PSTAT174 Final Project U.S. Energy Generation Forecast

Aaron Lee (3410388)

2023-05-30

Contents

Abstract:

In this time series project, the goal is to forecast energy generation by all sectors in the United States using monthly data from January 2001 to March 2022. Energy has become an indispensable thing for human beings, the purpose of this project is to seek and identify trends and patterns in energy production in the United States over this period. In this project, I applied many T.S. techniques including transforming the original data, plotting, acf and pacf to evaluate the best fit model for the energy generation data. In this project, we will select 12 existing data as the basis for forecasting, testing and evaluating the accuracy of models and predictions based on these 12 data points. All in all, $SARIMA(2,1,1) \times (1,1,2)_{12}$ is found to be the most appropriate and accurate predictive model. Through the result, the amount of energy generating is showing a steady development trend, moreover an obvious seasonality.

Introduction:

The purpose of this project is to forecast energy generation by all sectors in the United States using monthly data from January 2001 to March 2022. The generation of energy is tightly related to time, climate, environment, and human factors. Finding the future trends and pattern is extremely important and necessary for the entire country, species, and for the future generation. Because energy has been an irreplaceable element in our life, calculating the potential energy, the storage of energy, energy consumption, to prevent overuse of energy must start from understanding the generating of energy. The goal to find out whether the trend of energy production in the United States is going up or down because of the impact of extreme weather in recent years and the rise of environmental awareness, whether it has an impact on energy production.

The results of the project will provide insight into the past behavior of U.S. energy production and illuminate its future trajectory. The positive result would receive the efficacy of selected models in predicting the dynamics of energy production in all sections of the United States. Methods I used in the project including, Box-Cox transformation, checking variance, acf/pacf, differencing, and diagnostic checking.

The data I used in this project is from a reliable and reputable source, kaggle,and the U.S. Energy Information Administration (EIA) database. Packages that were used: "astsa", "MuMIn", "tsdl", "MASS", "ggplot2", "ggfortify", "qpcR" and "forecast"

Data Source detail:

Source: U.S. Energy Information Administration

Release: U.S. Department of Energy

Units: thousand megawatts hours

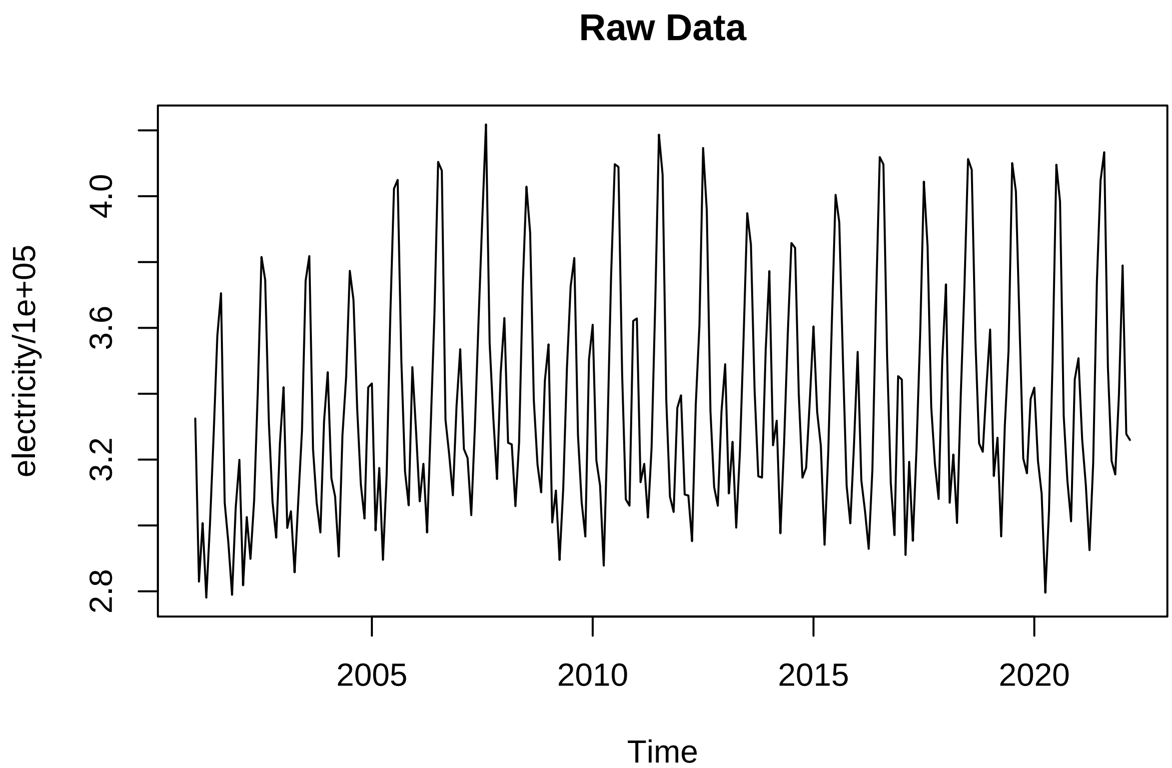
Frequency: Monthly

EIA collects data about the net electricity generation for the U.S. (spanning monthly from 2001-01-01 to 2022-03-01)

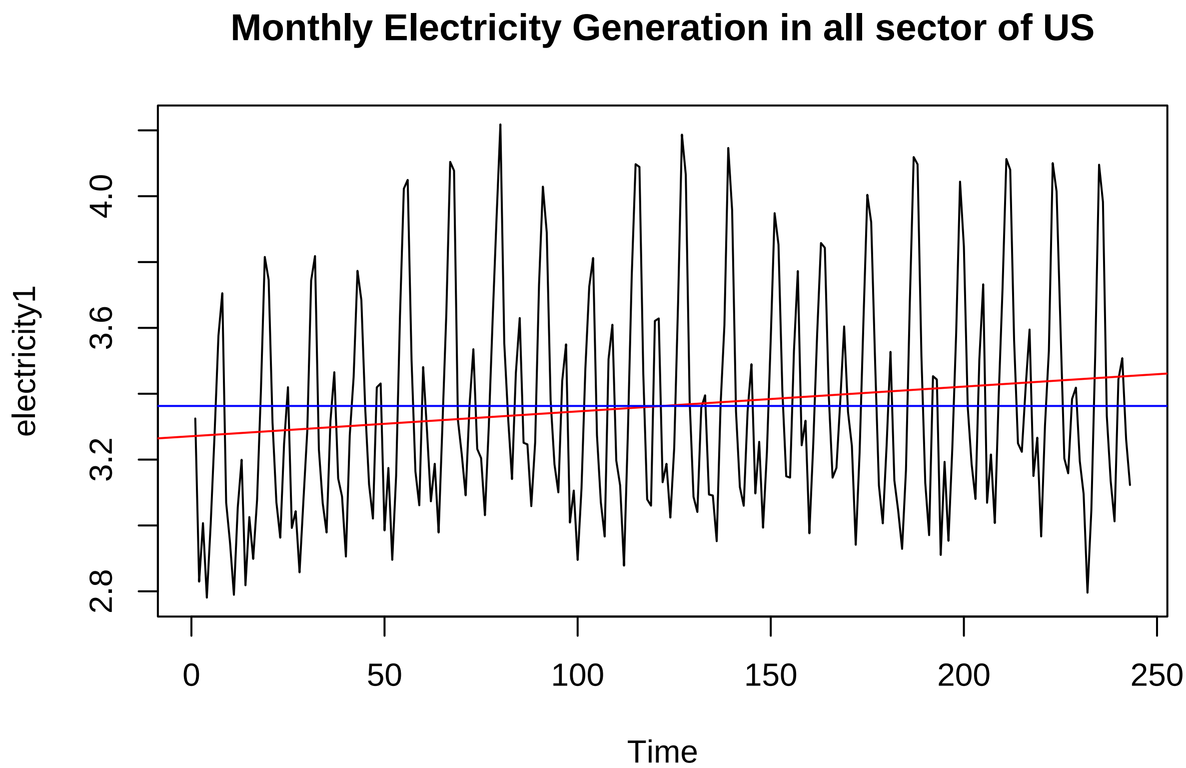
```
electricity.csv <- read.table("electricity_data.csv", sep = ",",
                              header = FALSE, skip = 1, nrows = 255)
electricity <- ts(electricity.csv[, 2], start = c(2001, 1, 1), frequency=12)
electricity1 = electricity[c(1: 243)]/100000 # devide by 100000 or Box-Cox would be too small
electricity1_test = electricity[c(244: 255)]/100000
```

Plotting the original data

```
ts.plot(electricity/100000, main = "Raw Data")
```




```
ts.plot(electricity1, main="Monthly Electricity Generation in all sector of US",
        ylab="electricity1")
ele_fit <- lm(electricity1 ~ as.numeric(1:length(electricity1))); abline(ele_fit, col="red")
abline(h=mean(electricity1), col="blue")
```

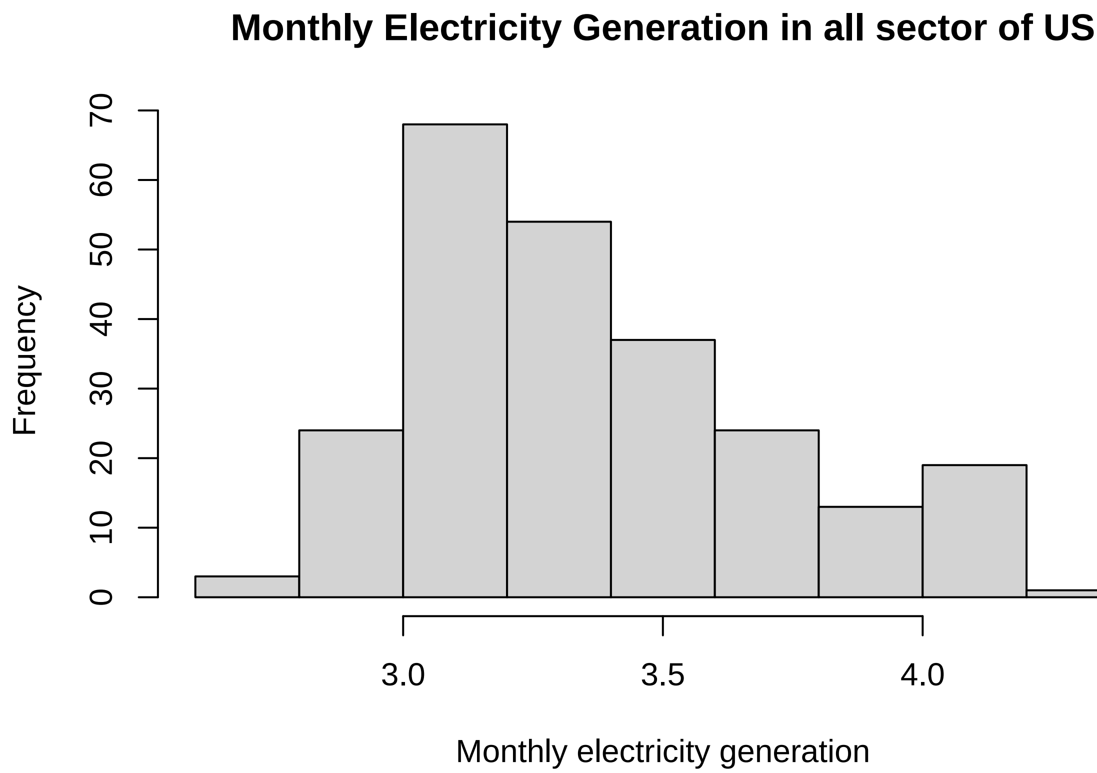


- We can see that the data is not stationary;
- Through the plot above, we observed that there is an upper trend; seasonality; No constant variance and mean.

Confirming non-stationary of the original data through plot.

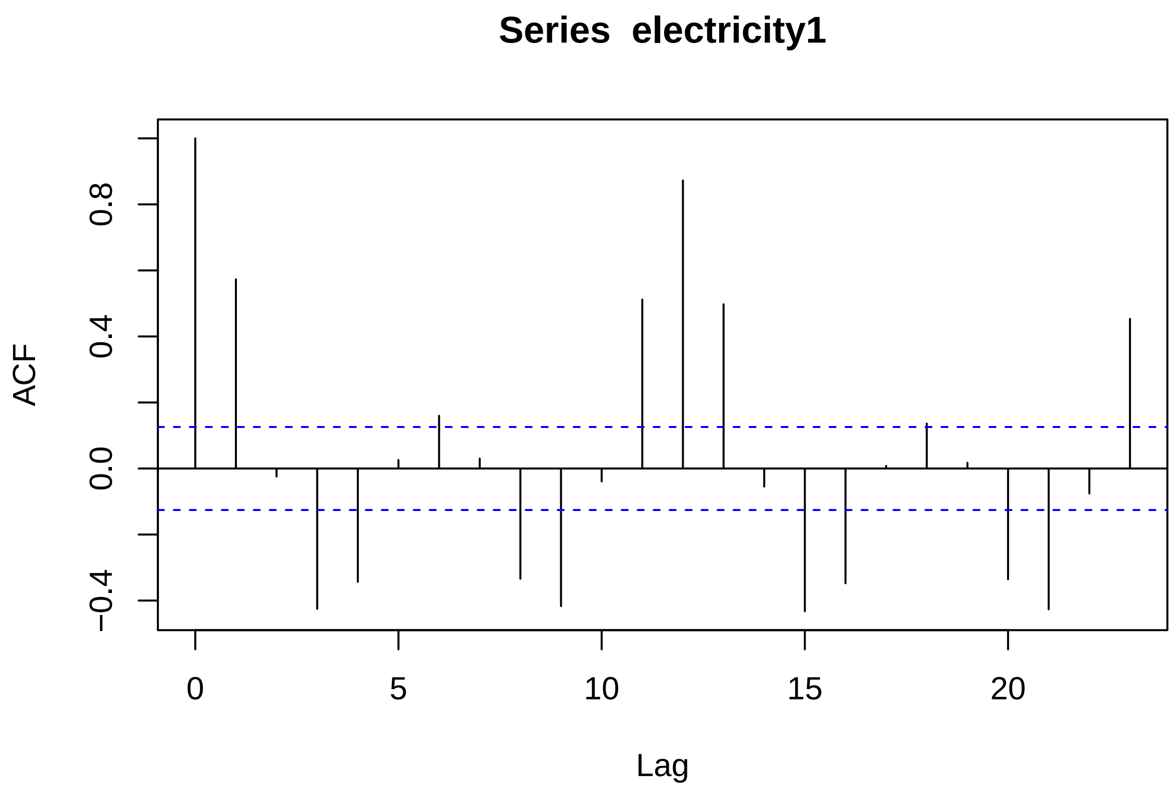
The histogram is skewed, not bell shaped.

```
hist(electricity1, main="Monthly Electricity Generation in all sector of US",
     xlab="Monthly electricity generation")
```



The acf remains large periodic.

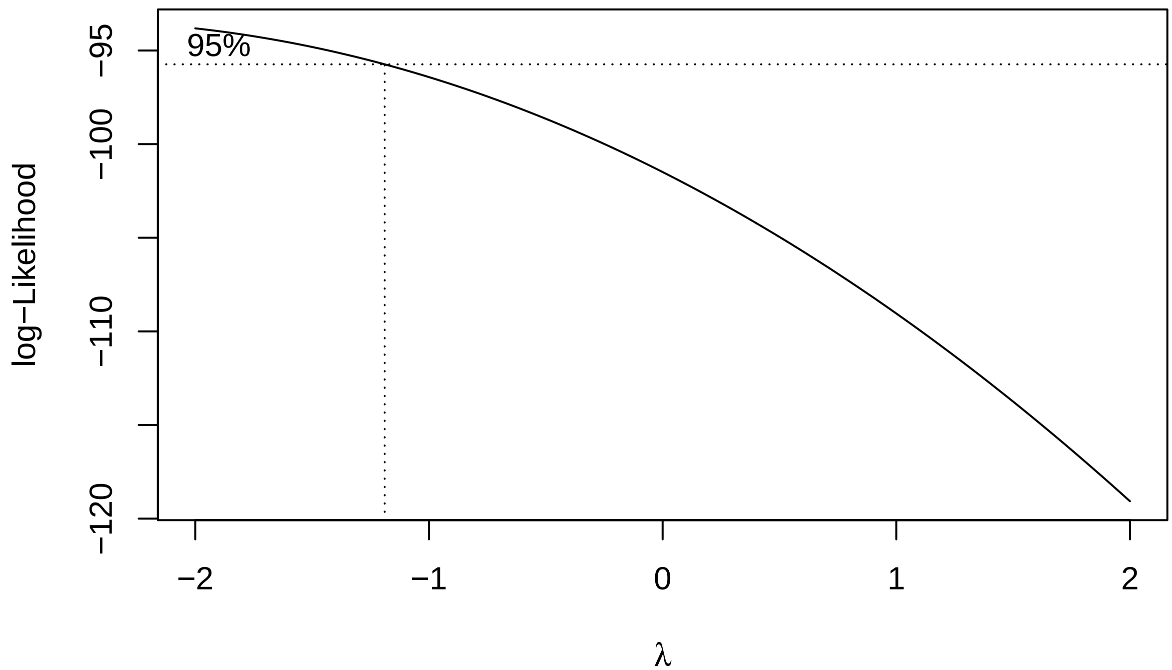
```
acf(electricity1)
```



Transformation of the energy generation data

The purpose of transforming the data is to stabilize the variance. Through the information above, the data is skewed & variance is non constant. **Try Box-Cox Transformation**

```
library(MASS)
t <- 1:length(electricity1)
bcTransform <- boxcox(electricity1 ~ t, plotit=TRUE) # plotting the graph
```

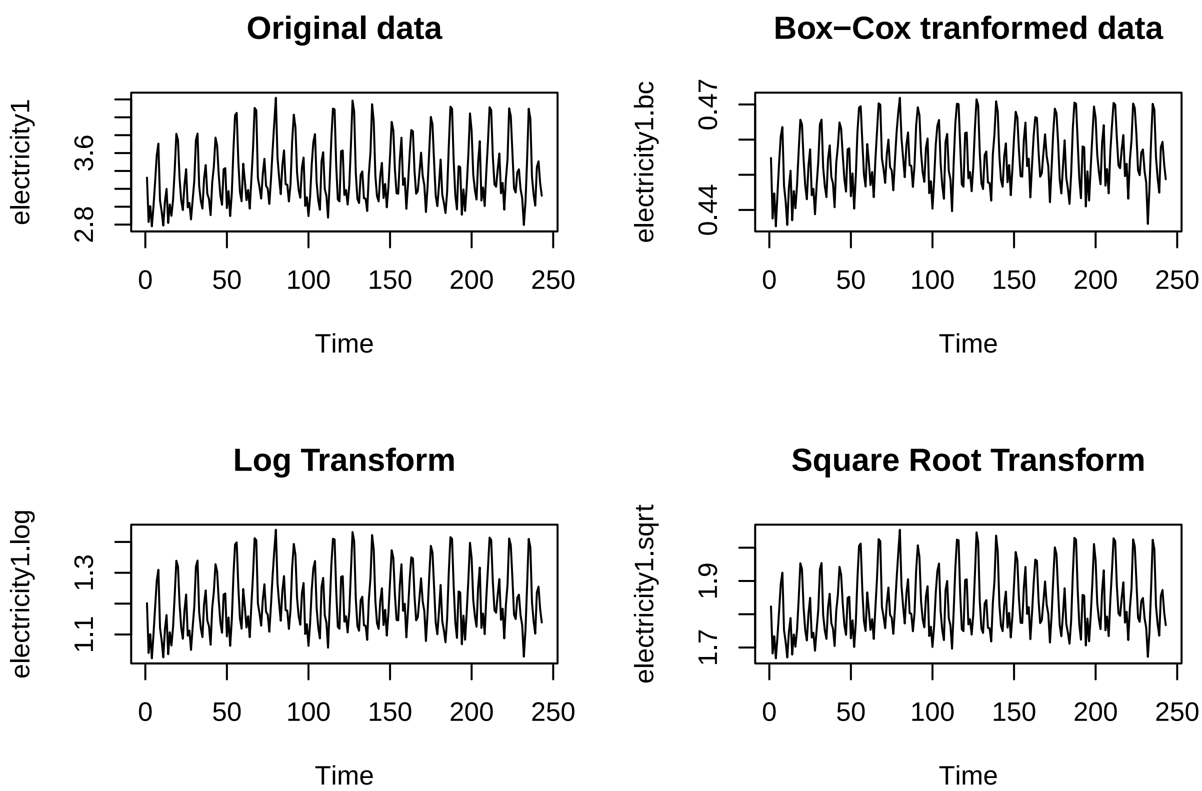



```
bcTransform$x[which(bcTransform$y == max(bcTransform$y))] # get the value of lambda
```

```
## [1] -2
```

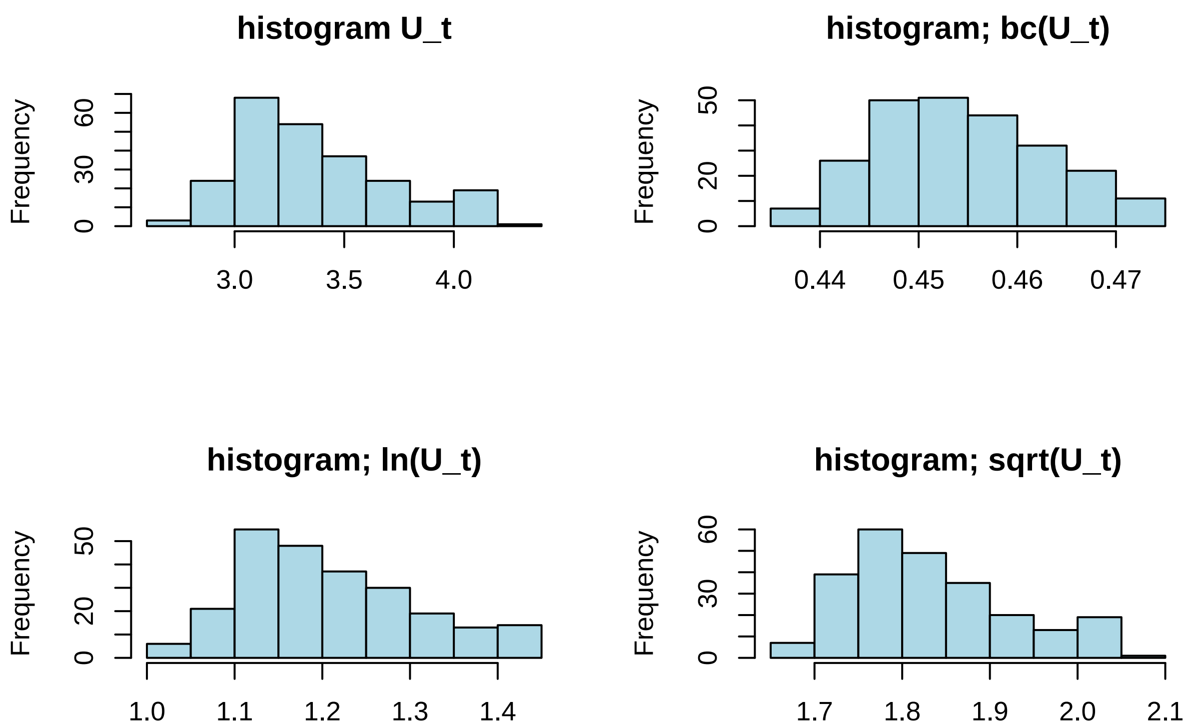
Comparing the difference between each transformation.

```
# Because lambda is -2, I tentatively set box-cox as the best transformation
lambda = bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
electricity1.bc = (1/lambda)*(electricity1^lambda-1)
electricity1.log = log(electricity1)
electricity1.sqrt = sqrt(electricity1)

op <- par(mfrow = c(2,2))
ts.plot(electricity1,main = "Original data")
ts.plot(electricity1.bc,main = "Box-Cox tranformed data")
ts.plot(electricity1.log, main = "Log Transform")
ts.plot(electricity1.sqrt, main = "Square Root Transform")
```




```
par(op)
```

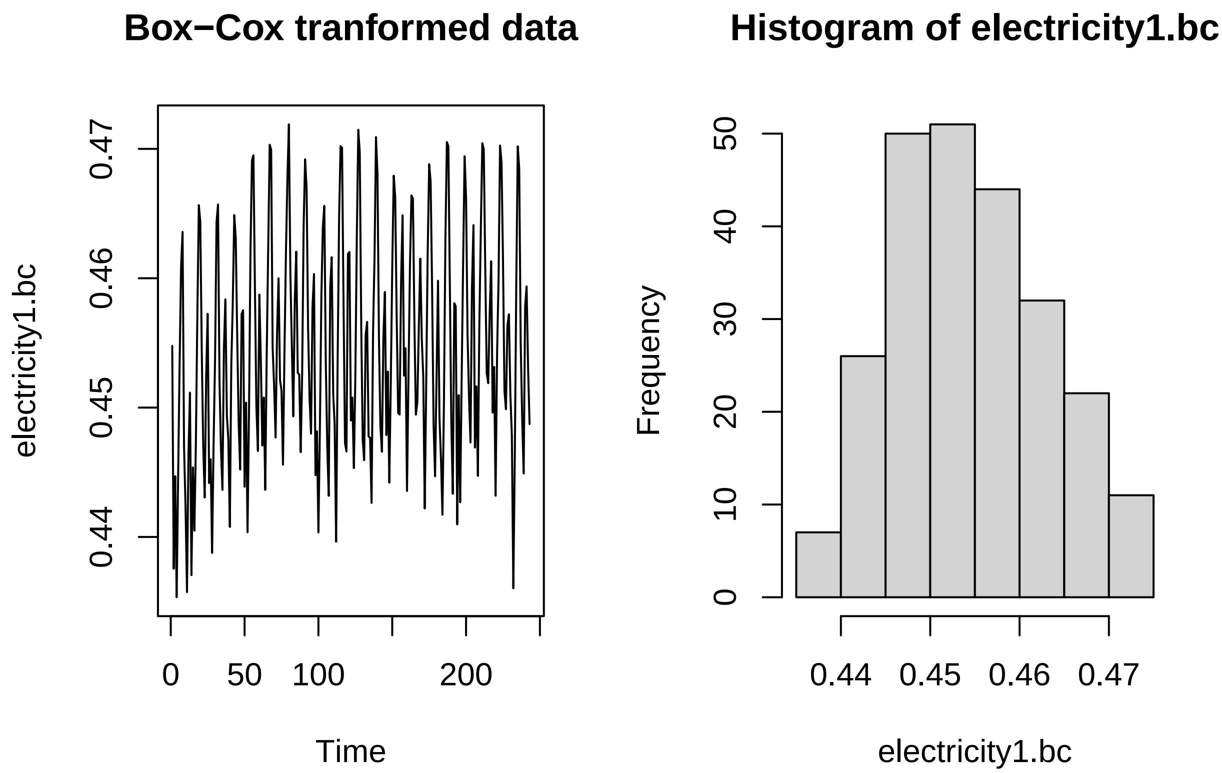
```
op <- par(mfrow = c(2,2))
hist(electricity1, col = "light blue", xlab = "", main = "histogram U_t")
hist(electricity1.bc, col = "light blue", xlab = "", main = "histogram; bc(U_t)")
hist(electricity1.log,col = "light blue", xlab = "", main = "histogram; ln(U_t)")
hist(electricity1.sqrt,col = "light blue", xlab = "", main = "histogram; sqrt(U_t)")
```




```
par(op)
```

Select Box-Cox Transformation

```
op <- par(mfrow = c(1,2))
ts.plot(electricity1.bc,main = "Box-Cox tranformed data")
hist(electricity1.bc)
```



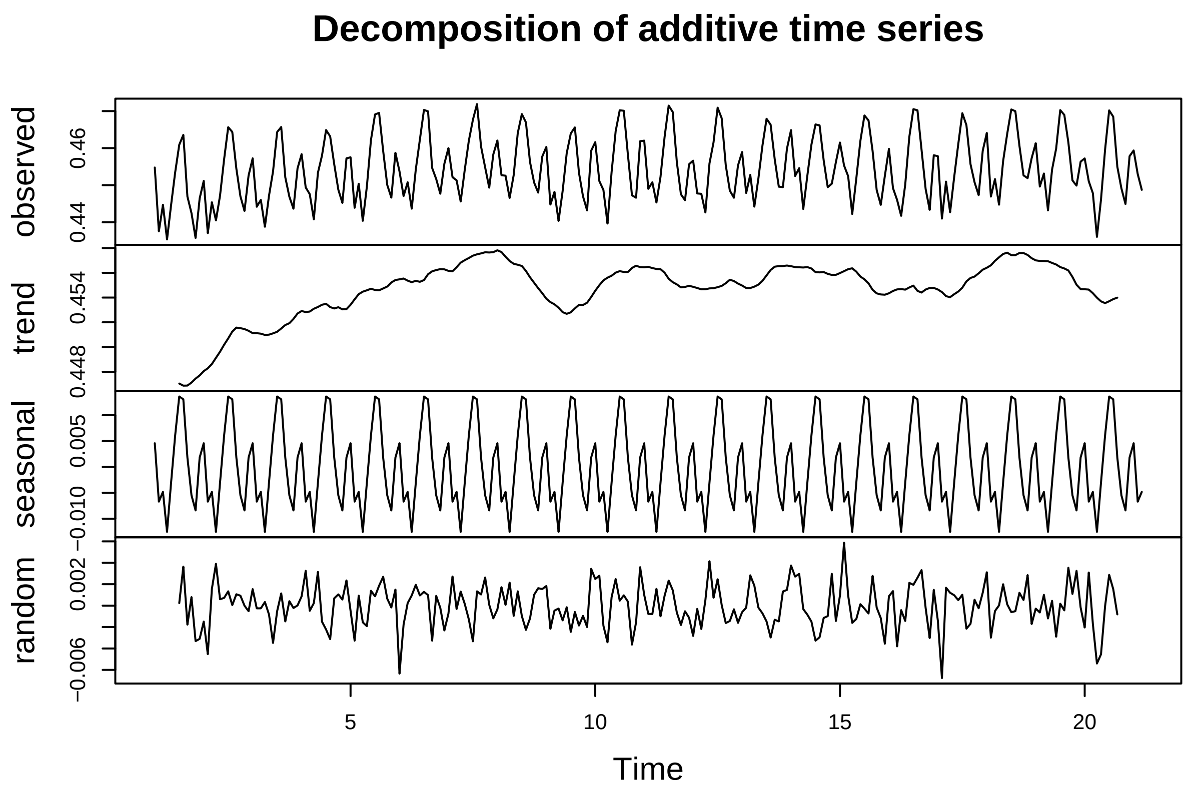

```
par(op)
```

- Box-Cox transform gives a more symmetric histogram.
- The variance of the data after box-cox transformation looks more even.

Produce decomposition of Box-Cox U_t

We can see that decomposition of Box-Cox U_t show us the seasonality and trend.

```
library(ggplot2)
#install.packages('ggfortify')
library(ggfortify)
y <- ts(as.ts(electricity1.bc), frequency = 12)
decomp <- decompose(y)
plot(decomp)
```



Check if the transformation is necessary

```
# Calculate the sample variance and plot the acf/pacf
var(electricity1)
```

```
## [1] 0.116328
```

```
var(electricity1.bc) # the variance before difference
```

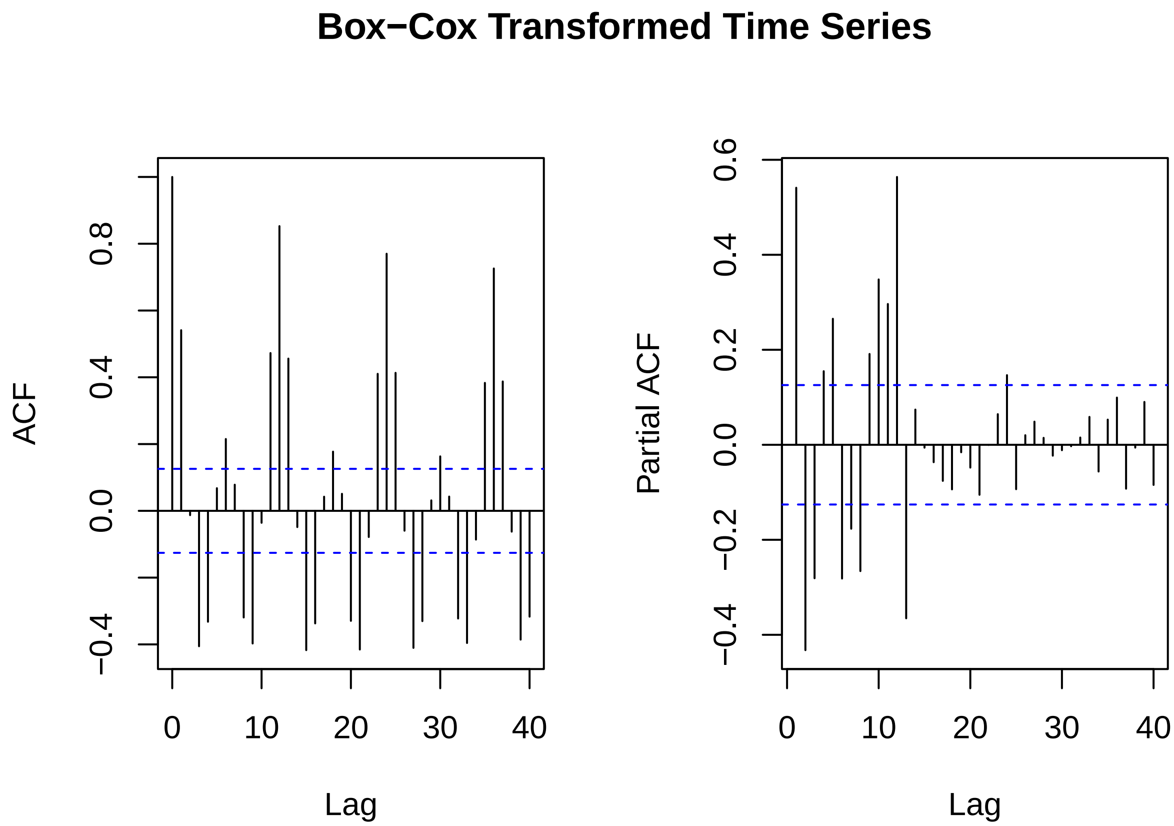
```
## [1] 7.342597e-05
```

The variance decreases after the transformation. Therefore, the transformation is necessary.

Plotting ACF/PACF before removing trend and seasonality.

ACF decays slowly and exist multiple peaks, that indicates non-stationarity.

```
op = par(mfrow = c(1,2))
acf(electricity1.bc,lag.max = 40,main = "")
pacf(electricity1.bc,lag.max = 40,main = "")
title("Box-Cox Transformed Time Series", line = -1, outer=TRUE)
```




```
par(op)
```

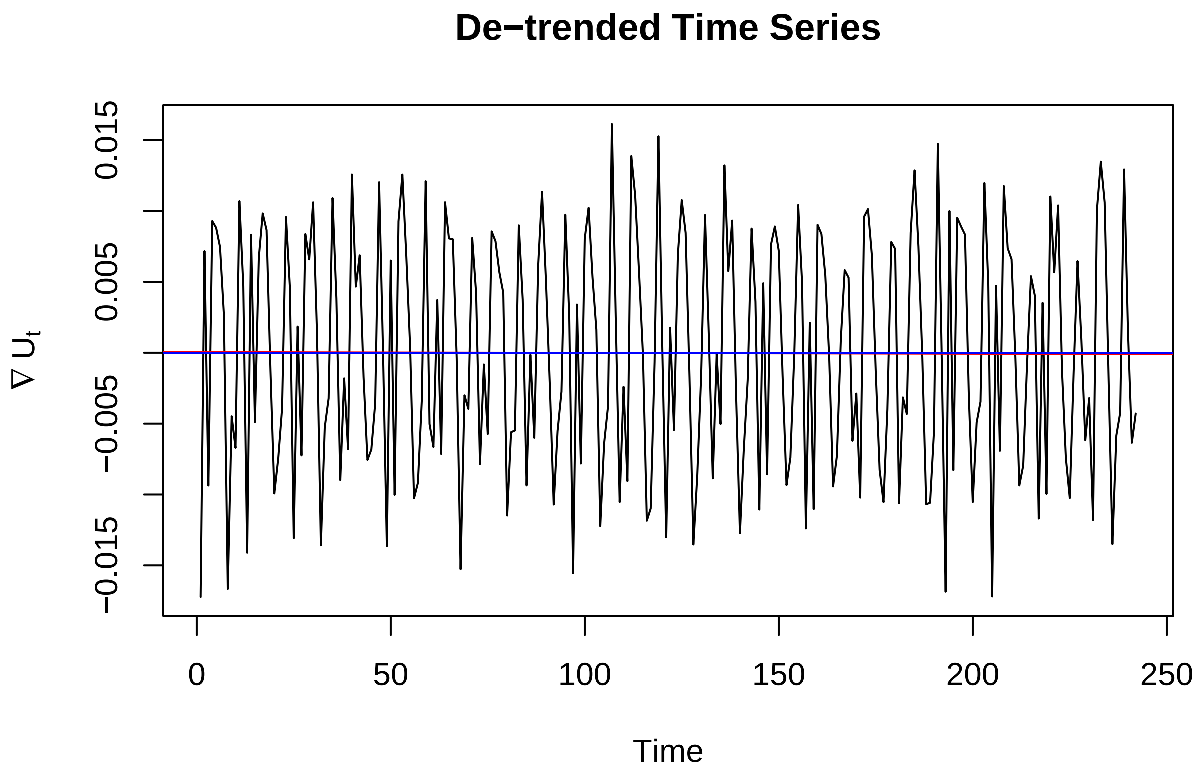
Differencing Box-Cox U_t

Differenced at lag 1 to removing the trend component

```
y1 = diff(electricity1.bc, 1)
plot.ts(y1,main = "De-trended Time Series",ylab = expression(nabla~U[t]))
fit1 <- lm(y1 ~ as.numeric(1:length(y1)))
abline(fit1, col = "red")
mean(y1)
```

```
## [1] -2.497014e-05
```

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




```
var(y1) # smaller that 7.342619e-5
```

```
## [1] 6.7531e-05
```

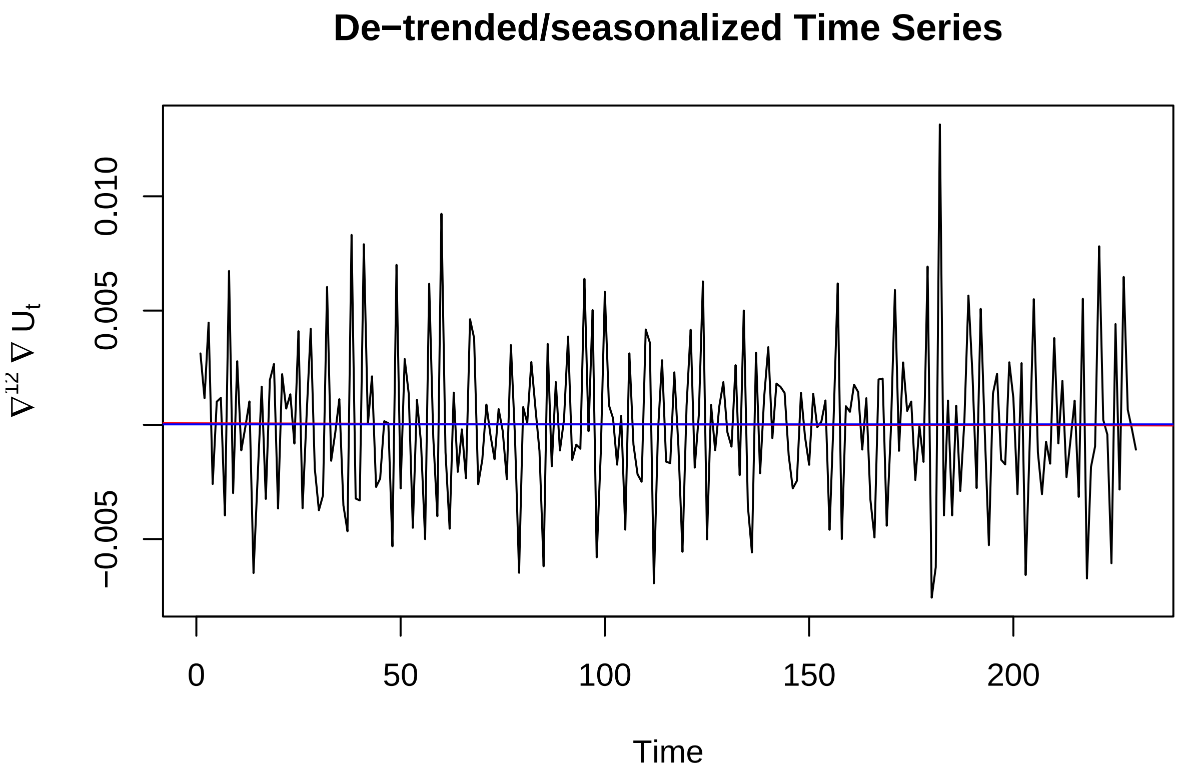
- The upper trend is no longer apparent
- The variance: 6.7531e-05 is smaller than 7.342597e-05.
- Seasonality still exists.

Differenced at lag 1 and then lag = 12 (cycle determined by the ACF) to remove seasonal component

```
y12 = diff(y1, 12)
plot.ts(y12, main = "De-trended/seasonalized Time Series", ylab =
          expression(nabla^{12}~nabla~U[t]))
fit1 <- lm(y12 ~ as.numeric(1:length(y12)))
abline(fit1, col = "red")
mean(y12)
```

```
## [1] 1.963438e-05
```

```
abline(h = mean(y12), col = "blue")
```




```r
var(y12) # smaller than 7.342619e-5 and 6.753134e-5
```

```
## [1] 1.163955e-05
```

- No trend and seasonality
- The variance: 1.163955e-05 (getting lower)
- Data looks stationary, next step check acf

Plot of $bc(U_t)$

- Seasonality
- Trend
- Variance: 7.342597e-05

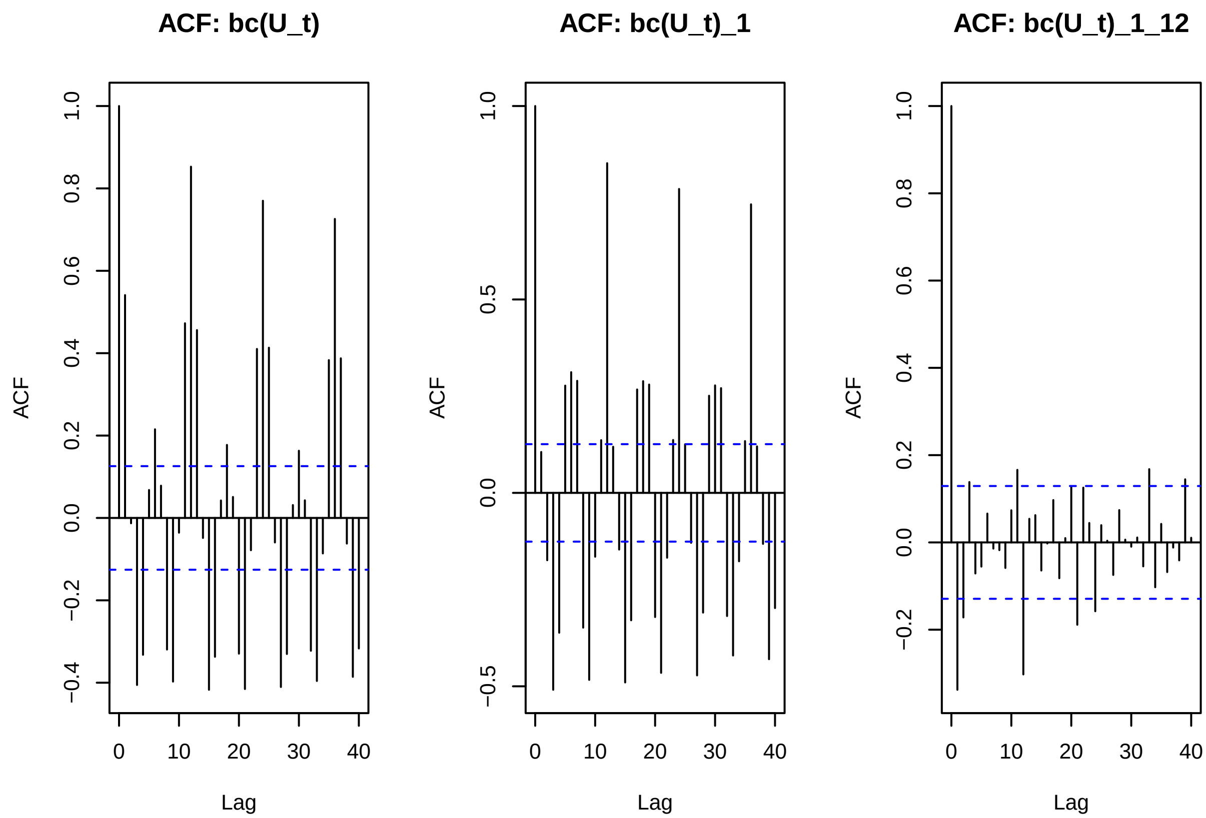
Plot of $bc(U_t)$ differenced at lag 1

- Seasonality
- The upper trend is no longer apparent
- Variance: 6.7531e-05

Plot of $bc(U_t)$ differenced at lag 1 and then 12

- No Seasonality
- No trend
- Variance: 1.163955e-05

```r
par(mfrow = c(1, 3))
acf(electricity1.bc,lag.max = 40,main = "ACF: bc(U_t)")
acf(y1,lag.max = 40,main = "ACF: bc(U_t)_1")
acf(y12,lag.max = 40,main = "ACF: bc(U_t)_1_12")
```




```r
par(op)
```

Plot of ACF of $bc(U_t)$

- ACF decays slowly and exist multiple peaks, that indicates non-stationarity.
- Seasonality

Plot of ACF of $bc(U_t)$ differenced at lag 1

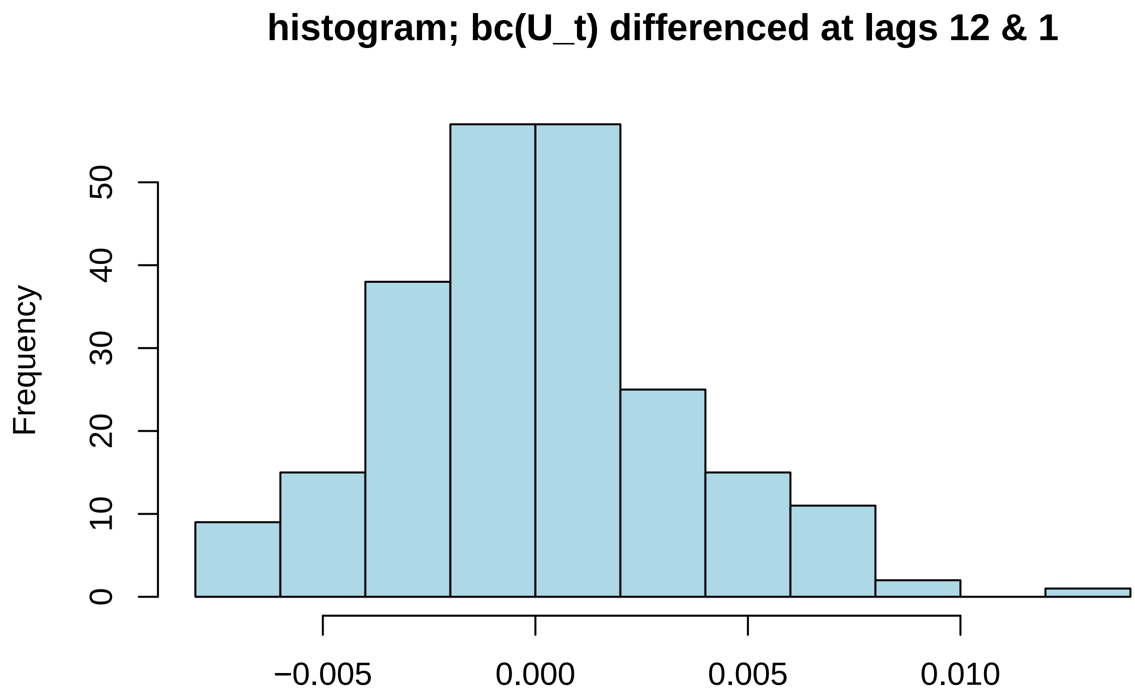
- ACF decays slowly and exist multiple peaks, that indicates non-stationarity.
- Seasonality still exist.

Plot of ACF of $bc(U_t)$ differenced at lag 1 and 12

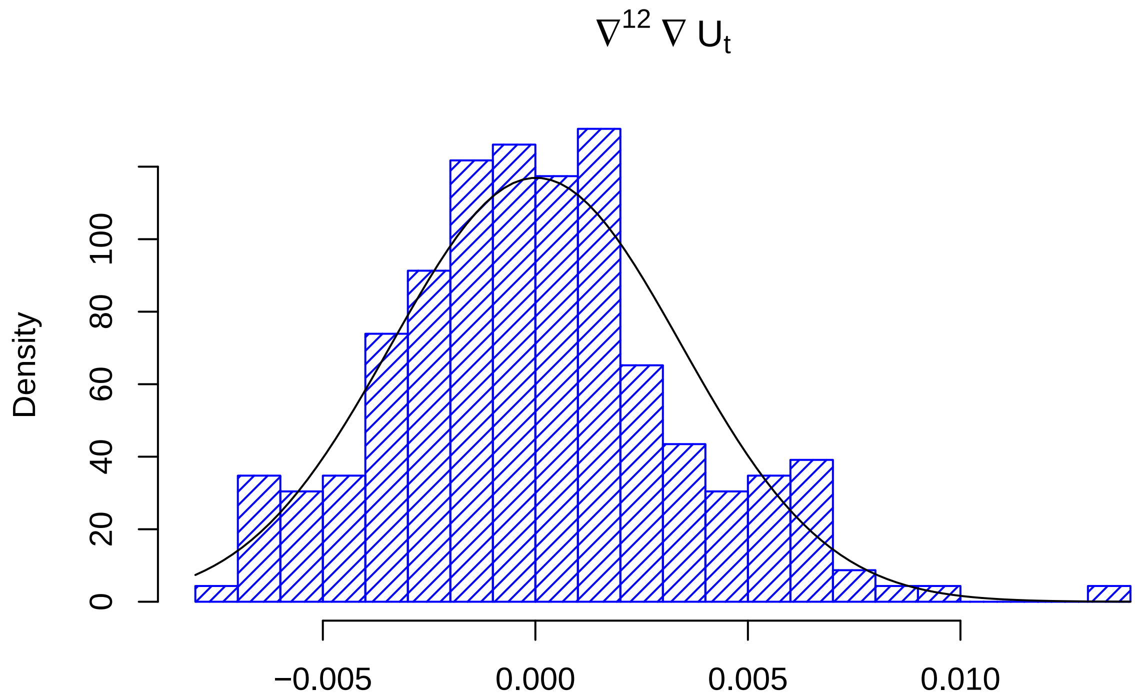
- ACF decay corresponds to a stationary process
- Work with data $\nabla_1\nabla_{12}bc(U_t)$, U_t = the first 243 observations of the original data.

Histogram of $\nabla_1\nabla_{12}bc(U_t)$

```r
# Compare histograms of Box-Cox (Ut) to the normal curve, really similar.
hist(y12, col="light blue", xlab="", main="histogram; bc(U_t) differenced at lags 12 & 1")
```



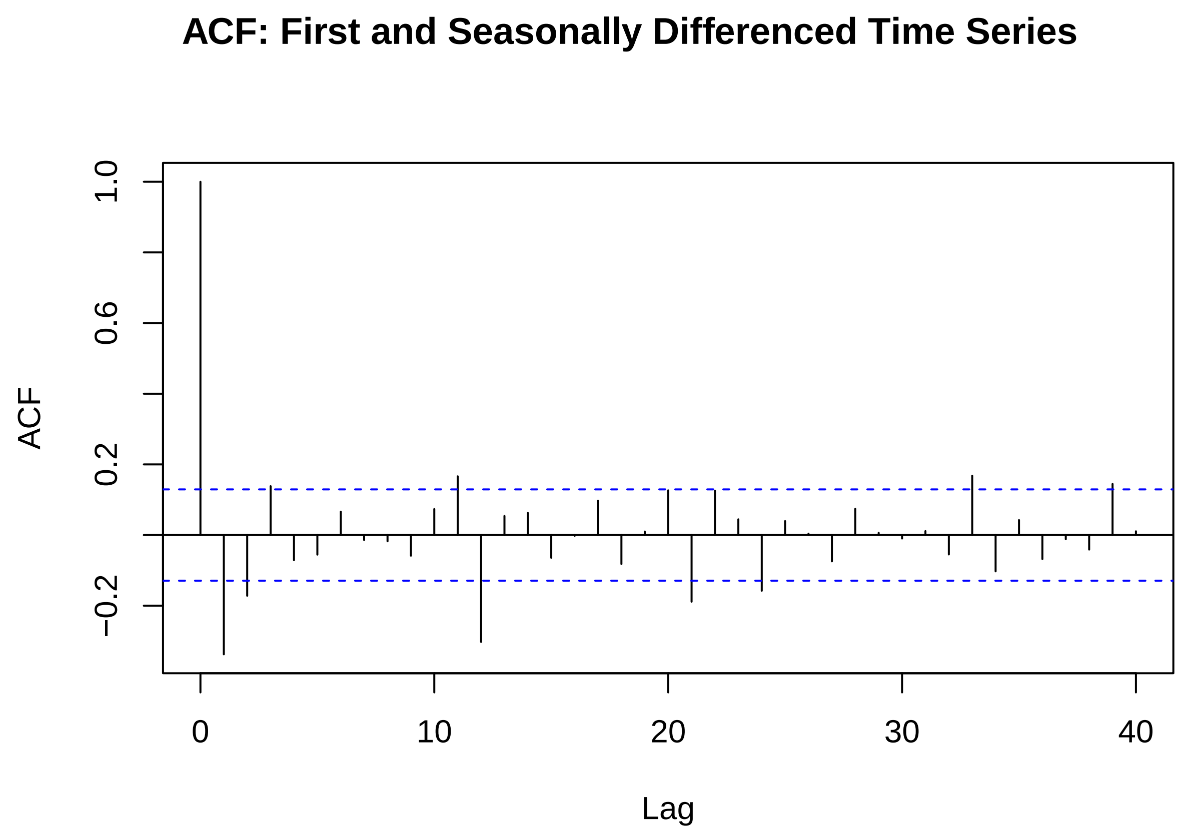

```
hist(y12, density=20,breaks=20, col="blue", xlab="", main = expression(nabla^{12}~nabla~U[t]),
     prob=TRUE)
m<-mean(y12)
std<- sqrt(var(y12))
curve( dnorm(x,m,std), add=TRUE )
```



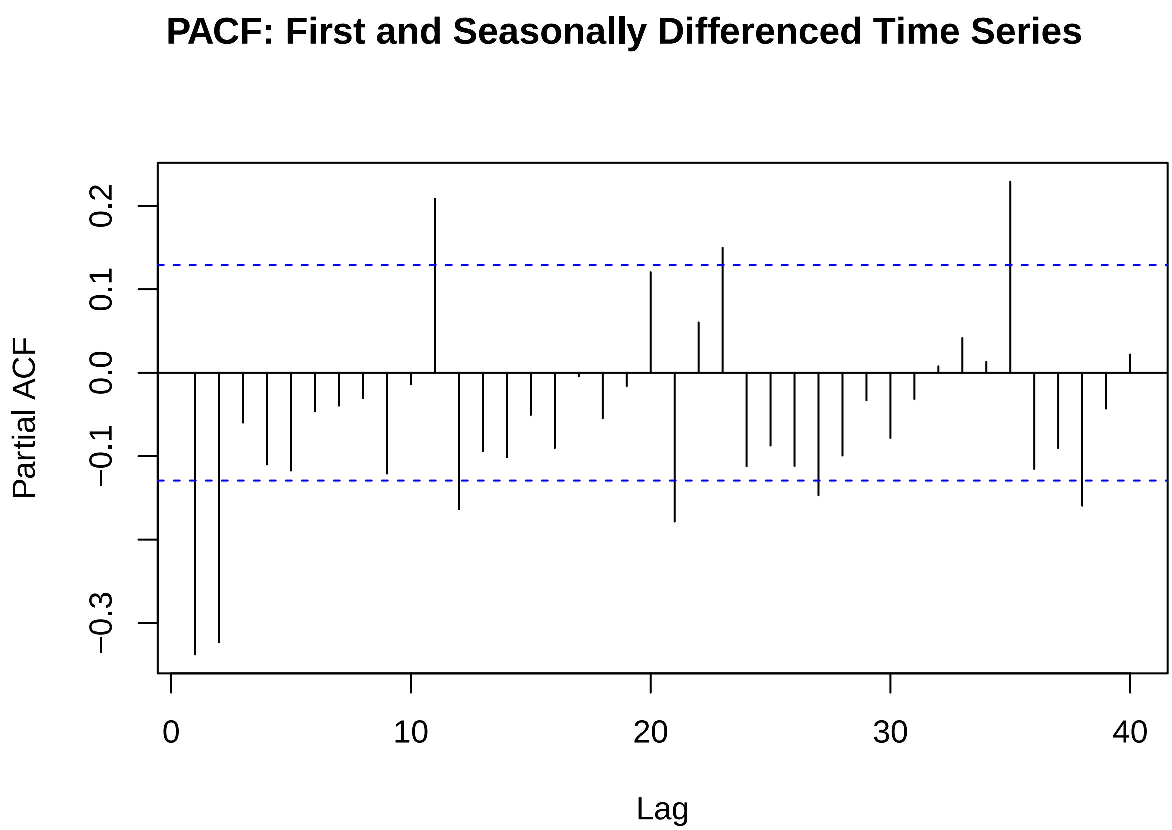
Histogram of $\nabla_1 \nabla_{12} bc(U_t)$ looks symmetric and almost Gaussian.

ACF and PACF of Box-Cox U_t after differences at lag 1 and lag 12

```
acf(y12,lag.max = 40,main = "")
title("ACF: First and Seasonally Differenced Time Series", line = -1, outer = TRUE)
```




```r
pacf(y12,lag.max = 40,main = "")
title("PACF: First and Seasonally Differenced Time Series", line = -1, outer = TRUE)
```



Determine possible candidate models $SARIMA(p, d, q) \times (P, D, Q)_s$ for the series bcU_t *Modeling the seasonal part* (P, D, Q): For this part, focus on the seasonal lags h = 1s, 2s, etc.

- We applied one seasonal differencing so D = 1 at lag s = 12.
- The ACF shows a strong peak at h = 1s and smaller peaks appearing at h = 2s. A good choice for the MA part could be Q = 1 or Q = 2.
- The PACF shows there is a peak at h = 1s. A good choice for the AR part could be P = 1.

Modeling the non-seasonal part (p, d, q): In this case focus on the within season lags, h = 1,. . . ,11.

- We applied one differencing to remove the trend: d = 1.
- A good choice for the MA part could be q = 0 or q = 1 respectively.
- A good choice for the AR part could be p = 2

As an illustration, the model might be:
$MA(33)$
$SARIMA(2, 1, 0) \times (1, 1, 1)_{12}$
$SARIMA(2, 1, 1) \times (1, 1, 1)_{12}$
$SARIMA(2, 1, 0) \times (1, 1, 2)_{12}$
$SARIMA(2, 1, 1) \times (1, 1, 2)_{12}$

Evaluating Models:

SMA models tried: Q=1, 2, q=0,1. Model producing the lowest AICc:

```r
library(astsa)
library(MuMIn)
```

```r
arima(electricity1.bc, order = c(0,1,1), seasonal = list(order = c(0,1,2),
                                       period = 12), method="ML")
```

```
##
## Call:
## arima(x = electricity1.bc, order = c(0, 1, 1), seasonal = list(order = c(0,
##     1, 2), period = 12), method = "ML")
##
## Coefficients:
##           ma1     sma1     sma2
##       -0.6406  -0.7834  -0.2164
## s.e.   0.0674   0.1087   0.0773
##
## sigma^2 estimated as 5.337e-06:  log likelihood = 1053.52,  aic = -2099.04
```

```r
# Calculating AICc
AICc(arima(electricity1.bc, order = c(0,1,1), seasonal = list(order = c(0,1,2), period = 12),
           method="ML"))
```

```
## [1] -2098.865
```

$SARIMA(0,1,1) \times (0,1,2)_{12}$ AICc: -2098.865

```r
arima(electricity1.bc, order = c(0,1,0), seasonal = list(order = c(0,1,2),
                                       period = 12), method="ML")
```

```
##
## Call:
## arima(x = electricity1.bc, order = c(0, 1, 0), seasonal = list(order = c(0,
##     1, 2), period = 12), method = "ML")
##
## Coefficients:
##          sma1     sma2
##       -0.7795  -0.2205
## s.e.   0.1666   0.0788
##
## sigma^2 estimated as 6.906e-06:  log likelihood = 1024.18,  aic = -2042.36
```

```r
AICc(arima(electricity1.bc, order = c(0,1,0), seasonal = list(order = c(0,1,2), period = 12),
           method="ML"))
```

```
## [1] -2042.249
```

$SARIMA(0,1,0) \times (0,1,2)_{12}$ AICc: -2042.249 (bigger than -2098.865)

```r
arima(electricity1.bc, order = c(0,1,1), seasonal = list(order = c(0,1,1), period = 12), method="ML")
```

```
##
## Call:
```

```
## arima(x = electricity1.bc, order = c(0, 1, 1), seasonal = list(order = c(0,
##     1, 1), period = 12), method = "ML")
##
## Coefficients:
##           ma1     sma1
##       -0.6530  -0.9816
## s.e.   0.0689   0.2676
##
## sigma^2 estimated as 5.498e-06:  log likelihood = 1050.11,  aic = -2094.22
```

```
AICc(arima(electricity1.bc, order = c(0,1,1), seasonal = list(order = c(0,1,1), period = 12),
           method="ML"))
```

```
## [1] -2094.118
```

$SARIMA(0,1,1) \times (0,1,1)_{12}$ AICc: -2094.118 (bigger than -2098.865)

```
arima(electricity1.bc, order = c(0,1,0), seasonal = list(order = c(0,1,1), period = 12), method="ML")
```

```
##
## Call:
## arima(x = electricity1.bc, order = c(0, 1, 0), seasonal = list(order = c(0,
##     1, 1), period = 12), method = "ML")
##
## Coefficients:
##          sma1
##       -0.9121
## s.e.   0.0668
##
## sigma^2 estimated as 7.481e-06:  log likelihood = 1020.46,  aic = -2036.92
```

```
AICc(arima(electricity1.bc, order = c(0,1,0), seasonal = list(order = c(0,1,2), period = 12),
           method="ML"))
```

```
## [1] -2042.249
```

$SARIMA(0,1,0) \times (0,1,1)_{12}$ AICc: -2042.249 (bigger than -2098.865)

SAR

```
arima(electricity1.bc, order = c(2,1,0), seasonal = list(order = c(1,1,0), period = 12), method="ML")
```

```
##
## Call:
## arima(x = electricity1.bc, order = c(2, 1, 0), seasonal = list(order = c(1,
##     1, 0), period = 12), method = "ML")
##
## Coefficients:
##           ar1      ar2     sar1
##       -0.4371  -0.3102  -0.3058
## s.e.   0.0629   0.0627   0.0656
##
## sigma^2 estimated as 8.353e-06:  log likelihood = 1017.57,  aic = -2027.14
```

```r
AICc(arima(electricity1.bc, order = c(2,1,0), seasonal = list(order = c(1,1,0), period = 12),
           method="ML"))
```

```
## [1] -2026.967
```

$SARIMA(2,1,0) \times (1,1,0)_{12}$ AICc: -2026.967 (bigger than -2098.865)

SARIMA(2,1,1)(1,1,2)_s=12

```r
arima(electricity1.bc, order = c(2,1,1), seasonal = list(order = c(1,1,2), period = 12), method="ML")
```

```
##
## Call:
## arima(x = electricity1.bc, order = c(2, 1, 1), seasonal = list(order = c(1,
##     1, 2), period = 12), method = "ML")
##
## Coefficients:
##          ar1     ar2      ma1     sar1     sma1     sma2
##       0.3109  0.0653  -0.8828  -0.2344  -0.5707  -0.4286
## s.e.  0.0922  0.0836   0.0636   0.2006   0.2238   0.1865
##
## sigma^2 estimated as 5.103e-06:  log likelihood = 1057.82,  aic = -2101.64
```

```r
AICc(arima(electricity1.bc, order = c(2,1,1), seasonal = list(order = c(1,1,2), period = 12),
           method="ML"))
```

```
## [1] -2101.137
```

$SARIMA(2,1,1) \times (1,1,2)_{12}$ AICc: -2101.137 (smaller than -2098.865)

Best fit model (smallest AICc)

```r
arima(electricity1.bc, order = c(2,1,1), seasonal = list(order = c(1,1,2), period = 12),
      fixed = c(NA,0,NA,NA,NA,NA),method="ML")
```

```
##
## Call:
## arima(x = electricity1.bc, order = c(2, 1, 1), seasonal = list(order = c(1,
##     1, 2), period = 12), fixed = c(NA, 0, NA, NA, NA, NA), method = "ML")
##
## Coefficients:
##          ar1  ar2      ma1     sar1     sma1     sma2
##       0.2889    0  -0.8478  -0.2373  -0.5696  -0.4305
## s.e.  0.0995    0   0.0644   0.2002   0.2199   0.1861
##
## sigma^2 estimated as 5.117e-06:  log likelihood = 1057.53,  aic = -2103.06
```

```r
AICc(arima(electricity1.bc, order = c(2,1,1), seasonal = list(order = c(1,1,2), period = 12),
           fixed = c(NA,0,NA,NA,NA,NA),method="ML"))
```

```
## [1] -2102.679
```

$SARIMA(2,1,1) \times (1,1,2)_{12}$ AICc: -2102.679 (smaller than -2101.137)

MA(33)

```
arima(electricity1.bc, order = c(0,0,33), seasonal = list(order = c(0,0,0), period = 12),
      method="ML")
AICc(arima(electricity1.bc, order = c(0,0,33), seasonal = list(order = c(0,0,0), period = 12),
           method="ML"))
```

AICc: -1981.432 (not smaller than -2102.679)

---

second less AICc

```
arima(electricity1.bc, order = c(2,1,1), seasonal = list(order = c(1,1,1), period = 12),method="ML")
```

```
##
## Call:
## arima(x = electricity1.bc, order = c(2, 1, 1), seasonal = list(order = c(1,
##     1, 1), period = 12), method = "ML")
##
## Coefficients:
##          ar1     ar2      ma1    sar1     sma1
##       0.3157  0.0656  -0.8819  0.1519  -0.9999
## s.e.  0.0940  0.0845   0.0660  0.0735   0.1372
##
## sigma^2 estimated as 5.197e-06:  log likelihood = 1055.99,  aic = -2099.97
```

```
AICc(arima(electricity1.bc, order = c(2,1,1), seasonal = list(order = c(1,1,1), period = 12),
           method="ML"))
```

```
## [1] -2099.598
```

$SARIMA(2,1,1) \times (1,1,1)_{12}$ AICc: -2099.598 (not smaller than -2102.679)

Notice that Θ_1 is -0.9999, which is extremely close to -1

```
arima(electricity1.bc, order = c(2,1,1), seasonal = list(order = c(1,1,1), period = 12),
      fixed = c(NA,0,NA,NA,NA),method="ML")
```

```
##
## Call:
## arima(x = electricity1.bc, order = c(2, 1, 1), seasonal = list(order = c(1,
##     1, 1), period = 12), fixed = c(NA, 0, NA, NA, NA), method = "ML")
##
## Coefficients:
##          ar1  ar2      ma1    sar1     sma1
##       0.2924    0  -0.8456  0.1500  -1.0001
## s.e.  0.1005    0   0.0653  0.0729   0.1313
##
## sigma^2 estimated as 5.214e-06:  log likelihood = 1055.7,  aic = -2101.39
```

```
AICc(arima(electricity1.bc, order = c(2,1,1), seasonal = list(order = c(1,1,1), period = 12),
           fixed = c(NA,0,NA,NA,NA),method="ML"))
```

```
## [1] -2101.127
```

AICc: -2101.127
$\Theta_1 = -1.0001$, $|\Theta_1| > 1$. Therefore, it's not invertible.

Conclude:
$SARIMA(2,1,1) \times (1,1,2)_{12}$ is the best fit model.

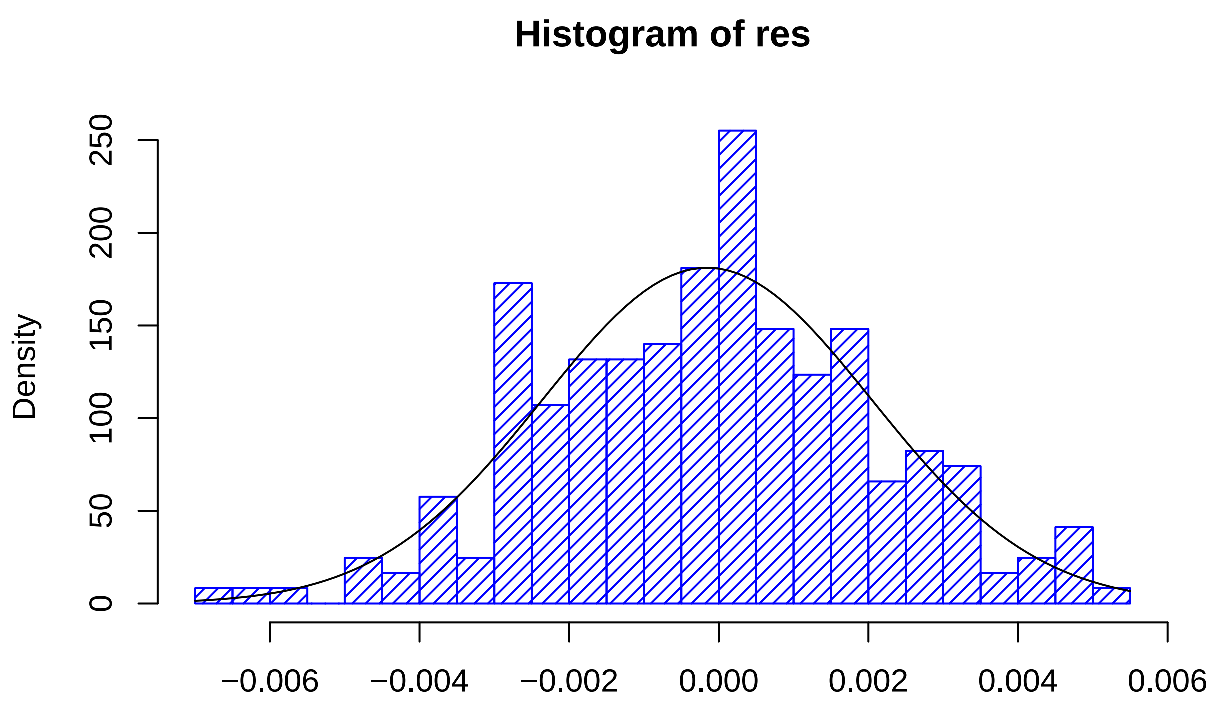
- $\phi_1 = 0.2889$ s.e. $= 0.0995$
- $\theta_1 = -0.8478$ s.e. $= 0.0644$
- $\Phi_1 = -0.2373$ s.e. $= 0.2002$
- $\Theta_1 = -0.5696$ s.e. $= 0.2199$
- $\Theta_2 = -0.4305$ s.e. $= 0.1861$
- $\hat{\sigma^2} = 5.117e-06$

Model:

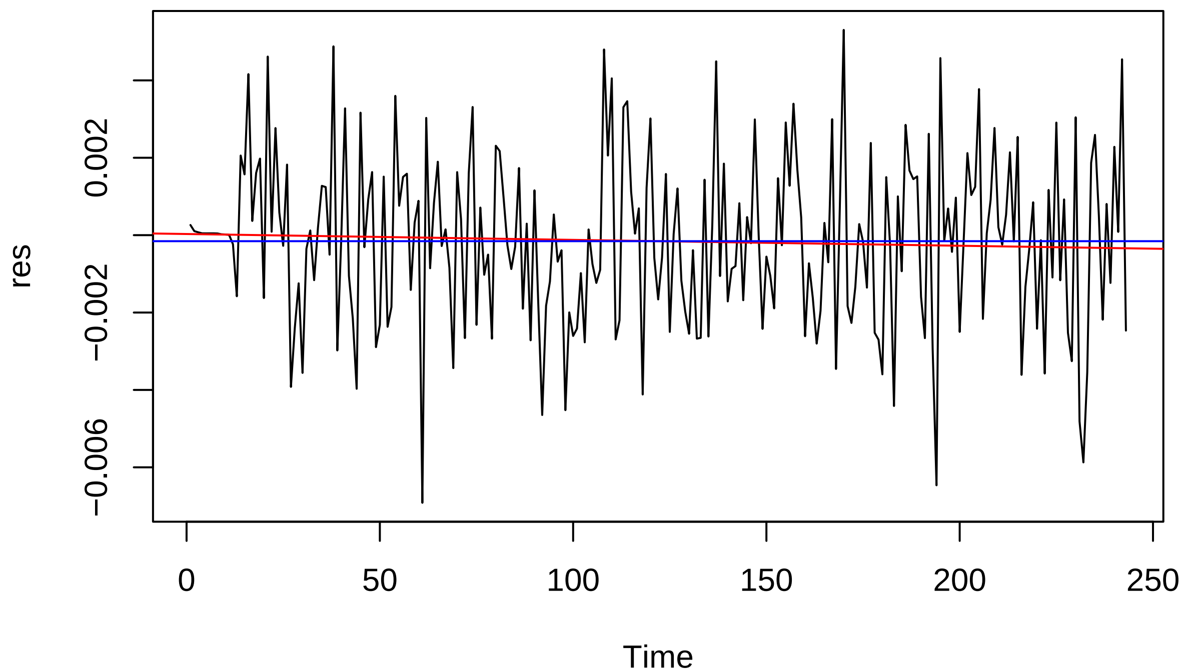
$$(1-0.2889B) \times (1+0.2373B^{12}) \times (1-B) \times (1-B^{12})X_t = (1-0.8478B) \times (1-0.5696B^{12}-0.4305B^{24})Z_t$$

Diagnostic checking

```
# Residual:
fit <- arima(electricity1.bc, order = c(2,1,1), seasonal = list(order = c(1,1,2), period = 12),
             fixed = c(NA,0,NA,NA,NA,NA),method="ML")
res <- residuals(fit)
hist(res,density=20,breaks=20, col="blue", xlab="", prob=TRUE)
m <- mean(res)
std <- sqrt(var(res))
curve( dnorm(x,m,std), add=TRUE )
```



```r
plot.ts(res)
fitt <- lm(res~as.numeric(1:length(res))); abline(fitt, col="red")
abline(h=mean(res), col="blue")
```




```r
var(res)
```

```
## [1] 4.850023e-06
```

```r
m
```

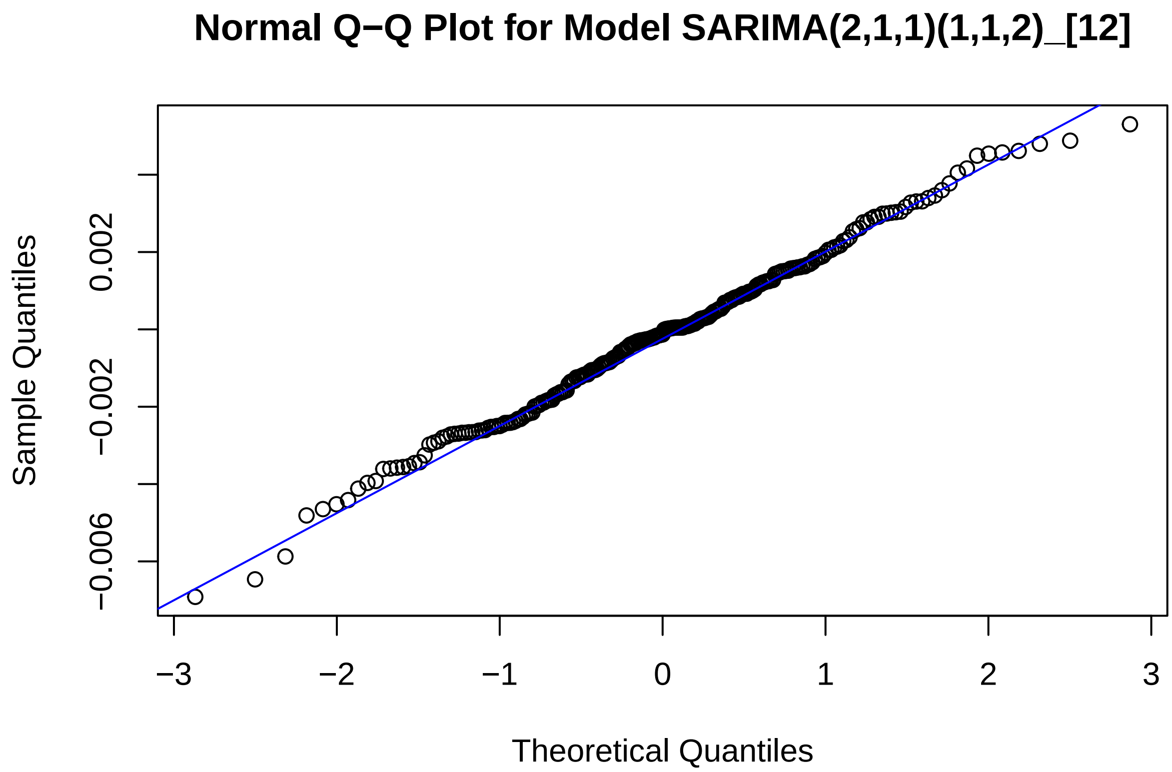
```
## [1] -0.0001558557
```

Plot residuals:

- resemble WN
- No trend, no seasonality, no visible change of variance
- Sample mean is almost zero: -0.0001558557

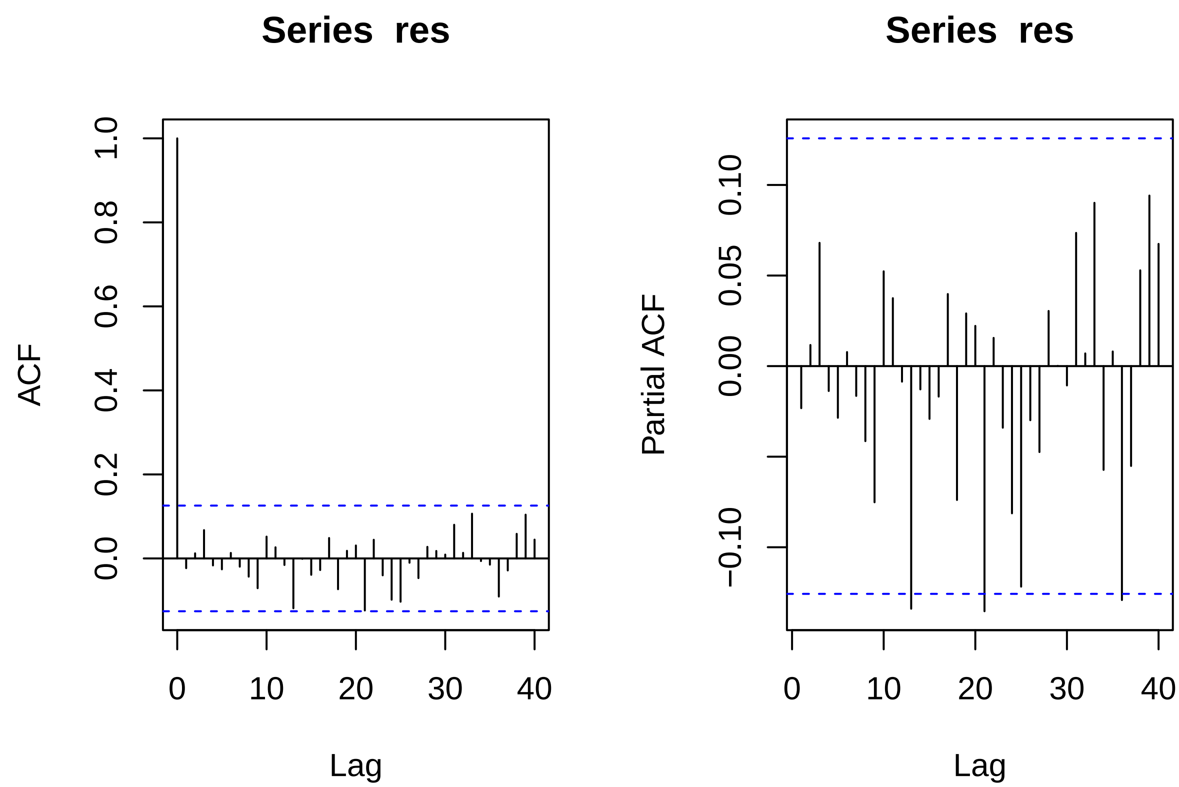
Plot a histogram of residuals: resemble Gaussian

```r
qqnorm(res,main= "Normal Q-Q Plot for Model SARIMA(2,1,1)(1,1,2)_[12]")
qqline(res,col="blue")
```



Normal Q-Q plot: close to straight line

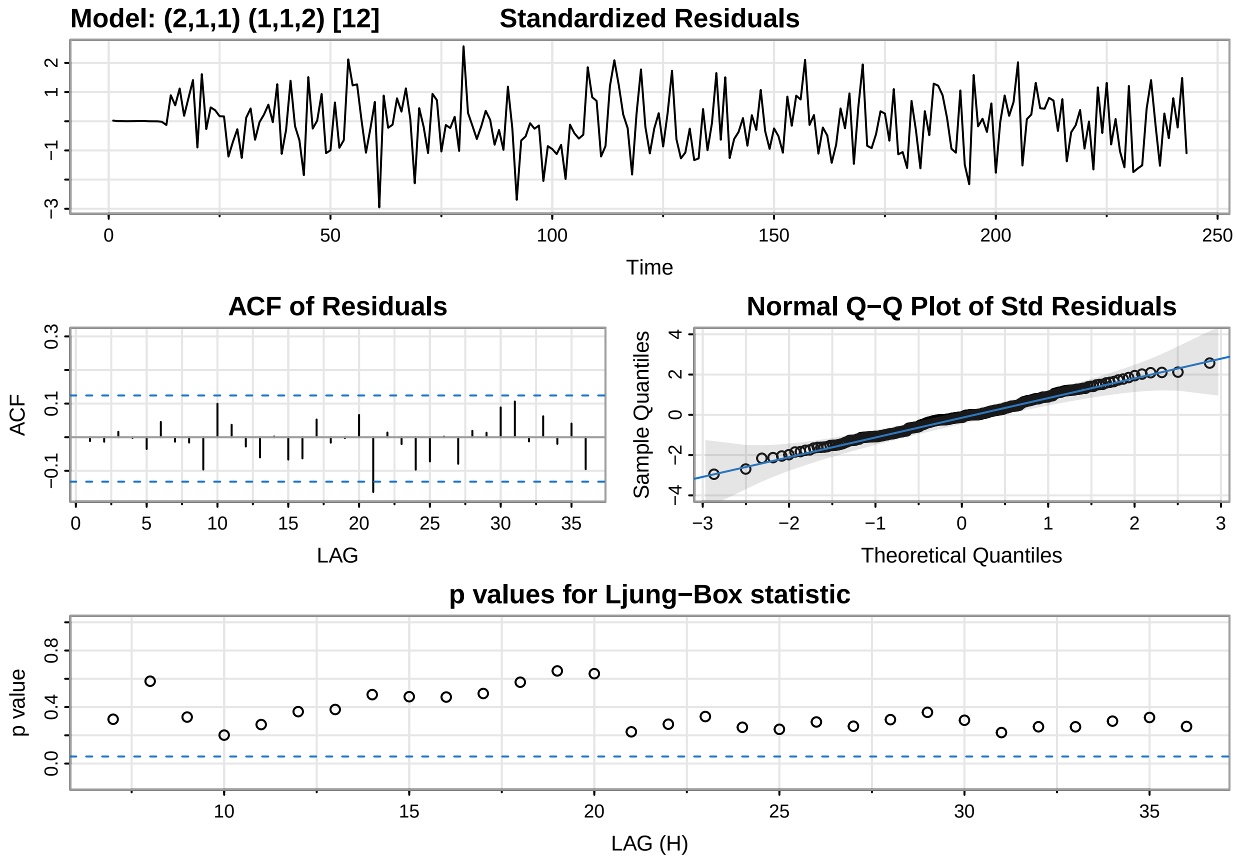
```
par(mfrow = c(1, 2))
acf(res, lag.max=40)
pacf(res, lag.max=40)
```




```
par(op)
```

All acf and pacf of residuals are within confidence intervals and can be counted as zeros.

```
fit.i <- sarima(xdata=electricity1, p=2, d=1, q=1, P=1, D=1, Q=2, S=12)
```



```r
# p-value should be bigger that 0.05
shapiro.test(res) # p-value should be bigger that 0.05
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.99464, p-value = 0.5489
```

```r
Box.test(res, lag = 16, type = c("Box-Pierce"), fitdf = 3)
```

```
##
##  Box-Pierce test
##
## data:  res
## X-squared = 8.1828, df = 13, p-value = 0.8315
```

```r
Box.test(res, lag = 16, type = c("Ljung-Box"), fitdf = 3)
```

```
##
##  Box-Ljung test
##
## data:  res
## X-squared = 8.6088, df = 13, p-value = 0.8018
```

```r
Box.test(res^2, lag = 16, type = c("Ljung-Box"), fitdf = 0)
```

```
##
##  Box-Ljung test
##
## data:  res^2
## X-squared = 15.206, df = 16, p-value = 0.5096
```

h = 16 because$\sqrt{243} \approx 16$ Box-Pierce and Ljung-Box fitdf = 3, because p+q=3 All p-value is larger than 0.05.

```r
ar(res, aic = TRUE,  order.max = NULL, method = c("yule-walker"))
```

```
##
## Call:
## ar(x = res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  4.85e-06
```

Fitted residual to AR(0), White noise **Pass Diagnostic checking.** $SARIMA(2,1,1) \times (1,1,2)_{12}$ ready to be used for forecasting.

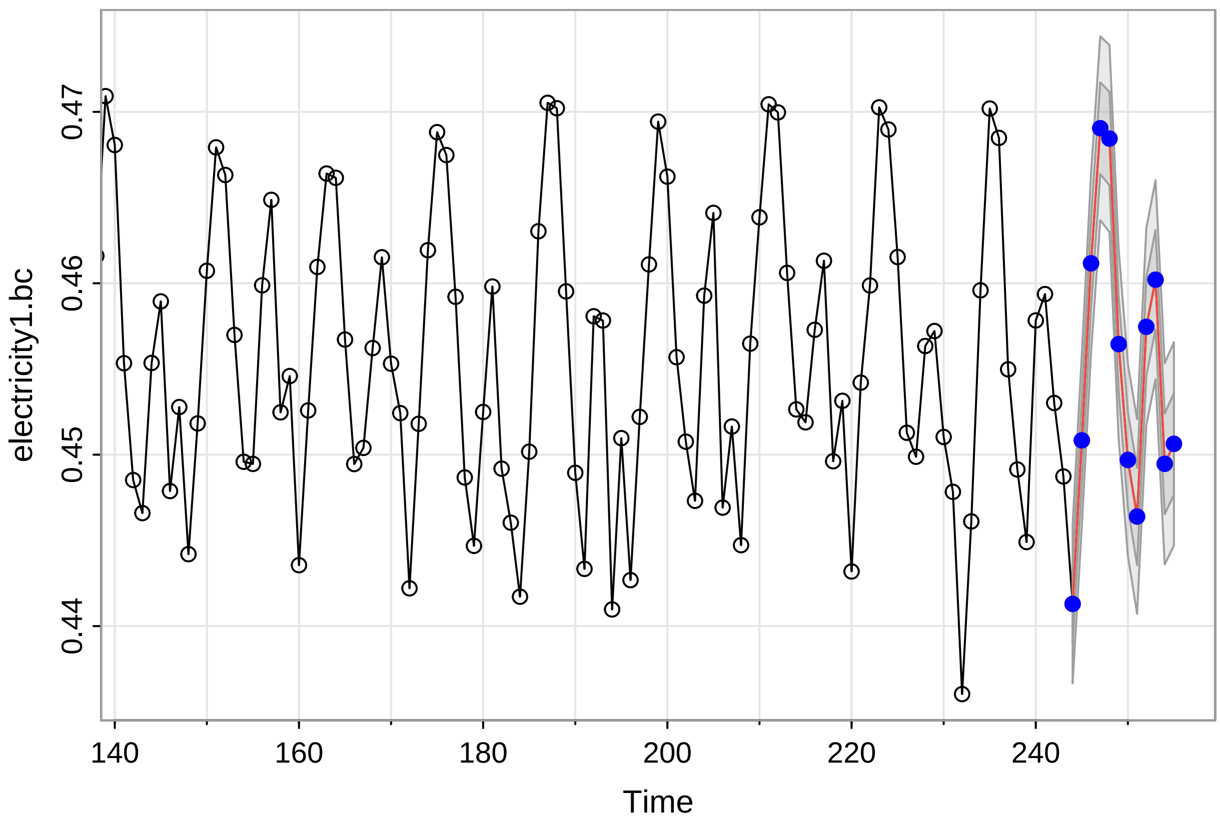
Forecasting Data

```r
library(forecast)
```

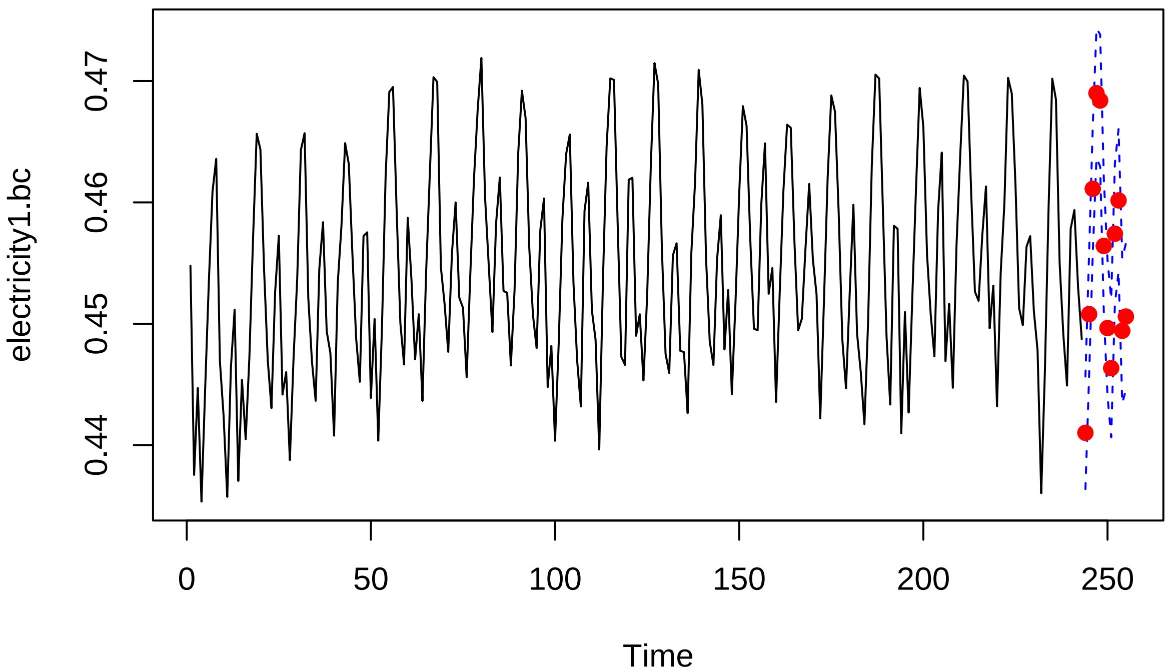
```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

## Registered S3 methods overwritten by 'forecast':
##   method                 from
##   autoplot.Arima         ggfortify
##   autoplot.acf           ggfortify
##   autoplot.ar            ggfortify
##   autoplot.bats          ggfortify
##   autoplot.decomposed.ts ggfortify
##   autoplot.ets           ggfortify
##   autoplot.forecast      ggfortify
##   autoplot.stl           ggfortify
##   autoplot.ts            ggfortify
##   fitted.ar              ggfortify
##   fortify.ts             ggfortify
##   residuals.ar           ggfortify

##
## Attaching package: 'forecast'

## The following object is masked from 'package:astsa':
##
##     gas
```

Forecast the transformed data

```r
pred.tr <- sarima.for(electricity1.bc,n.ahead = 12,p=2,d=1,q=1,P=1,D=1,Q=2, S=12)
points(length(electricity1) + 1:length(electricity1_test),pred.tr$pred, col="blue",pch = 19)
```

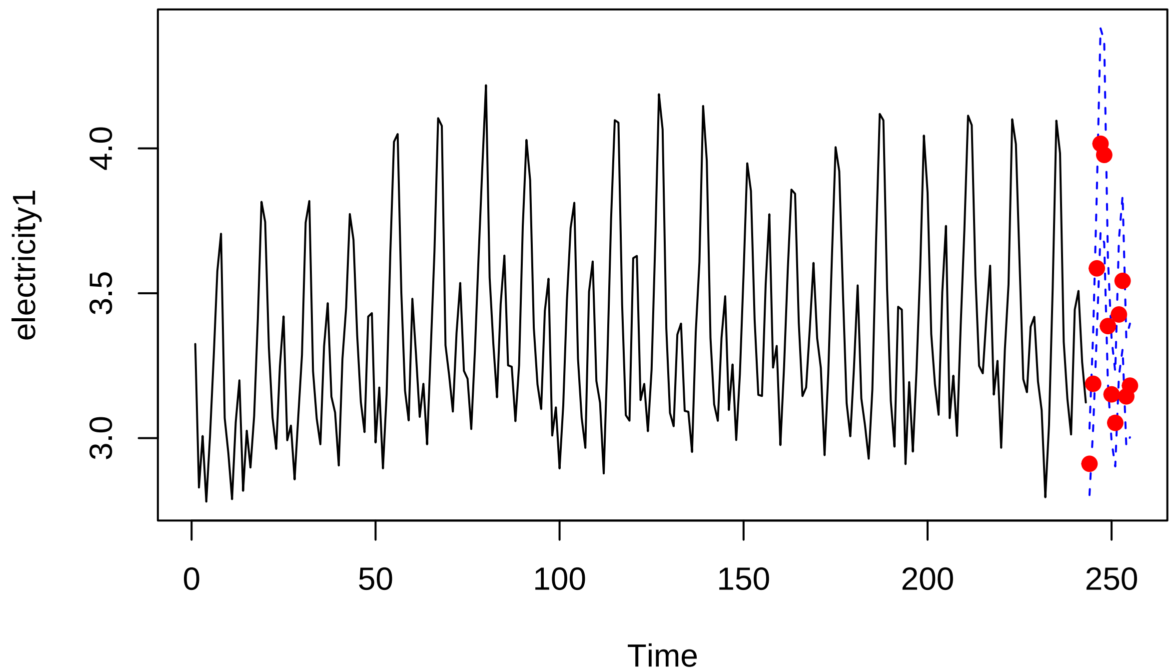



```
# Forecasting using model SARIMA(2,1,1)(1,1,2)_{12}:
fit.A <- arima(electricity1.bc, order = c(2,1,1), seasonal = list(order = c(1,1,2), period = 12),
               fixed = c(NA,0,NA,NA,NA,NA),method="ML")
forecast(fit.A)

# To produce graph with 12 forecasts on transformed data:
pred.tr1 <- predict(fit.A, n.ahead = 12)
U.tr = pred.tr1$pred + 2*pred.tr1$se # upper bound of the prediction interval
L.tr = pred.tr1$pred - 2*pred.tr1$se # lower bound
plot.ts(electricity1.bc, xlim=c(1,length(electricity1.bc)+12), ylim = c(min(electricity1.bc), max(U.tr))
lines(U.tr, col="blue",lty="dashed")
lines(L.tr, col="blue",lty="dashed")
points((length(electricity1.bc)+1):(length(electricity1.bc)+12), pred.tr1$pred, col="red",pch = 19)
```

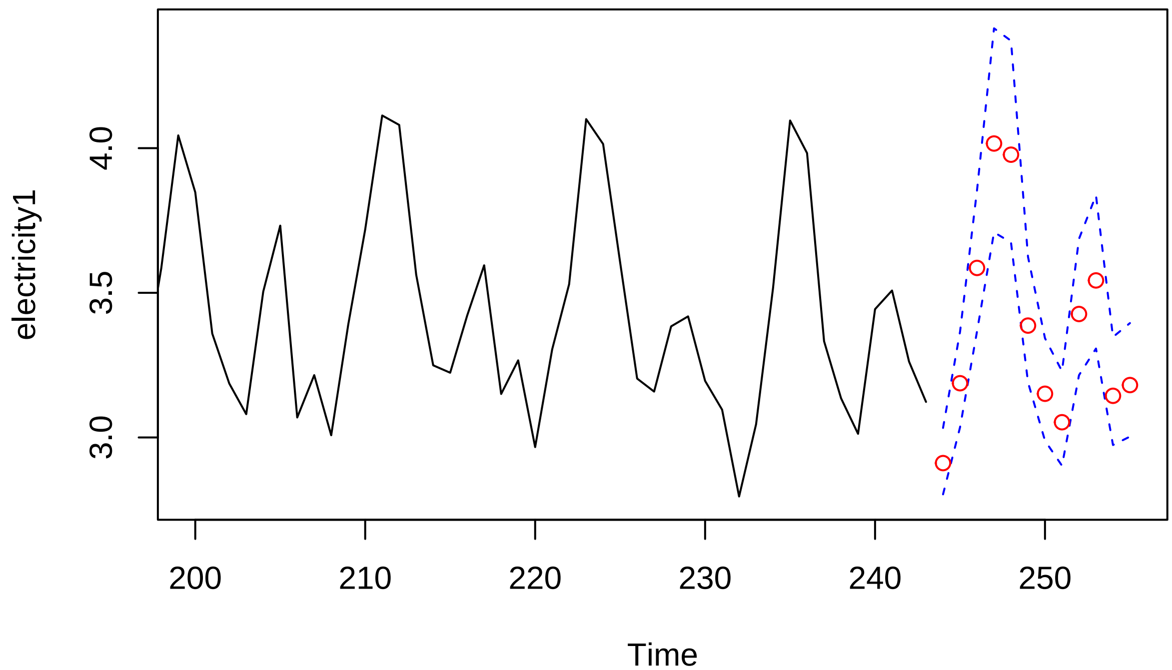


Forecasting original data

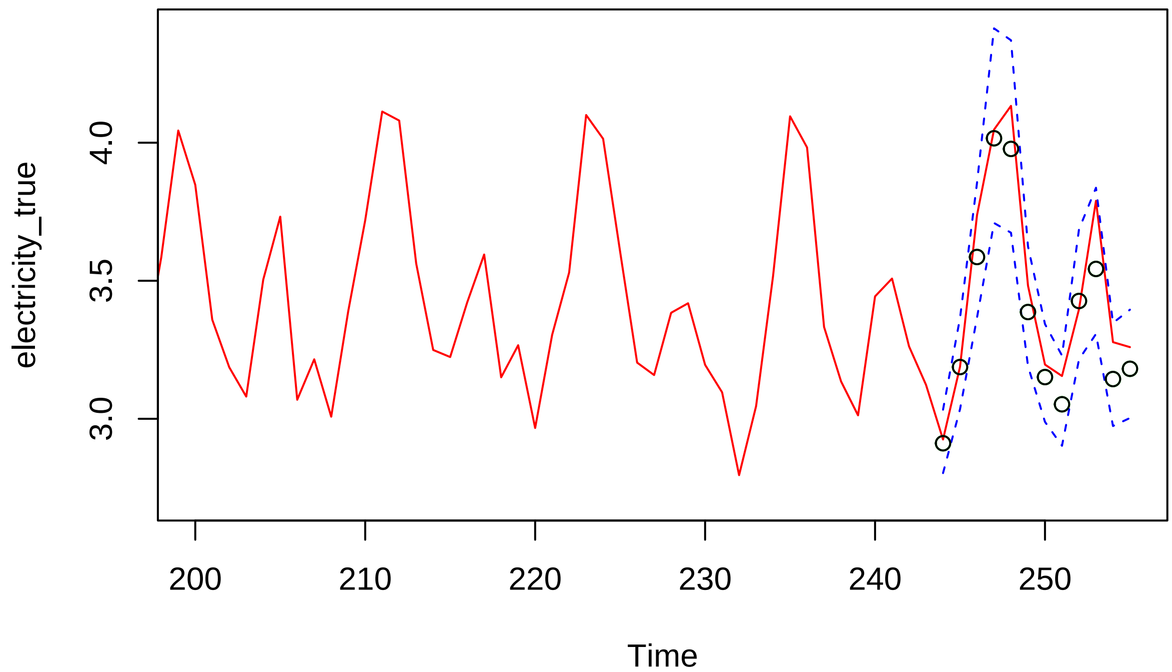
```
# To produce graph with forecasts on original data:
pred.orig <- InvBoxCox(pred.tr1$pred, lambda)
U= InvBoxCox(U.tr, lambda)
L= InvBoxCox(L.tr, lambda)
plot.ts(electricity1, xlim=c(1,length(electricity1)+12), ylim = c(min(electricity1),max(U)))
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(electricity1)+1):(length(electricity1)+12), pred.orig, col="red",pch = 19)
```




```r
# To zoom the graph, starting from entry 200
ts.plot(electricity1, xlim = c(200,length(electricity1)+12), ylim = c(min(electricity1),max(U)))
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(electricity1)+1):(length(electricity1)+12), pred.orig, col="red")
```




```
# To plot zoomed forecasts and true values (in electricity):
electricity_true <- electricity[1:255]/100000
plot.ts(electricity_true, xlim = c(200,length(electricity1)+12), ylim = c(2.7,max(U)), col="red")
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(electricity1)+1):(length(electricity1)+12), pred.orig, col="green")
points((length(electricity1)+1):(length(electricity1)+12), pred.orig, col="black")
```



Forcast:

- Black Circle, forecasting the original data using model $SARIMA(2,1,1) \times (1,1,2)_{12}$
- Red Line, the original data
- Test set is within prediction intervals

Conclude:

The project concluded that the selected model exhibited effectiveness in forecasting energy production dynamically in all sections of the United States. The forecasting result provides valuable information for energy policy-making, energy use assessment, decision-making for large enterprises, factories, or companieswith high demand for electricity, and energy companies. This project contributes to a better understandingof the United States energy production

---

Acknowledgments:

I would like to thank Professor Feldman for her support throughout the project. I spent a lot of time in her office hour asking about the quizzes and the final project.

Reference:

The U.S. Energy Information Administration (EIA)

APPENDIX

Appendix A: Data Preparation

```
electricity.csv <- read.table("electricity_data.csv", sep = ",",
                              header = FALSE, skip = 1, nrows = 255)
electricity <- ts(electricity.csv[, 2], start = c(2001, 1, 1), frequency=12)
electricity1 = electricity[c(1: 243)]/100000
electricity1_test = electricity[c(244: 255)]/100000
```

- read all 255 data
- electricity1: training dataset
- electricity1_test: test dataset

Appendix B: Data Examination

```
ts.plot(electricity/100000, main = "Raw Data")
ts.plot(electricity1, main="Monthly Electricity Generation in all sector of US",
        ylab="electricity1")
ele_fit <- lm(electricity1 ~ as.numeric(1:length(electricity1))); abline(ele_fit, col="red")
abline(h=mean(electricity1), col="blue")
```

Plotting raw data and training data.

```
hist(electricity1, main="Monthly Electricity Generation in all sector of US",
     xlab="Monthly electricity generation")
acf(electricity1)
```

plotting histogram and acf of the training data to confirm non-stationary by finding out if it's symmetric/bell shaped or periodic or not.

Appendix C: Comparing Transformation

```
library(MASS)
t <- 1:length(electricity1)
bcTransform <- boxcox(electricity1 ~ t, plotit=TRUE) # plotting the graph
bcTransform$x[which(bcTransform$y == max(bcTransform$y))] # get the value of lambda

lambda = bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
electricity1.bc = (1/lambda)*(electricity1^lambda-1)
electricity1.log = log(electricity1)
electricity1.sqrt = sqrt(electricity1)

op <- par(mfrow = c(2,2))
ts.plot(electricity1,main = "Original data")
ts.plot(electricity1.bc,main = "Box-Cox tranformed data")
ts.plot(electricity1.log, main = "Log Transform")
ts.plot(electricity1.sqrt, main = "Square Root Transform")
par(op)
```

```
op <- par(mfrow = c(2,2))
hist(electricity1, col = "light blue", xlab = "", main = "histogram U_t")
hist(electricity1.bc, col = "light blue", xlab = "", main = "histogram; bc(U_t)")
hist(electricity1.log,col = "light blue", xlab = "", main = "histogram; ln(U_t)")
hist(electricity1.sqrt,col = "light blue", xlab = "", main = "histogram; sqrt(U_t)")
par(op)
```

Transformation is improve the distributional properties and reduce variance. When comparing histogram through symmetric/bell-shaped distribution; comparing plots through variance.

```
# Checking if transformation is necessary
# Calculate the sample variance and plot the acf/pacf
var(electricity1)
var(electricity1.bc) # the variance before difference
```

Appendix D: Trend and Seasonality Analysis

Producing the decomposition of the Box-cox transformed data $bc(U_t)$ to find seasonality and trend.

Also, plot the acf/pacf of $bc(U_t)$

```
library(ggplot2)
#install.packages('ggfortify')
library(ggfortify)
y <- ts(as.ts(electricity1.bc), frequency = 12)
decomp <- decompose(y)
plot(decomp)
```

```
op = par(mfrow = c(1,2))
acf(electricity1.bc,lag.max = 40,main = "")
pacf(electricity1.bc,lag.max = 40,main = "")
title("Box-Cox Transformed Time Series", line = -1, outer=TRUE)
par(op)
```

Appendix E: Differencing $bc(U_t)$

Differencing $bc(U_t)$ at lag 1 to remove trend and then lag 12 to remove seasonality. Checking the variance during differencing making sure not to over/lack differencing.

```
y1 = diff(electricity1.bc, 1)
plot.ts(y1,main = "De-trended Time Series",ylab = expression(nabla~U[t]))
fit1 <- lm(y1 ~ as.numeric(1:length(y1)))
abline(fit1, col = "red")
mean(y1)
abline(h = mean(y1), col = "blue")
var(y1) # smaller that 7.342619e-5

# The upper trend is no longer apparent
# Seasonality still exists

y12 = diff(y1, 12)
plot.ts(y12, main = "De-trended/seasonalized Time Series", ylab =
          expression(nabla^{12}~nabla~U[t]))
```

```
fit1 <- lm(y12 ~ as.numeric(1:length(y12)))
abline(fit1, col = "red")
mean(y12)
abline(h = mean(y12), col = "blue")
var(y12) # smaller than 7.342619e-5 and 6.753134e-5

# No trend and seasonality
# Stationary behavior
```

Histogram of transformed and differenced data with normal curve:

```
# Compare histograms of Box-Cox (Ut) to the normal curve, really similar.
hist(y12, col="light blue", xlab="", main="histogram; bc(U_t) differenced at lags 12 & 1")
hist(y12, density=20,breaks=20, col="blue", xlab="", main = expression(nabla^{12}~nabla~U[t]),
     prob=TRUE)
m<-mean(y12)
std<- sqrt(var(y12))
curve( dnorm(x,m,std), add=TRUE )
```

Appendix F: Identifying SARIMA Models

$SARIMA(p, d, q) \times (P, D, Q)_s$ for the series bc(U_t)

ACF and PACF of Box-Cox U_t after differences at lag 1 and lag 12:

```
acf(y12,lag.max = 40,main = "")
title("ACF: First and Seasonally Differenced Time Series", line = -1, outer = TRUE)
pacf(y12,lag.max = 40,main = "")
title("PACF: First and Seasonally Differenced Time Series", line = -1, outer = TRUE)
```

The ACF and PACF plots of $\nabla_1 \nabla_{12} bc(U_t)$ provide information for selecting AR and MA component for the candidate SARIMA models.

Possible candidate models:
$MA(33)$
$SARIMA(2, 1, 0) \times (1, 1, 1)_{12}$
$SARIMA(2, 1, 1) \times (1, 1, 1)_{12}$
$SARIMA(2, 1, 0) \times (1, 1, 2)_{12}$
$SARIMA(2, 1, 1) \times (1, 1, 2)_{12}$

Appendix G: Evaluating Candidate Models

SMA models tried: Q=1, 2, q=0,1. Model producing the lowest AICc:

```
arima(electricity1.bc, order = c(0,1,1), seasonal = list(order = c(0,1,2),
                                       period = 12), method="ML")
# Calculating AICc
AICc(arima(electricity1.bc, order = c(0,1,1), seasonal = list(order =
                                     c(0,1,2), period = 12), method="ML"))
```

```
arima(electricity1.bc, order = c(0,1,0), seasonal = list(order = c(0,1,2),
                                        period = 12), method="ML")
# AICc
```

```r
AICc(arima(electricity1.bc, order = c(0,1,0), seasonal = list(order =
                                       c(0,1,2), period = 12), method="ML"))
```

```r
arima(electricity1.bc, order = c(0,1,1), seasonal = list(order = c(0,1,1),
                                        period = 12), method="ML")
# AICc
AICc(arima(electricity1.bc, order = c(0,1,1), seasonal = list(order =
                                       c(0,1,1), period = 12), method="ML"))
```

```r
arima(electricity1.bc, order = c(0,1,0), seasonal = list(order = c(0,1,1),
                                        period = 12), method="ML")
#AICc
AICc(arima(electricity1.bc, order = c(0,1,0), seasonal = list(order =
                                       c(0,1,2), period = 12), method="ML"))
```

SAR

```r
arima(electricity1.bc, order = c(2,1,0), seasonal = list(order = c(1,1,0),
                                      period = 12), method="ML")
AICc(arima(electricity1.bc, order = c(2,1,0), seasonal = list(order =
                                      c(1,1,0), period = 12), method="ML"))
```

SARIMA(2,1,1)(1,1,2)_s=12

```r
arima(electricity1.bc, order = c(2,1,1), seasonal = list(order = c(1,1,2),
                                      period = 12), method="ML")
AICc(arima(electricity1.bc, order = c(2,1,1), seasonal = list(order =
                                       c(1,1,2), period = 12), method="ML"))
```

Best fit model (smallest AICc)

```r
arima(electricity1.bc, order = c(2,1,1), seasonal = list(order = c(1,1,2), period = 12),
      fixed = c(NA,0,NA,NA,NA,NA),method="ML")
AICc(arima(electricity1.bc, order = c(2,1,1), seasonal = list(order = c(1,1,2), period = 12),
           fixed = c(NA,0,NA,NA,NA,NA),method="ML"))
```

MA(33)

```r
arima(electricity1.bc, order = c(0,0,33), seasonal = list(order = c(0,0,0), period = 12),
      method="ML")
AICc(arima(electricity1.bc, order = c(0,0,33), seasonal = list(order = c(0,0,0), period = 12),
           method="ML"))
```

second less AICc

```r
arima(electricity1.bc, order = c(2,1,1), seasonal = list(order = c(1,1,1), period = 12),
      method="ML")
AICc(arima(electricity1.bc, order = c(2,1,1), seasonal = list(order = c(1,1,1), period = 12),
           method="ML"))

arima(electricity1.bc, order = c(2,1,1), seasonal = list(order = c(1,1,1), period = 12),
      fixed = c(NA,0,NA,NA,NA),method="ML")
AICc(arima(electricity1.bc, order = c(2,1,1), seasonal = list(order = c(1,1,1), period = 12),
           fixed = c(NA,0,NA,NA,NA),method="ML"))
```

$SARIMA(2,1,1) \times (1,1,2)_{12}$ is the best fit model.

Model:

$$(1 - 0.2889B) \times (1 + 0.2373B^{12}) \times (1 - B) \times (1 - B^{12})X_t = (1 - 0.8478B) \times (1 - 0.5696B^{12} - 0.4305B^{24})Z_t$$

Appendix H: Fit Model

```
fit <- arima(electricity1.bc, order = c(2,1,1), seasonal = list(order = c(1,1,2), period = 12),
             fixed = c(NA,0,NA,NA,NA,NA),method="ML")
# Residual:
res <- residuals(fit)
mean(res)
var(res)
```

Appendix I: Diagnostic Checking

Histogram of residuals: resemble Gaussian

Plot residuals resemble WN

No trend, no seasonality, no visible change of variance

```
hist(res,density=20,breaks=20, col="blue", xlab="", prob=TRUE)
m <- mean(res)
std <- sqrt(var(res))
curve( dnorm(x,m,std), add=TRUE )
plot.ts(res)
fitt <- lm(res~as.numeric(1:length(res))); abline(fitt, col="red")
abline(h=mean(res), col="blue")
```

```
## QQ-plot close to straight line
qqnorm(res,main= "Normal Q-Q Plot for Model SARIMA(2,1,1)(1,1,2)_[12]")
qqline(res,col="blue")
```

```
# Check acf and pacf of residuals are within confidence intervals
par(mfrow = c(1, 2))
acf(res, lag.max=40)
pacf(res, lag.max=40)
par(op)
```

```
# p-value should be bigger that 0.05
shapiro.test(res) # p-value should be bigger that 0.05
Box.test(res, lag = 16, type = c("Box-Pierce"), fitdf = 3)
Box.test(res, lag = 16, type = c("Ljung-Box"), fitdf = 3)
Box.test(res^2, lag = 16, type = c("Ljung-Box"), fitdf = 0)
```

- h = 16 because$\sqrt{243} \approx 16$
- Box-Pierce and Ljung-Box fitdf = 3, because p+q=3, so df=13
- McLeod fitdf = 0, df=h=16

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

Use Yule-Walker estimation: should fit into AR(0)

Appendix J: Forecast Models

Forecasting transformed data

```
library(forecast)
# Forecast the transformed data
pred.tr <- sarima.for(electricity1.bc,n.ahead = 12,p=2,d=1,q=1,P=1,D=1,Q=2, S=12)
points(length(electricity1) + 1:length(electricity1_test),pred.tr$pred, col="blue",pch = 19)
```

```
# Forecasting using model SARIMA(2,1,1)(1,1,2)_{12}:
fit.A <- arima(electricity1.bc, order = c(2,1,1), seasonal = list(order = c(1,1,2), period = 12),
               fixed = c(NA,0,NA,NA,NA,NA),method="ML")
forecast(fit.A)

# To produce graph with 12 forecasts on transformed data:
pred.tr1 <- predict(fit.A, n.ahead = 12)
U.tr = pred.tr1$pred + 2*pred.tr1$se # upper bound of the prediction interval
L.tr = pred.tr1$pred - 2*pred.tr1$se # lower bound
plot.ts(electricity1.bc, xlim=c(1,length(electricity1.bc)+12),
        ylim = c(min(electricity1.bc), max(U.tr)))
lines(U.tr, col="blue",lty="dashed")
lines(L.tr, col="blue",lty="dashed")
points((length(electricity1.bc)+1):(length(electricity1.bc)+12),
       pred.tr1$pred, col="red",pch = 19)
```

Forecasting original data

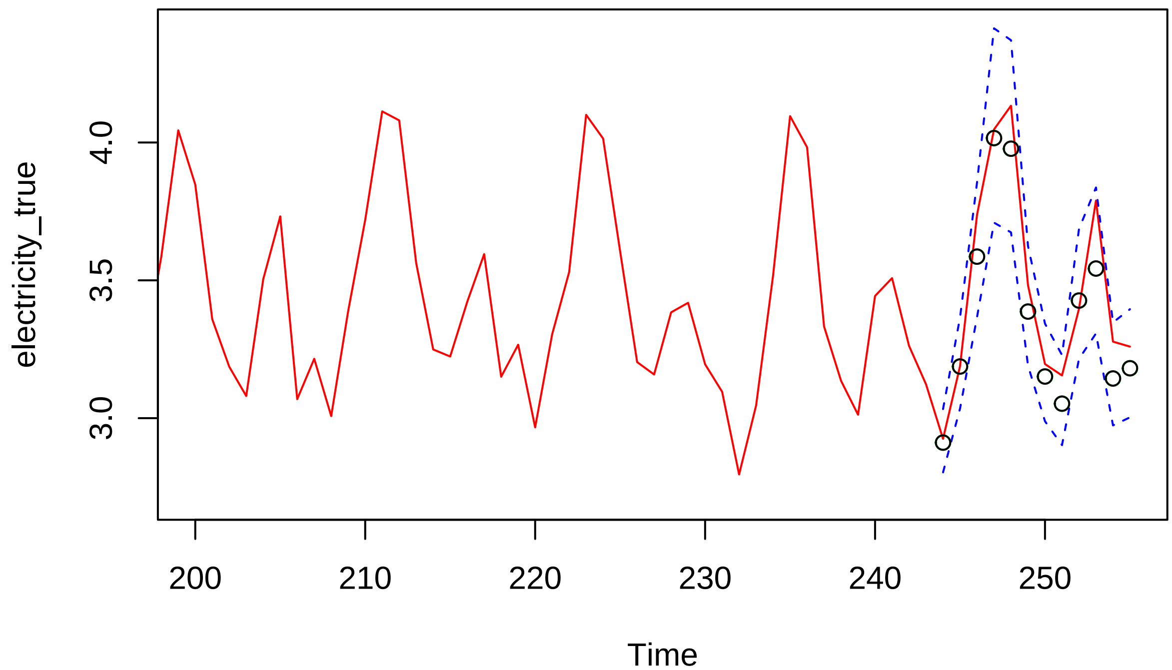
```
# To produce graph with forecasts on original data:
pred.orig <- InvBoxCox(pred.tr1$pred, lambda)
U= InvBoxCox(U.tr, lambda)
L= InvBoxCox(L.tr, lambda)
plot.ts(electricity1, xlim=c(1,length(electricity1)+12),
        ylim = c(min(electricity1),max(U)))
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(electricity1)+1):(length(electricity1)+12),
       pred.orig, col="red",pch = 19)
```

To zoom the graph, starting from entry 200

```
# To zoom the graph, starting from entry 200
ts.plot(electricity1, xlim = c(200,length(electricity1)+12),
        ylim = c(min(electricity1),max(U)))
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(electricity1)+1):(length(electricity1)+12),
       pred.orig, col="red")
```

Adding true value in the forecast plot

```
# To plot zoomed forecasts and true values (in electricity):
electricity_true <- electricity[1:255]/100000
plot.ts(electricity_true, xlim = c(200,length(electricity1)+12), ylim = c(2.7,max(U)), col="red")
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(electricity1)+1):(length(electricity1)+12), pred.orig, col="green")
points((length(electricity1)+1):(length(electricity1)+12), pred.orig, col="black")
```



We can see that the Test set is within prediction intervals.