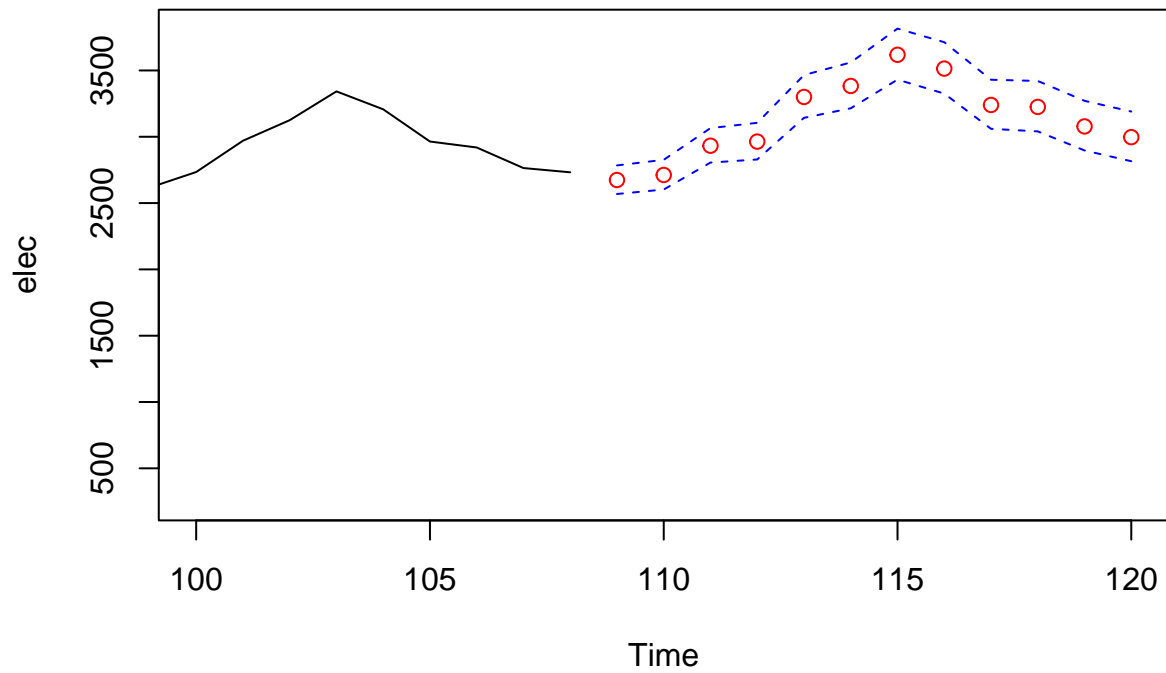
$$(1 + 0.2204_{(0.1480)}B)(1 - B)X_t = (1 - 0.6288_{(0.2391)}B^{12} - 0.3711_{(0.1407)}B^{24})Z_t$$

Forecasting

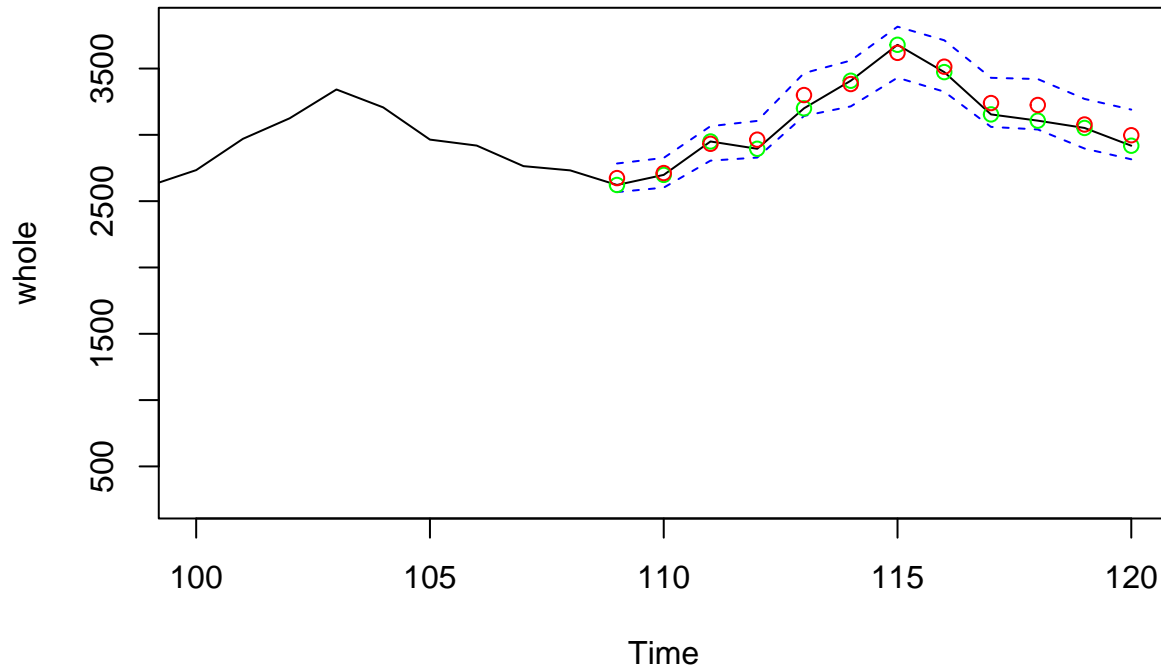




Zoomed in graphs of the forecast:



Here is a comparison of the points I predicted (in red) and the true data points (in green).



Conclusion

This project aims to predict the electricity production in Austria in 1965, using data from Jan 1956. The final time series model I picked was $SARIMA(2, 1, 1) \times (0, 1, 2)_{12}$, specifically:

$$(1 + 0.2204_{(0.1480)}B)(1 - B)X_t = (1 - 0.6288_{(0.2391)}B^{12} - 0.3711_{(0.1407)}B^{24})Z_t$$

It performed pretty well in forecasting. All the data points I predicted are within the 95% confidence interval and located not far from the true testing data points.

References

Lecture Notes and Slices

tsdl library from <http://cran.us.r-project.org>

Appendix

All the code I used:

```
#loading libraries used
library(tsd1)
library(tidyverse)
library(MASS)
library(asts)
library(qpcR)
library(UnitCircle)
library(forecast)

#dividing the training: first 9 years
elec= tsdl[[122]][1:108]
#setting the testing set: 10th year(1965)
test = ts(tsd1[[122]][109:120])
whole = tsdl[[122]][1:120]

#ploting the original data
#ploting the time series of electricity production to see the pattern
plot.ts(elec, main = "Raw Data")
fit = lm(elec ~ as.numeric(1:length(elec)))
abline(fit, col = "red")
abline(h = mean(elec), col = "blue")

#histogram and acf of original data

hist(elec, freq = FALSE, xlim = range(1000,4000),
      main = "Histogram of Raw Data", col = "light blue")
curve(dnorm(x,mean(elec), sqrt(var(elec))), add = TRUE, col = "blue")
acf(elec, lag.max = 60, main = "ACF of Raw Data")

#Transformations
t = 1:length(elec)
fit = lm(elec ~ t)
bcTransform = boxcox(elec ~ t,plotit = TRUE)
lambda = bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
lambda
elec.bc = (1/lambda)*(elec^lambda-1)

elec.log = log(elec)
elec.sqrt = sqrt(elec)
op= par(mfrow=c(2,2))
ts.plot(elec, main = "Original Times Series", col = "red")
ts.plot(elec.bc, main = "Box-Cox Transform",col = "blue")
ts.plot(elec.log, main = "Log Transform",col = "blue")
ts.plot(elec.sqrt, main = "Square Root Transform",col = "blue")

set.seed(10)
normal = c(rnorm(100))
op= par(mfrow=c(2,2))
hist(normal, freq = FALSE)
curve(dnorm(x,mean(normal), sqrt(var(normal))), add = TRUE)
```

```

hist(elec.bc, freq = FALSE, ylim = c(0,35))
curve(dnorm(x,mean(elec.bc), sqrt(var(elec.bc))), add = TRUE)
hist(elec.log, freq = FALSE, ylim = c(0,2))
curve(dnorm(x,mean(elec.log), sqrt(var(elec.log))), add = TRUE)
hist(elec.sqrt, freq = FALSE)
curve(dnorm(x,mean(elec.sqrt), sqrt(var(elec.sqrt))), add = TRUE)

hist(elec.log, freq = FALSE, main = "Histogram of Log Transformed data",
     ylim = c(0,2), col = "light blue")
curve(dnorm(x, mean(elec.log), sqrt(var(elec.log))),
     add = TRUE, col = "blue")

#plot decomposition

y = ts(as.ts(elec.log), frequency = 12)
decomp = decompose(y)
plot(decomp)

#differencing and comparing variances

elec_12 = diff(elec.log, 12)
differenced_12 = c("No", "Yes")
variance = c(var(elec.log), var(elec_12))
cbind(differenced_12, variance)

# bc stands for box-cox transform
elec_12, main = "log(U_t) Differenced at lag 12")
fit = lm(elec_12 ~ as.numeric(1:length(elec_12)))
abline(fit, col = "red")
abline(h = mean(elec_12), col = "blue")

acf(elec_12, lag.max = 60, main = "ACF after Differencing at lag 12")

#comparing variance after differencing at 1

elec_12_1 = diff(elec_12, 1)
variance = c(var(elec_12), var(elec_12_1))
differenced_1 = c("No", "Yes")
cbind(differenced_1, variance)

plot.ts(elec_12_1, main = "log(U_t) Differenced at lag 12 and 1" )
fit = lm(elec_12_1 ~ as.numeric(1:length(elec_12_1)))
abline(fit, col = "red")
abline(h = mean(elec_12_1), col = "blue")
acf(elec_12_1, lag.max = 50,
    main = "ACF of log(U_t) Differenced at lag 12 and 1")

# Finding p and q, model selection

op= par(mfrow=c(1,2))
acf(elec_12, main = "ACF of log(U_t) with D = 12", lag.max = 60)
pacf(elec_12, main = "PACF of log(U_t) with D = 12", lag.max = 60)

```

```

#fitting possible models

mod0_0_0 <- arima(elec.log, order = c(0,0,0), seasonal =
  list(order = c(1,1,2),period = 12), method = "ML")
mod0_0_6 <- arima(elec.log, order = c(0,0,6), seasonal =
  list(order = c(1,1,2),period = 12), method = "ML")
mod3_0_0 <- arima(elec.log, order = c(3,0,0), seasonal =
  list(order = c(1,1,2),period = 12), method = "ML")
mod3_0_6 <- arima(elec.log, order = c(3,0,6), seasonal =
  list(order = c(1,1,2),period = 12), method = "ML")

aicc = c(AICc(mod0_0_0),AICc(mod0_0_6),AICc(mod3_0_0),AICc(mod3_0_6))
model = c("mod0_0_0","mod0_0_6","mod3_0_0","mod3_0_6")
cbind(model,aicc)

#choose model A

model_A = arima(elec.log, order = c(3,0,0), seasonal =
  list(order = c(1,1,2),period = 12),
  method = "ML")
model_A

# improve model A by fix 0

model_A_1 = arima(elec.log, order = c(3,0,0), seasonal =
  list(order = c(1,1,2),period = 12),
  fixed = c(NA,NA,NA,NA,0,NA),
  method = "ML")
model_A_1

aicc = c(AICc(model_A),AICc(model_A_1))
fixed_0 = c("No", "Yes")
cbind(fixed_0, aicc)

# selecting model B

op= par(mfrow=c(1,2))
acf(elec_12_1, main = "ACF of bc(U_t) at lag 12 and 1",l
  ag.max = 60)
pacf(elec_12_1, main = "PACF of bc(U_t) at lag 12 and 1",
  lag.max = 60)

#fitting possible models

mod_Q_2 <- arima(elec.log, order = c(2,1,1), seasonal =
  list(order = c(0,1,2),period = 12), method = "ML")
mod_Q_0 <- arima(elec.log, order = c(2,1,1), seasonal =
  list(order = c(0,1,0),period = 12), method = "ML")

aicc = c(AICc(mod_Q_2),AICc(mod_Q_0))
model = c("mod_Q_2","mod_Q_0")
cbind(model,aicc)

```

```

model_B = arima(elec.log, order = c(2,1,1), seasonal =
                list(order = c(0,1,2),period = 12),
                method = "ML")
model_B

#improving modle B by fix 0

model_B_1 = arima(elec.log, order = c(2,1,1), seasonal =
                  list(order = c(0,1,2),period = 12),
                  fixed = c(NA,0,NA,NA,NA), method = "ML")
model_B_1

aicc = c(AICc(model_B),AICc(model_B_1))
fixed_0 = c("No", "Yes")
cbind(fixed_0, aicc)

#Model Diagnostics

#checking unit roots for model A
op= par(mfrow=c(2,2))
uc.check(pol_ = c(1, 0.2673,0.3904,0.3423),
         plot_output = TRUE,print_output = F)#AR
uc.check(pol_ = c(1,-0.5876),
         plot_output = TRUE,print_output = F)#seasonal AR
uc.check(pol_ = c(1,-0.9031),
         plot_output = TRUE,print_output = F)#seasonal MA

#checking unit roots for model B
op= par(mfrow=c(2,2))
uc.check(pol_ = c(1, -0.2204),
         plot_output = TRUE,print_output = F)#AR
uc.check(pol_ = c(1,-0.5526),
         plot_output = TRUE,print_output = F)#MA
uc.check(pol_ = c(1,-0.6288, -0.3711),
         plot_output = TRUE,print_output = F)#seasonal MA

#check residuals for model A
res_A = residuals(model_A_1)
op= par(mfrow=c(2,2))
plot.ts(res_A)
abline(h = mean(res_A), col = "blue")
hist(res_A, freq = FALSE)
curve(dnorm(x,mean(res_A), sqrt(var(res_A))), add = TRUE)
qqnorm(res_A)
qqline(res_A)
mean(res_A)
shapiro.test(res_A)

op= par(mfrow=c(1,2))
acf(res_A, lag.max = 50)
pacf(res_A, lag.max = 50)

#check residuals for model B

```

```

res_B = residuals(model_B_1)
op= par(mfrow=c(2,2))
plot.ts(res_B)
abline(h = mean(res_B), col = "blue")
hist(res_B, freq = FALSE)
curve(dnorm(x,mean(res_B), sqrt(var(res_B))), add = TRUE)
qqnorm(res_B)
qqline(res_B)
mean(res_B)
shapiro.test(res_B)

op= par(mfrow=c(1,2))
acf(res_B, lag.max = 50)
pacf(res_B, lag.max = 50)

## Portmanteau test

# Model A

Box.test(res_A, lag = 10, type = c("Box-Pierce"), fitdf = 5)
Box.test(res_A, lag = 10, type = c("Ljung-Box"), fitdf = 5)
Box.test(res_A^2, lag = 10, type = c("Ljung-Box"), fitdf = 0)

#AR(0) check
ar(res_A, aic = TRUE, order.max = NULL, method = c("yule-walker"))

#model B

Box.test(res_B, lag = 10, type = c("Box-Pierce"), fitdf = 4)
Box.test(res_B, lag = 10, type = c("Ljung-Box"), fitdf = 4)
Box.test(res_B^2, lag = 10, type = c("Ljung-Box"), fitdf = 0)

#AR(0) check
ar(res_B, aic = TRUE, order.max = NULL, method = c("yule-walker"))

#Forecasting

#forecast(model_B_1)
pred.tr = predict(model_B_1, n.ahead = 12)
U.tr = pred.tr$pred + 1.96*pred.tr$se
L.tr = pred.tr$pred - 1.96*pred.tr$se
ts.plot(elec.log, xlim = c(1,length(elec.log)+12),
        ylim = c(min(elec.log),max(U.tr)))
lines(U.tr, col = "blue", lty = "dashed")
lines(L.tr, col = "blue", lty = "dashed")
points((length(elec.log)+1):(length(elec.log)+12),
        pred.tr$pred, col = "red")

#plots of forecasting

pred.orig = exp(pred.tr$pred)
U = exp(U.tr)

```

```

L = exp(L.tr)
ts.plot(elec, xlim = c(1,length(elec)+12),
        ylim = c(min(elec), max(U)))
lines(U, col = "blue", lty = "dashed")
lines(L, col = "blue", lty = "dashed")
points((length(elec)+1):(length(elec)+12),
        pred.orig, col = "red")

#Zoomed in plots of forecasting
ts.plot(elec, xlim = c(100,length(elec)+12),
        ylim = c(250,max(U)))
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(elec)+1):(length(elec)+12),
        pred.orig, col="red")

#Zoomed in plots of forecasting with true value points
ts.plot(whole, xlim = c(100,length(elec)+12),
        ylim = c(250,max(U)))
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(elec)+1):(length(elec)+12),
        test, col="green")
points((length(elec)+1):(length(elec)+12),
        pred.orig, col="red")

```