PSTAT 174 Final Project

Yunning Chen

Abstract

This project revolves around studying and forecasting the number of employees in Total Nonfarm, given the attributes in the data set "All Employees: Total Nonfarm" provided by FRED, Federal Reserve Bank of St. Louis. More specifically, we will use the monthly and not seasonally adjusted data from 01/01/2010 to 12/01/2018 to predict the employment data from 01/01/2019 to 12/01/2019 monthly by using Box-Cox transformation and data differencing on original data. Then predict the SARIMA model based on the ACF/PACF of transformed and differenced data, estimate the coefficients of the model and perform diagnostic checking to verify our model. Finally, we successfully forecast the 12 months data in 2019 based on the model. The whole process was done in RStudio.

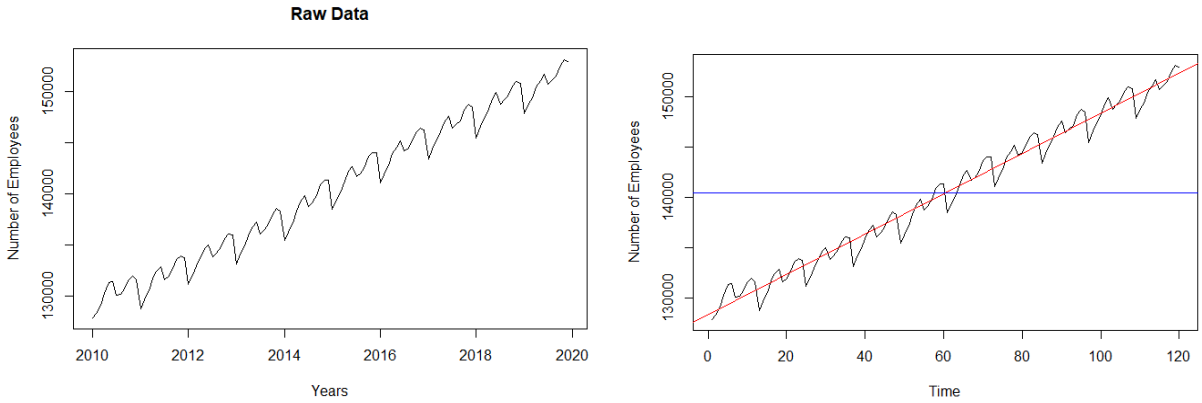
Introduction

In this project, we will study and forecast the number of employees in Total Nonfarm, which is a measure of the number of U.S. workers in the economy that excludes proprietors, private household employees, unpaid volunteers, farm employees, and the unincorporated self-employed. The dataset comes from U.S. Bureau of Labor Statistics, retrieved from FRED, Federal Reserve Bank of St. Louis. My motivation of studying this dataset is that I want to use all the time series techniques I learned in this class to get an insight of the US current economic situation, which can be directly observed in this measure since it shows increase or decrease of the number of jobs during a time period. And according to the website, this dataset contains the data of around 80% workers who contributes to Gross Domestic Product (GDP). So increase in employment indicates the growing of business and increase of individual disposable income.

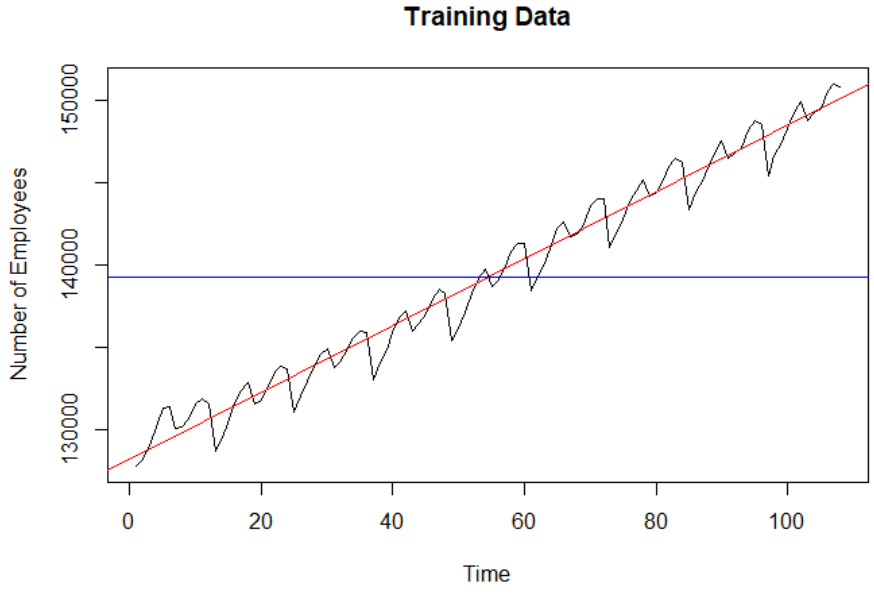
The data we forecasted is the employment data from 01/01/2019 to 12/01/2019, and we used the monthly and not seasonally adjusted data from 01/01/2010 to 12/01/2018 to find the approximate model. In order to do that, we first identify the main feature of the data plot from 01/01/2010 to 12/01/2018, then use necessary data transformation like Box-Cox transformation and log transformation, and choose the best transformation method to stabilize the variance of data. After that, we differenced the data to get a stationary series. Then, by analyzing the ACF and PACF, we identified the SARIMA models and estimate the coefficients of the model. After we confirm that the model is stationary and invertible, we performed diagnostic checking to check whether our model fits. Finally, we successfully used our finalized model to perform the forecasting.

Sections

First, since we only analyze the data from 01/01/2010 to 12/01/2018 and forecast the 12 following months. We only took data from 01/01/2010 to 12/01/2019 as the raw data. Below are the two plots of raw data.

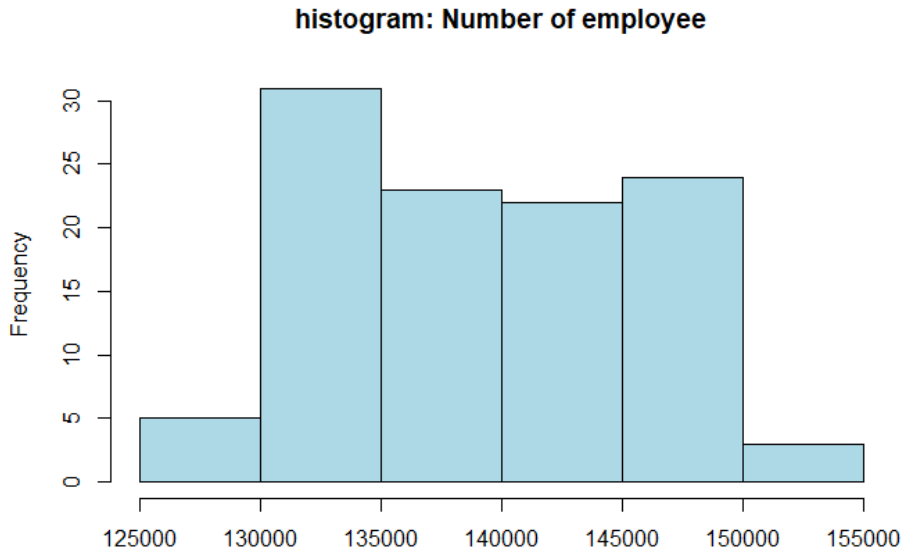


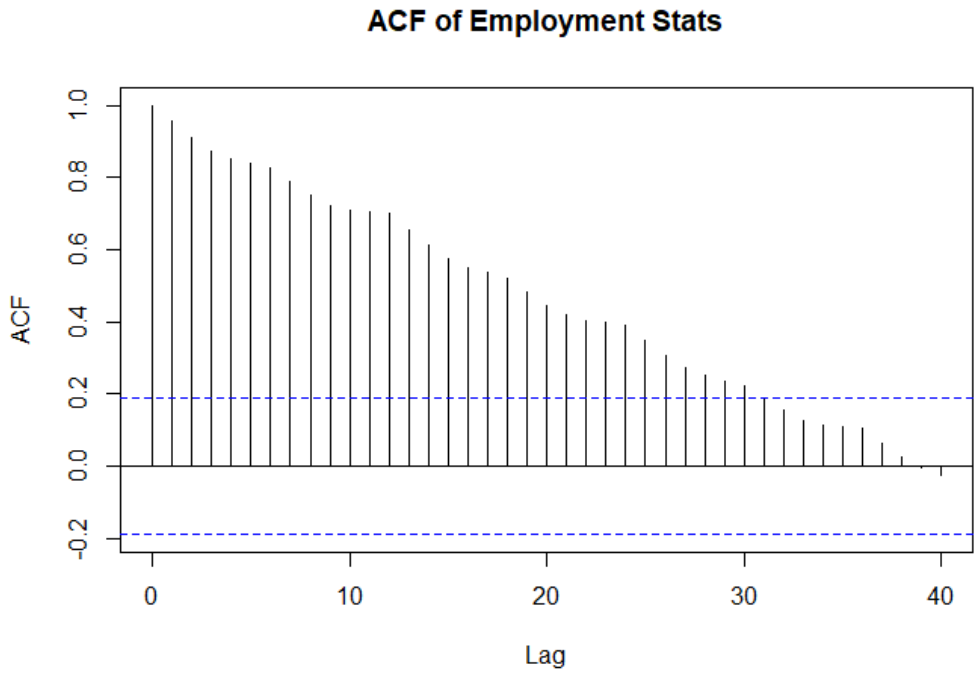
Since no new data is expected, we left 12 data points (the data in year 2019) for model validation. We named the training dataset "emp_train", which contains data from 01/01/2010 to 12/01/2018 and we will use it to build a model later. And we named the test dataset "emp_test", which contains the 12 months data in 2019. Then, we plotted the dataset emp_train, and we named it "Training Data". The plot is the one shown below.



According to the plot, we can immediately observe that the dataset is highly non-stationary as we can see there is an upward linear trend and seasonality in the data, and the plot does not have any apparent sharp changes in behavior. In addition, the dataset is

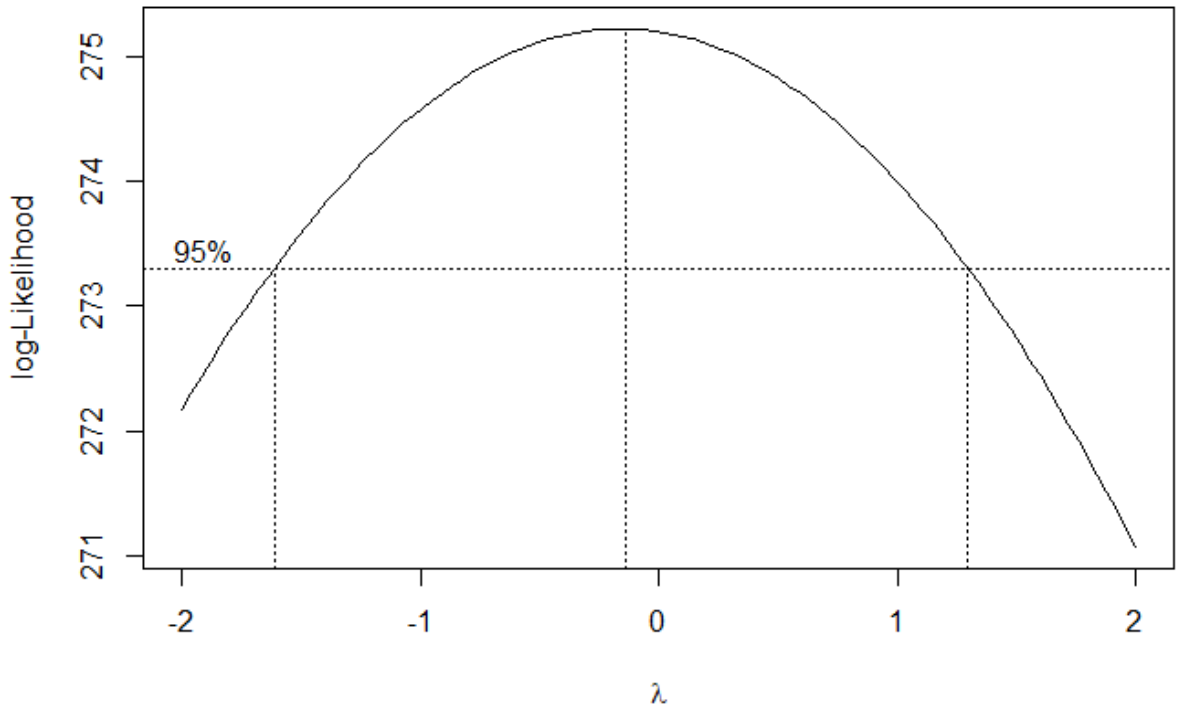
non-constant of variance and mean. In order to confirm non-stationarity of original data, we plotted the histogram and ACF of the training dataset below.



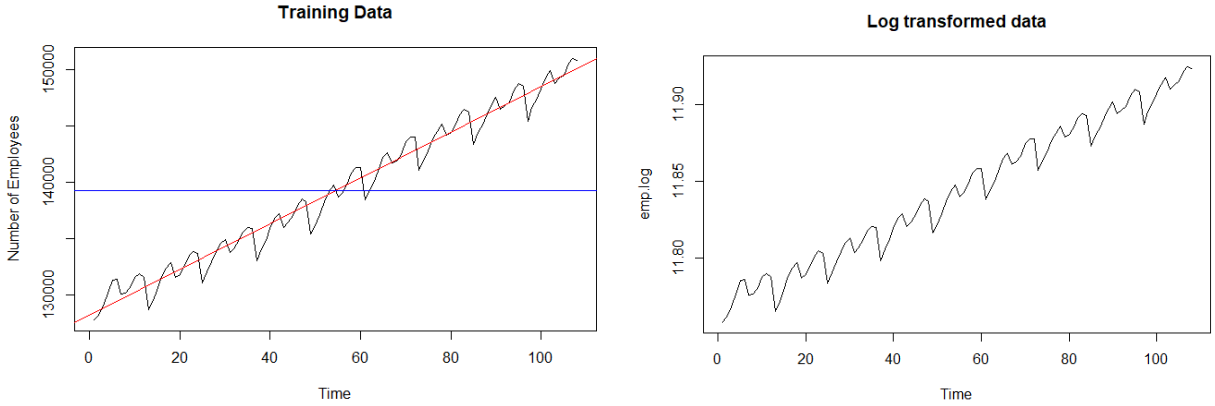


As we can see, the histogram is a little bit right skewed and the ACF remains large and have a little periodic behavior, which indicates seasonality and non-stationarity. Therefore, we need to use data transformation to stabilize variance. And we also need to difference the data to remove seasonality and trend.

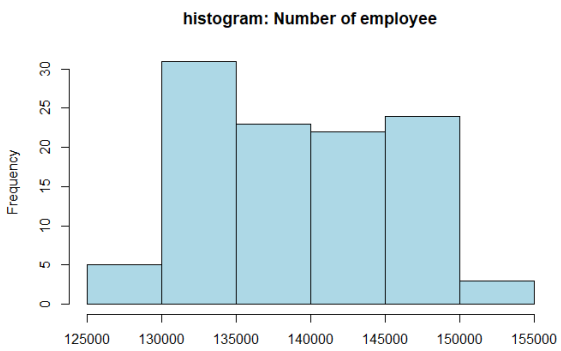
First, since the data is skewed and variance is non-constant, we used Box-Cox transformation to transform emp_train. Below is the graph.

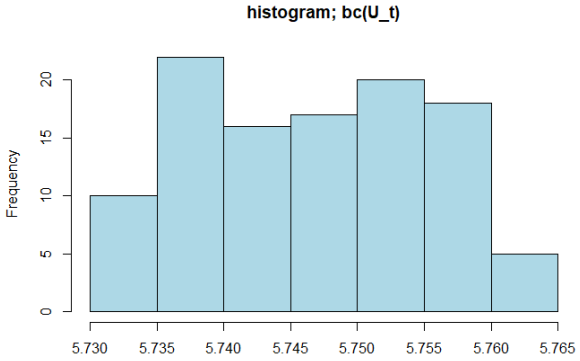


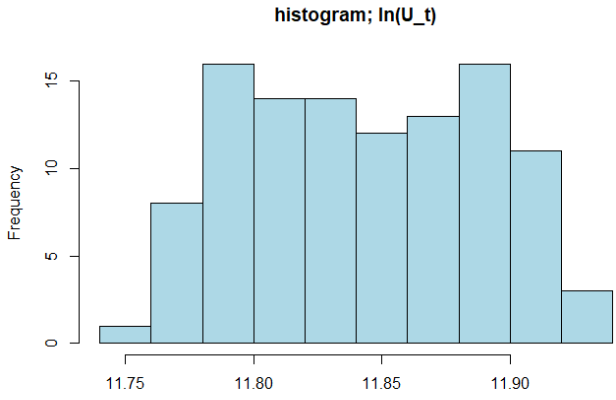
BcTransform command gives value $\boldsymbol{\lambda}$ = -0.1414141. Since $\boldsymbol{\lambda}$= 0 (log) is also in the confidence interval and is close to $\boldsymbol{\lambda}$ = -0.1414141, we will also try log transformation. Below are the plots of original dataset and data with both transformation:



As we can see, variance is more stable after transform, especially after Box-Cox transformation, according to the vertical scales. Next, we compared the histograms of original dataset and data with both transformations.

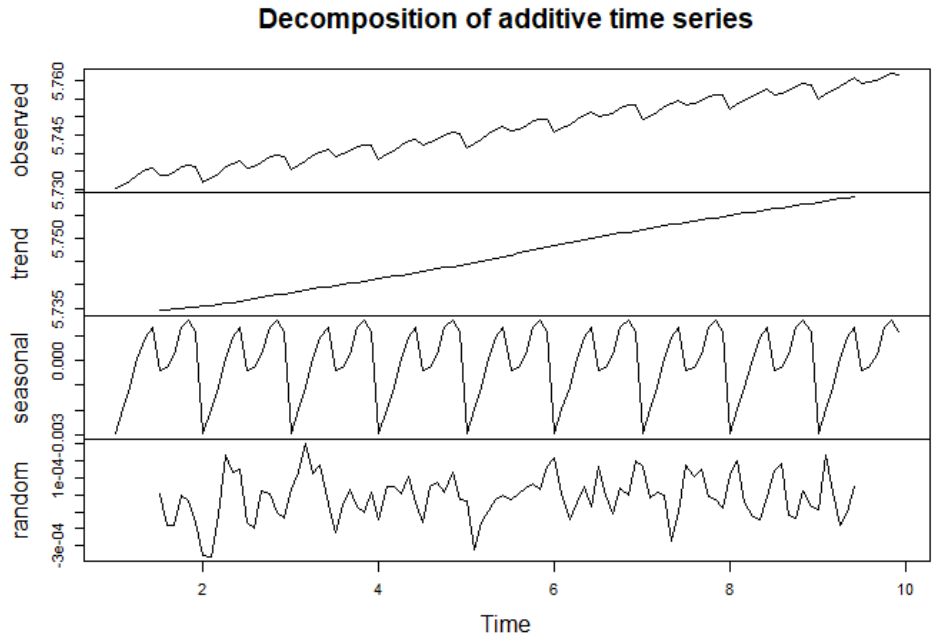




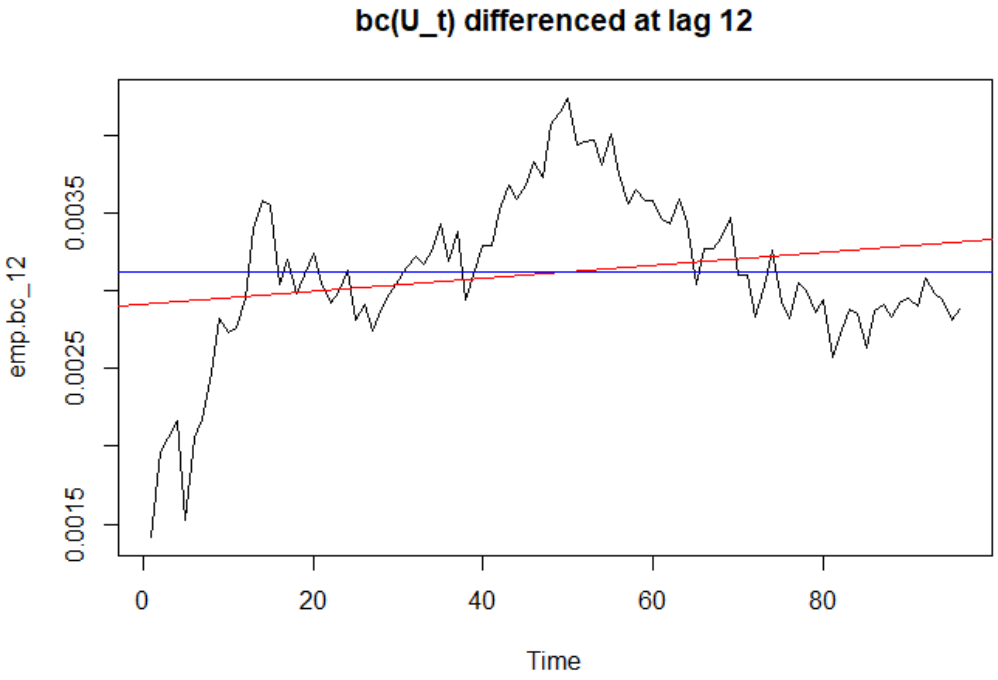


The histograms did not show much difference between transforms. Thus, we chose Box-Cox transformed data since this transformation gave a more stable and even variance.

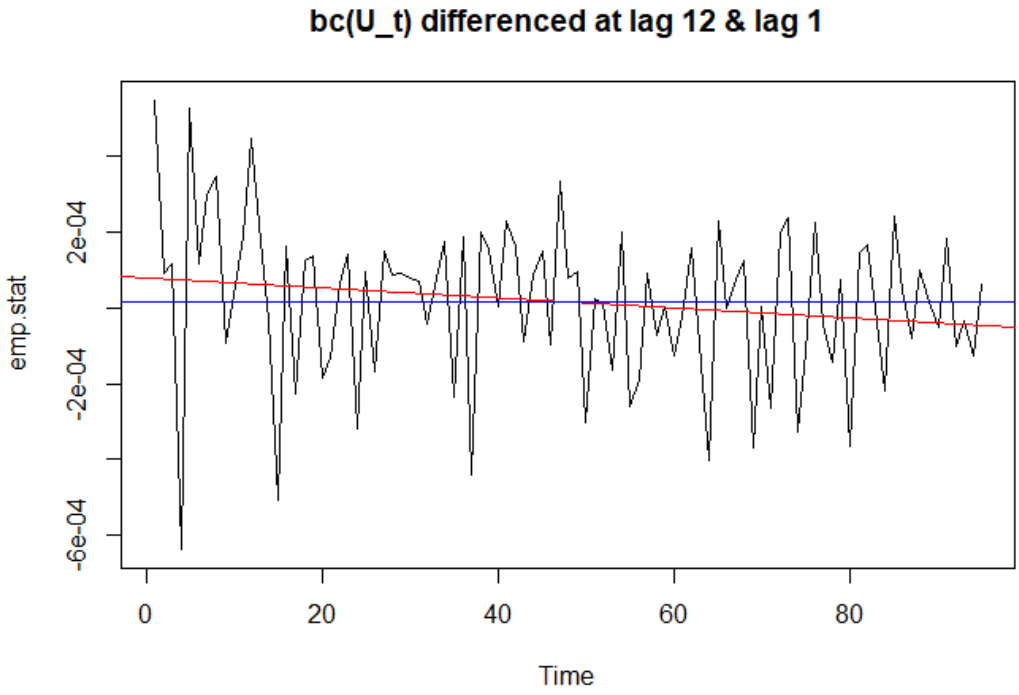
Next, we decomposed box-cox transformed data bc(U_t), here is the graph:



Decomposition of bc(U_t) shows seasonality and almost linear trend, therefore, we need to difference our data. First, we differenced bc(U_t) at lag 12 to remove seasonality.

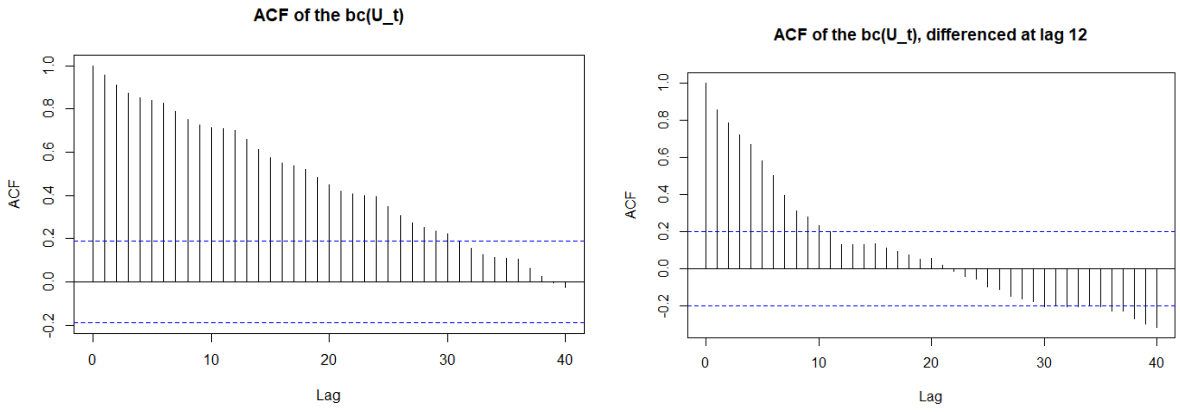


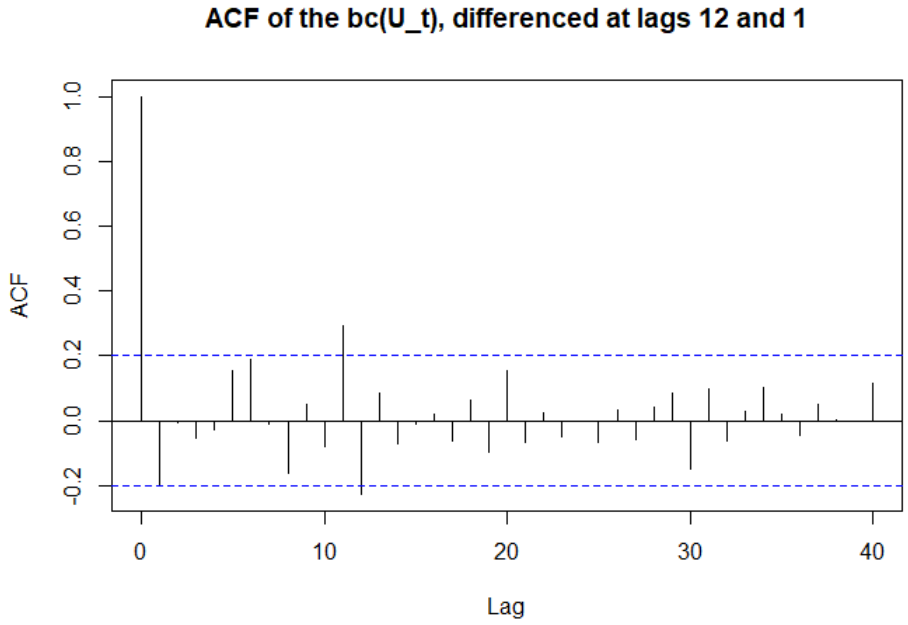
The variance of data after difference at lag 12 is 2.637454e-07, which is lower than the variance of original dataset: 41531786 and the variance after box-cox transformation: 7.509169-05. So we chose to use the differenced data. And the plot of bc(U_t) differenced at lag 12 showed that seasonality is no longer apparent, but trend is still here. Hence, we differenced bc(U_t) at lag 1 to remove the trend.



The variance after bc(U_t) differenced at lag 12 and lag 1 is 4.486463e-08, which is lower than the variance after difference at lag 12: 2.637454e-07. Thus we decided to use the data after difference at lag 12 and lag 1. And the plot of bc(U_t) differenced at lag 12 and lag 1 showed no seasonality. And we can see there is a small trend, but because the scale here is so small that it might be not a significant deviation. Hence we can ignore it.

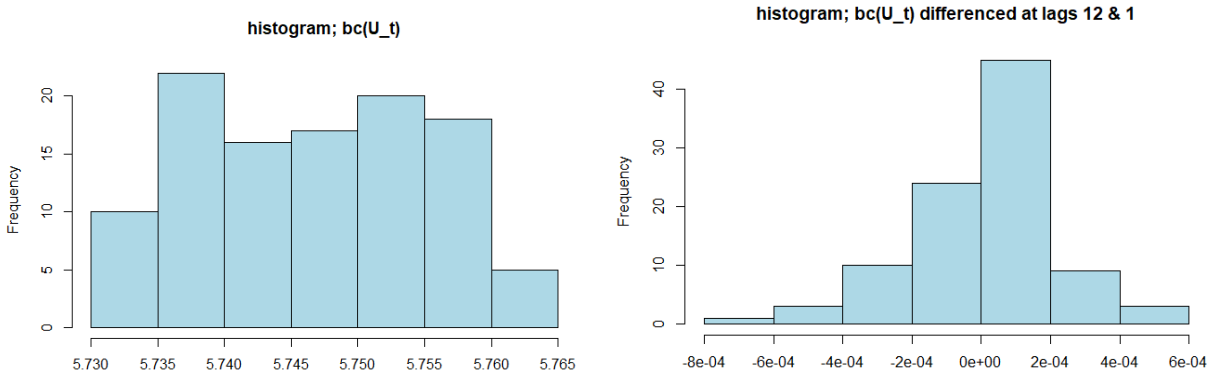
Then we compared the ACF of bc(U_t) and its differences.

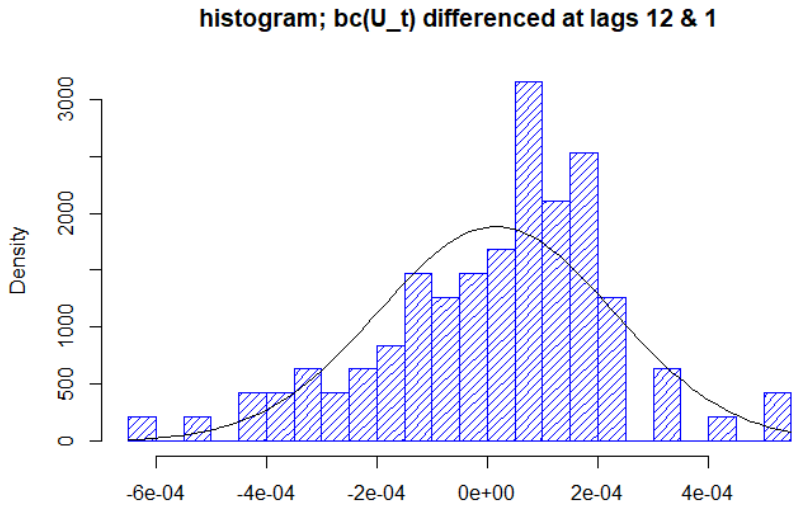




The plot of ACF of bc(U_t) decays slowly, which indicates non-stationary. And there is a slight periodic behavior, which shows seasonality. And the plot of ACF of bc(U_t) differenced at lag 12 showed that seasonality is no longer apparent, but the ACF decays slowly still indicates non-stationarity. For the plot of ACF of bc(U_t) differenced at lag 12 and lag 1, the ACF decay corresponds to a stationary process and no periodic behavior shown. Thus we concluded that we should work with data $\nabla_1 \nabla_{12}$ bc(U_t), where U_t is the first 108 observations of the original data, which correspond to the data from 01/01/2010 to 12/01/2018. We named $\nabla_1 \nabla_{12}$ bc(U_t) "emp.stat", and our series is now stationary.

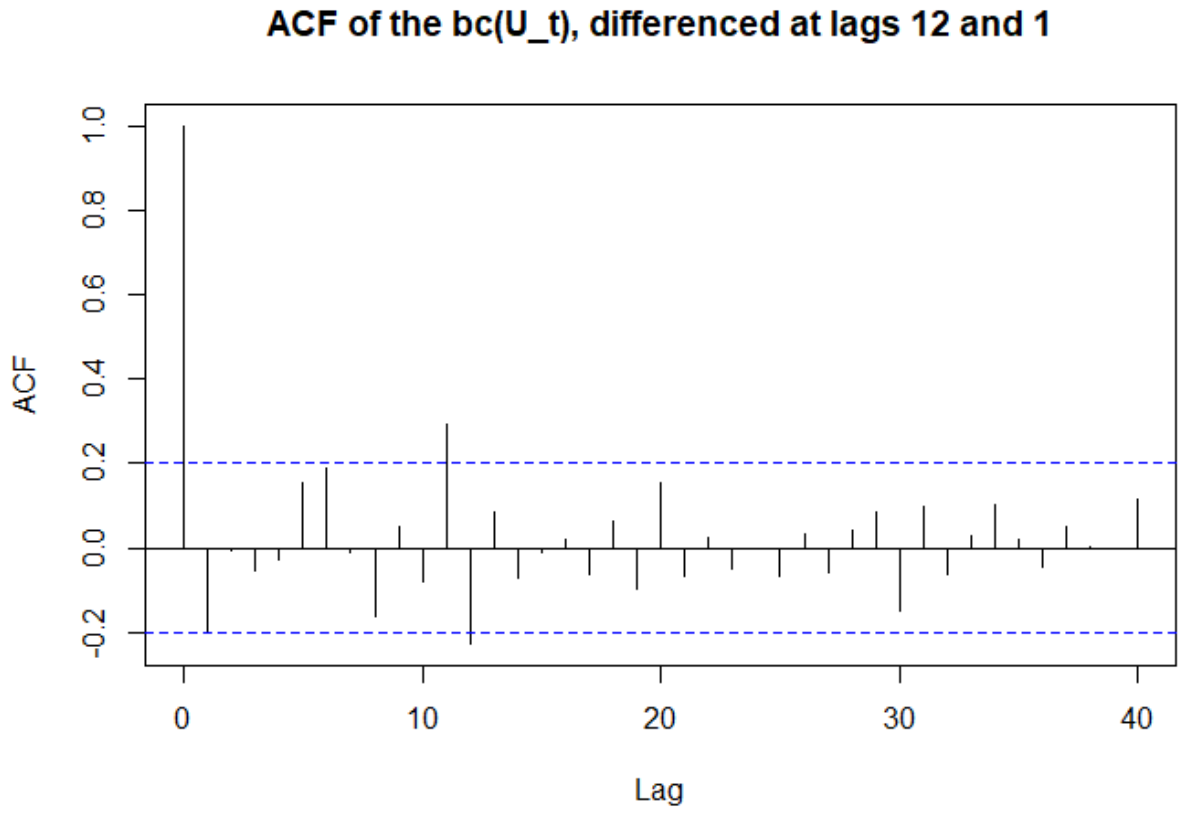
In addition, we compared the histogram of bc(U_t) below:

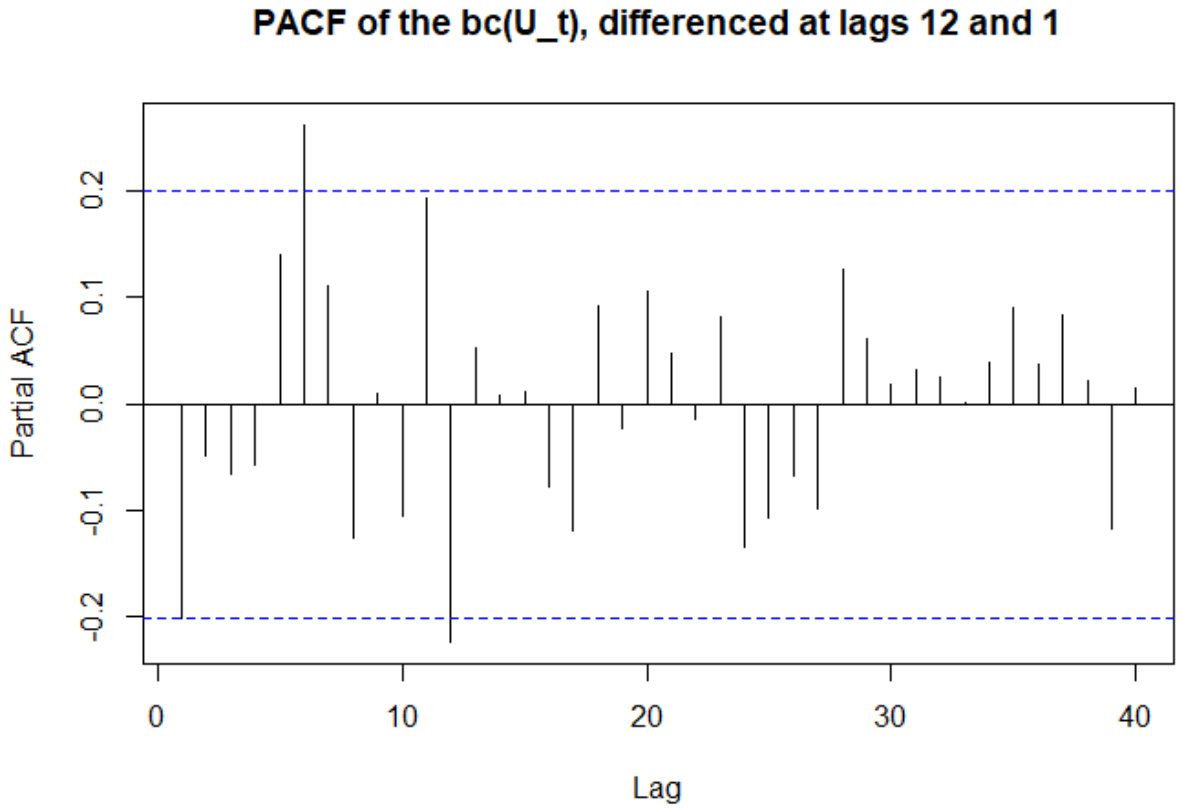




The histogram of $\nabla_1 \nabla_{12}$ bc(U_t) still has a tail but it looks more symmetric and more Gaussian than the histogram before differencing.

Next, we identified our model by analyzing ACF and PACF of $\nabla_1 \nabla_{12}$ bc(U_t). The plots of ACF and PACF are shown below:





The graph of ACF and PACF showed that ACF outside confidence intervals at lags 1(maybe), 11, 12, and PACF outside confidence intervals at lags 1(maybe), 6, 12. Accordingly, we have a list of candidate models for emp.stat ($\nabla_1 \nabla_{12}$ bc(U_t)) to try: SARIMA for bc(U_t): s=12, D=1, d=1, Q = 1, P = 1, q = 0 or 1 and p = 0 or 1 or 6.

For SMA model, we tried Q=1, q=0, 1, and neither of the models produce the lowest AICc value. Then we tried SAR models with P=1, p=0, 1, 6. And none of the SAR models produced the lowest AICc value either. Finally, we tried SARIMA model with P=Q=1, p = 0, 1, 6, q = 0, 1. And the model that produced the lowest AICc value is SARIMA(6,1,0)×$(1,1,1)_{12}$, which correspond to the model ACF and PACF plot suggested. The model and its coefficients are shown below:

```
> sarima(emp.bc, p=6,d=1,q=0,P=1,D=1,Q=1,S=12, details = FALSE)

Coefficients:
          ar1      ar2      ar3      ar4     ar5     ar6     sar1     sma1
      -0.1294  -0.1065  -0.1081  -0.0408  0.1637  0.4227  -0.0221  -0.5576
s.e.   0.1043   0.0963   0.0971   0.1025  0.1035  0.1124   0.2604   0.2392
```

with the lowest AICc value -12.72683.

From above, we observed that the coefficient of ar4 and sar1 are both below 0.05, which indicates non-significant. And the coefficients of ar1, ar2, ar3 and ar5 are all relatively low comparing to coefficient of ar6 and sma1, and all of their confidence interval contains a

non-significant value except for ar5. Hence, we tried to fix some of the coefficients and set the other coefficients to 0. And the two final model that produce the lowest and the second lowest AICc value are both SARIMA(6,1,0)×$(0,1,1)_{12}$ model with their coefficients shown below:

(1)

```
> sarima(emp.bc, p=6,d=1,q=0,P=0,D=1,Q=1,S=12, details = FALSE,
+        fixed = c(NA,0,0,0,0,NA,NA))

Coefficients:
          ar1  ar2  ar3  ar4  ar5     ar6     sma1
      -0.1514    0    0    0    0  0.4280  -0.5243
s.e.   0.1008    0    0    0    0  0.1146   0.1198
```

with the lowest AICc value -12.78164.

(2)

```
> sarima(emp.bc, p=6,d=1,q=0,P=0,D=1,Q=1,S=12, details = FALSE,
+        fixed = c(0,0,0,0,0,NA,NA))

Coefficients:
      ar1  ar2  ar3  ar4  ar5     ar6     sma1
        0    0    0    0    0  0.4158  -0.5697
s.e.    0    0    0    0    0  0.1162   0.1105
```

with the second lowest AICc value -12.78042.

Thus, we fit model A to be:

$$(1+0.1514_{(0.1008)}B-0.4280_{(0.1146)}B^6)(1-B)(1-B^{12})X_t = (1-0.5243_{(0.1198)}B^{12})Z_t,$$

where $Z_t \sim WN(0,\ 3.298e\text{-}08)$

and model B to be:

$$(1-0.4158_{(0.1162)}B^6)(1-B)(1-B^{12})X_t = (1-0.5697_{(0.1105)}B^{12})Z_t\ ,$$

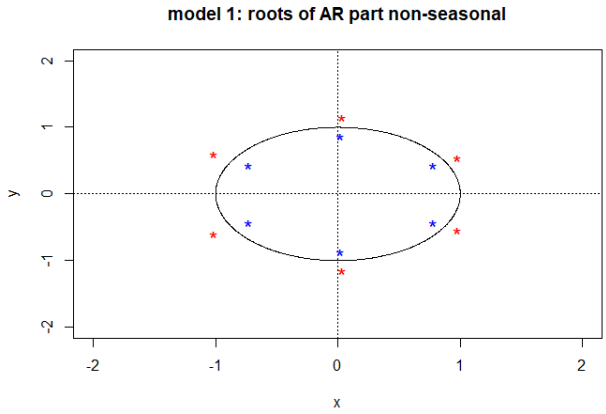
where $Z_t \sim WN(0,\ 3.352e\text{-}08)$

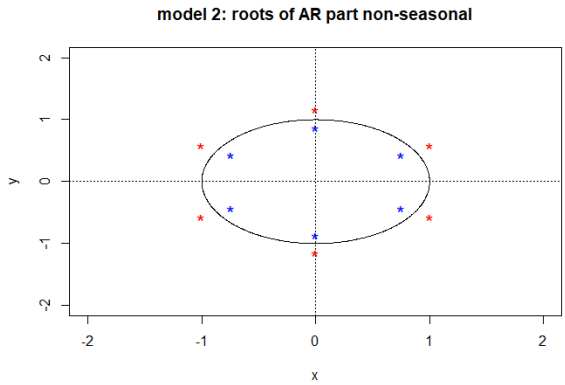
After fitting our models, we checked the stationarity of AR part in both models, and invertibility of MA part in both models. Since $|\boldsymbol{\Theta}_1|$ in both models are smaller than 1, both models are invertible. To check stationarity, we ran the following code to check if they have unit roots, the plots are below the codes:

```
> plot.roots(NULL,polyroot(c(1, -0.1514, 0, 0, 0, 0, 0.4280)),
+            main="model 1: roots of AR part non-seasonal")
```

and

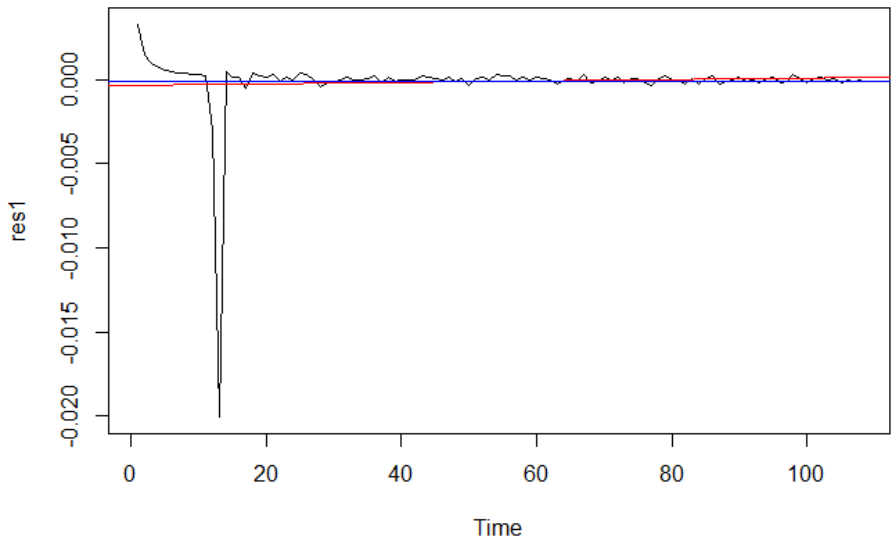
```
> plot.roots(NULL,polyroot(c(1,0, 0, 0, 0, 0, 0.4158)),
+            main="model 2: roots of AR part non-seasonal")
```



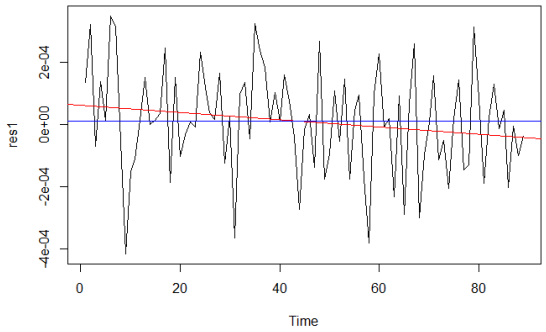


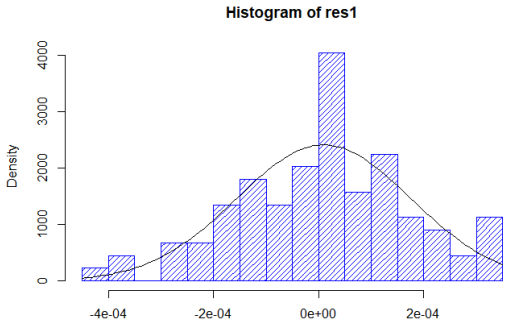
Based on the plots, for both models, all the roots are outside the unit circle. Therefore, we conclude that both models are stationary and invertible. Thus, our models are ready for diagnostic checking.

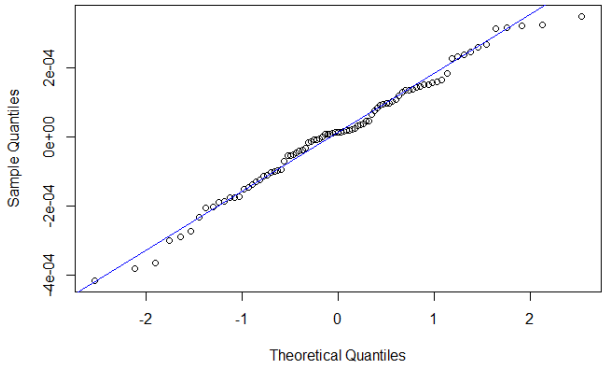
We first perform diagnostic checking of model A. For residuals of $\nabla_1\nabla_{12}\text{bc}(U_t)$, model A, we have the plot of residuals below:



As we can see, there is an extreme outlier before data point 20. But because we could not find the cause of the outlier since the original data plot and transformed data plot look fine, we opted out the residuals before time 20 and only took residuals start from time 20. And the residuals have the following plot:

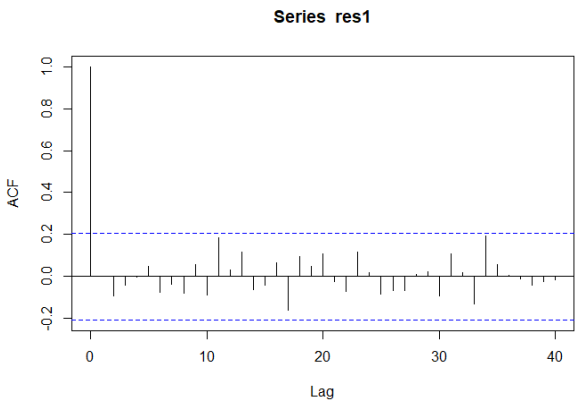


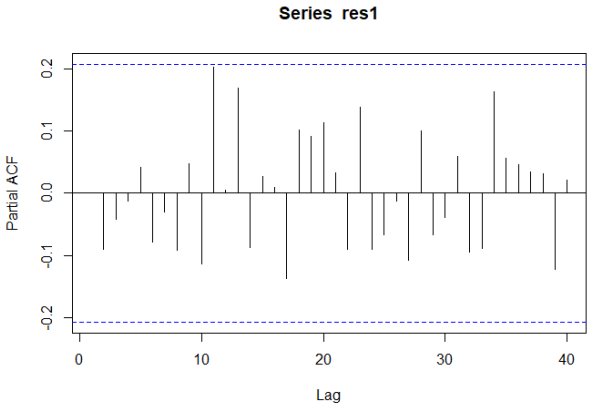




Based on the three plots above, we observed no seasonality in residuals. There was a small trend in the residuals plot but it is not a significant deviation due to the really small scale. The mean of residuals is 9.596777e-06, which is pretty close to 0. The histogram looks almost Gaussian but with a small tail, so we will use Shapiro-Wilk test to confirm the normality later. But overall, the histogram and the Q-Q plot looks fine.

Then, we need to check sample ACF and PACF of residuals, which is supposed to resemble white noise. The ACF and PACF plots are shown below:





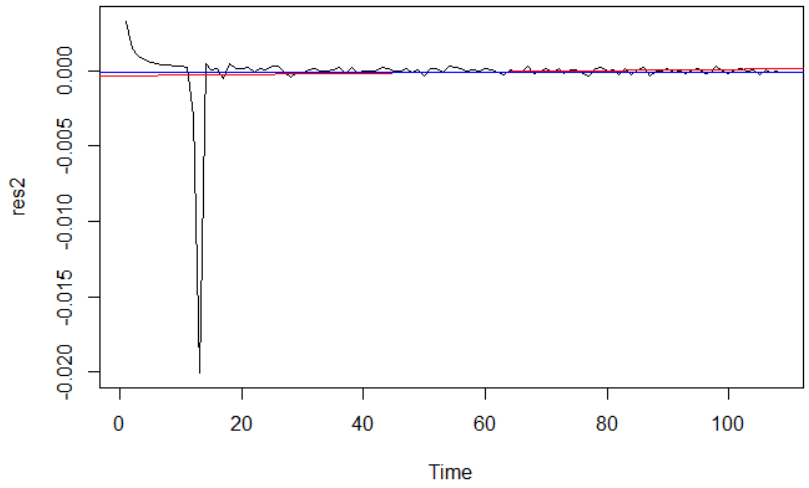
As we can see, all acf and pacf of residuals are within confidence intervals and can be counted as zeros. So model A passed ACF/PACF test.

After ACF/PACF test, we ran Shapiro-Wilk test of normality and Portmanteau tests. And we used the Yule-Walker estimation. The codes and results are shown below:

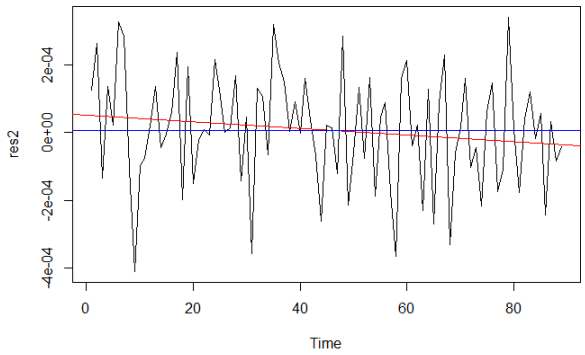
```
> shapiro.test(res1)

        Shapiro-Wilk normality test

data:  res1
W = 0.98666, p-value = 0.5004

> Box.test(res1, lag = 11, type = c("Box-Pierce"), fitdf = 3)

        Box-Pierce test

data:  res1
X-squared = 6.307, df = 8, p-value = 0.6129

> Box.test(res1, lag = 11, type = c("Ljung-Box"), fitdf = 3)

        Box-Ljung test

data:  res1
X-squared = 7.1516, df = 8, p-value = 0.5204

> Box.test(res1^2, lag = 11, type = c("Ljung-Box"), fitdf = 0)

        Box-Ljung test

data:  res1^2
X-squared = 7.5248, df = 11, p-value = 0.7551

> ar(res1, aic = TRUE, order.max = NULL, method = c("yule-walker"))

Call:
ar(x = res1, aic = TRUE, order.max = NULL, method = c("yule-walker"))


Order selected 0  sigma^2 estimated as  2.743e-08
```

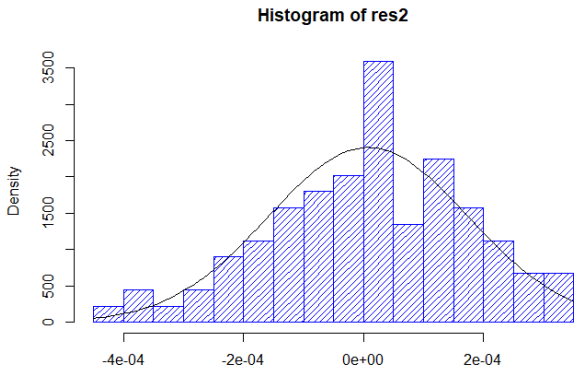
The p-value for Shapiro-Wilk normality test is 0.5004 > 0.05, so we accept that the residuals are normally distributed. Moreover, the p-value of Box-Pierce test is 0.6129 > 0.05, the p-value of Ljung-Box test is 0.5204 > 0.05, and the p-value of McLeod-Li test is 0.7551 > 0.05. Therefore, we accept that the residuals are independently distributed. Finally, the Yule-Walker estimation fit the residuals into AR(0) model. Thus, model A passed diagnostic checking and ready to be used for forecasting.

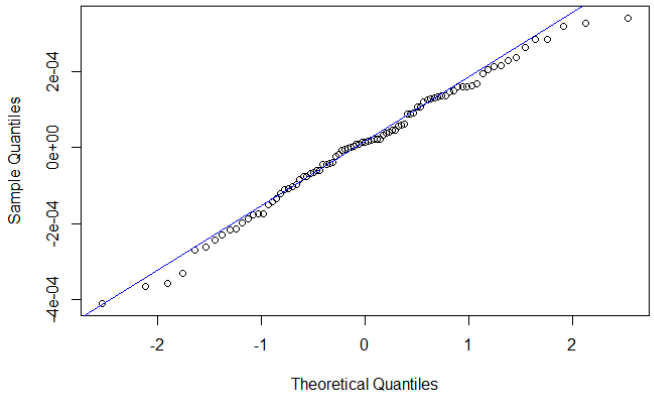
Now we need to perform diagnostic checking for model B. For residuals of $\nabla_1\nabla_{12}\text{bc}(U_t)$, model B, we have the plot of residuals below:



Like residuals in model A, there was an extreme outlier in the residuals. And we could not find out what caused the outlier. So we took residuals only after time 20, like what we did in model A. Below are the plots, histogram and Q-Q plot of residuals in model B:

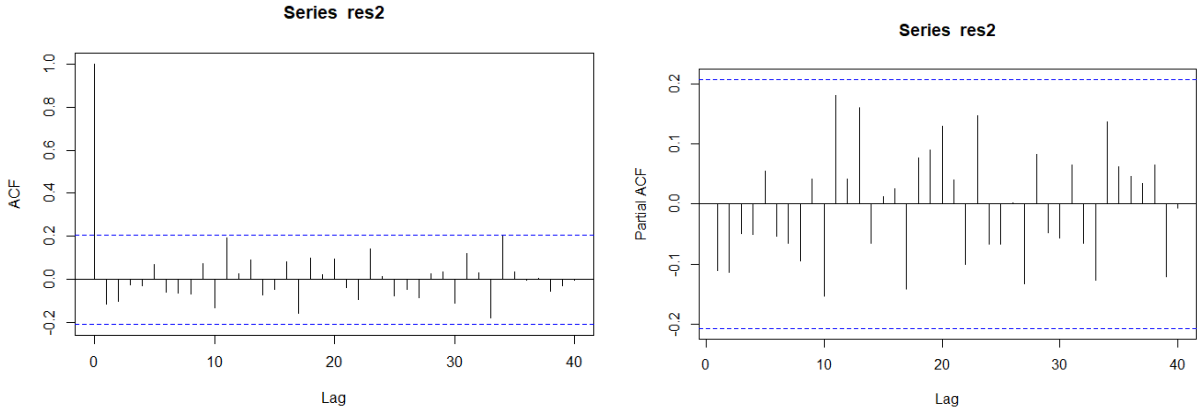






Based on the three plots above, we observed no seasonality in residuals. Like residuals in model A, there was a small trend in the residuals plot of model B, but we can ignore it because of the really small scale. The mean of residuals is 7.64192e-06, which is very close to 0. And the histogram looks almost Gaussian but with a small tail, so we will use Shapiro-Wilk test to confirm the normality. Overall, the histogram and the Q-Q plot looks fine.

Then, we need to check sample ACF and PACF of residuals of model B, which is supposed to resemble white noise. The ACF and PACF plots are shown below:



all acf and pacf of residuals are within confidence intervals and can be counted as zeros. So model A passed ACF/PACF test.

Next, we ran Shapiro-Wilk test of normality and Portmanteau tests for model B. And we used the Yule-Walker estimation The codes and results are shown below:

```
> shapiro.test(res2)

        Shapiro-Wilk normality test

data:  res2
W = 0.98816, p-value = 0.6042

> Box.test(res2, lag = 11, type = c("Box-Pierce"), fitdf = 2)

        Box-Pierce test

data:  res2
X-squared = 8.9038, df = 9, p-value = 0.4462

> Box.test(res2, lag = 11, type = c("Ljung-Box"), fitdf = 2)

        Box-Ljung test

data:  res2
X-squared = 9.9912, df = 9, p-value = 0.3512

> Box.test(res2^2, lag = 11, type = c("Ljung-Box"), fitdf = 0)

        Box-Ljung test

data:  res2^2
X-squared = 6.4016, df = 11, p-value = 0.8453
```

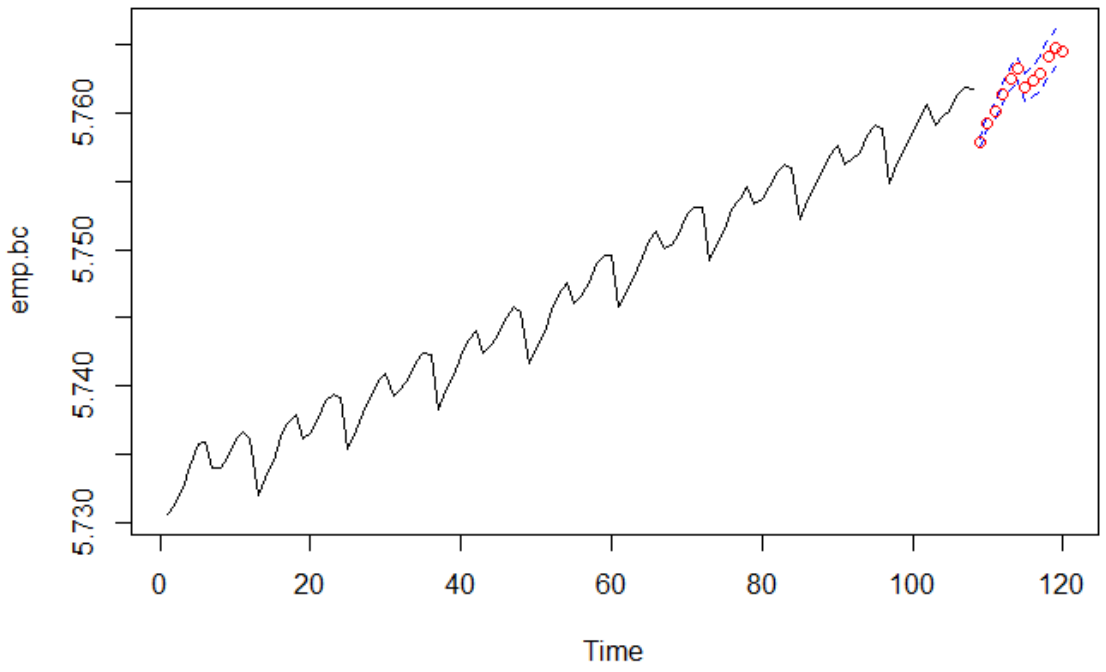
```
> ar(res2, aic = TRUE, order.max = NULL, method = c("yule-walker"))

Call:
ar(x = res2, aic = TRUE, order.max = NULL, method = c("yule-walker"))

Order selected 0  sigma^2 estimated as  2.751e-08
```

In Shapiro-Wilk normality test, the p-value is 0.6042 > 0.05, so we accept that the residuals are normally distributed. In addition, the p-value of Box-Pierce test, Ljung-Box test and McLeod-Li test are 0.4462 > 0.05, 0.3512 > 0.05, and 0.8452 > 0.05 respectively. Therefore, we accept that the residuals of model B are independently distributed. Lastly, the Yule-Walker estimation fit the residuals of model B into AR(0) model. Thus, model B passed diagnostic checking and ready to be used for forecasting.

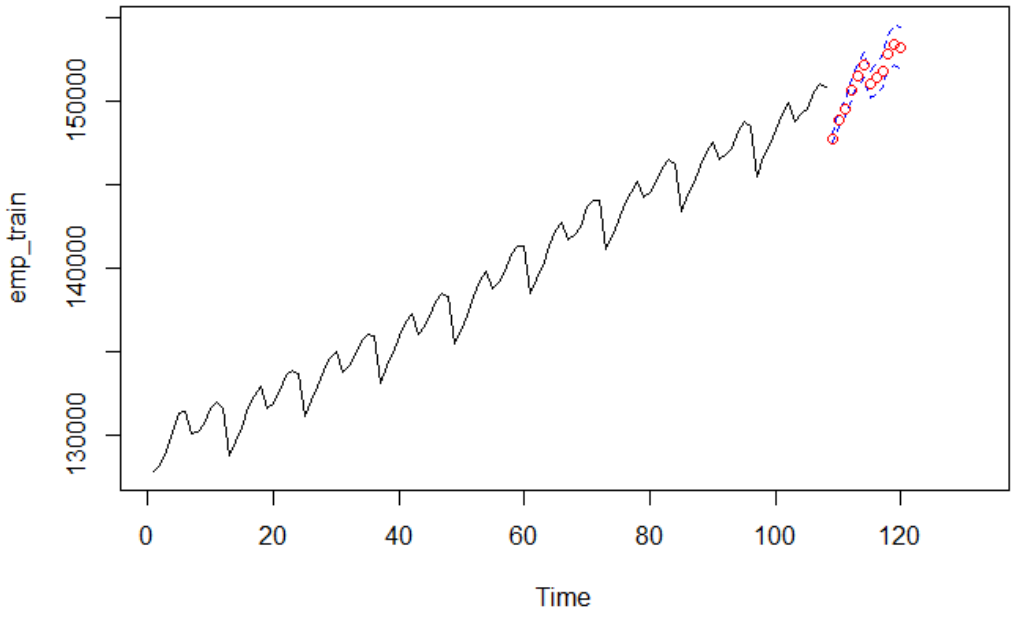
To sum up, since model A and model B both passed all the tests and their AICc value is pretty close (-12.78164 for model A and -12.78042 for model B). But model B has fewer parameter. Therefore, we chose model B for forecasting. So our finalized model used for forecasting is a SARIMA(6,1,0)×(0,1,1)$_{12}$ model with algebraic expression:

$(1-0.4158_{(0.1162)}B^{6})(1-B)(1-B^{12})X_t = (1-0.5697_{(0.1105)}B^{12})Z_t$, where $Z_t \sim WN(0,\ 3.352e\text{-}08)$.

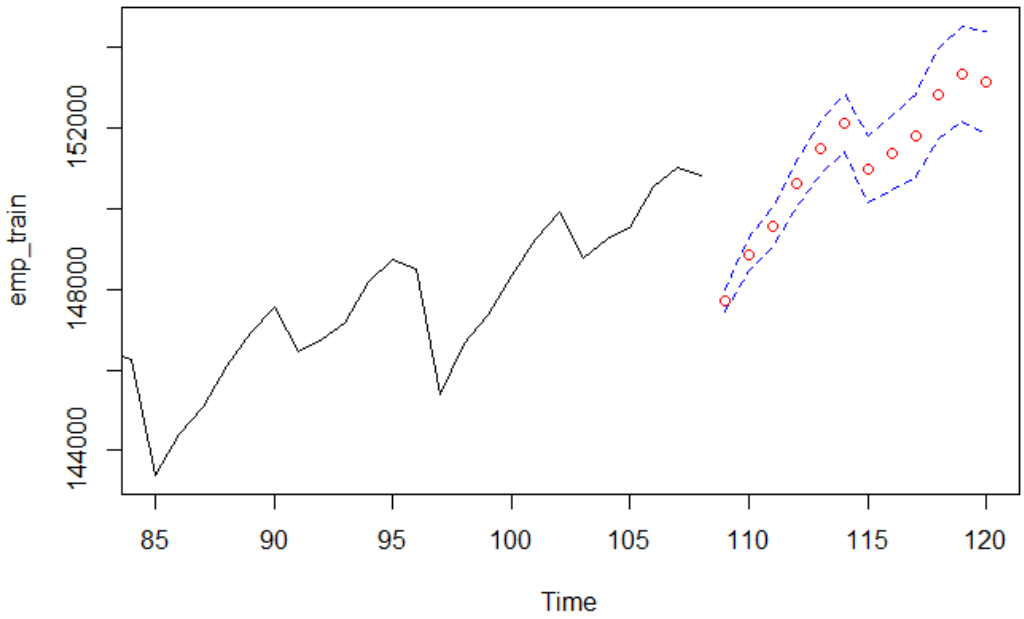
Now we forecasted transformed data using model B:



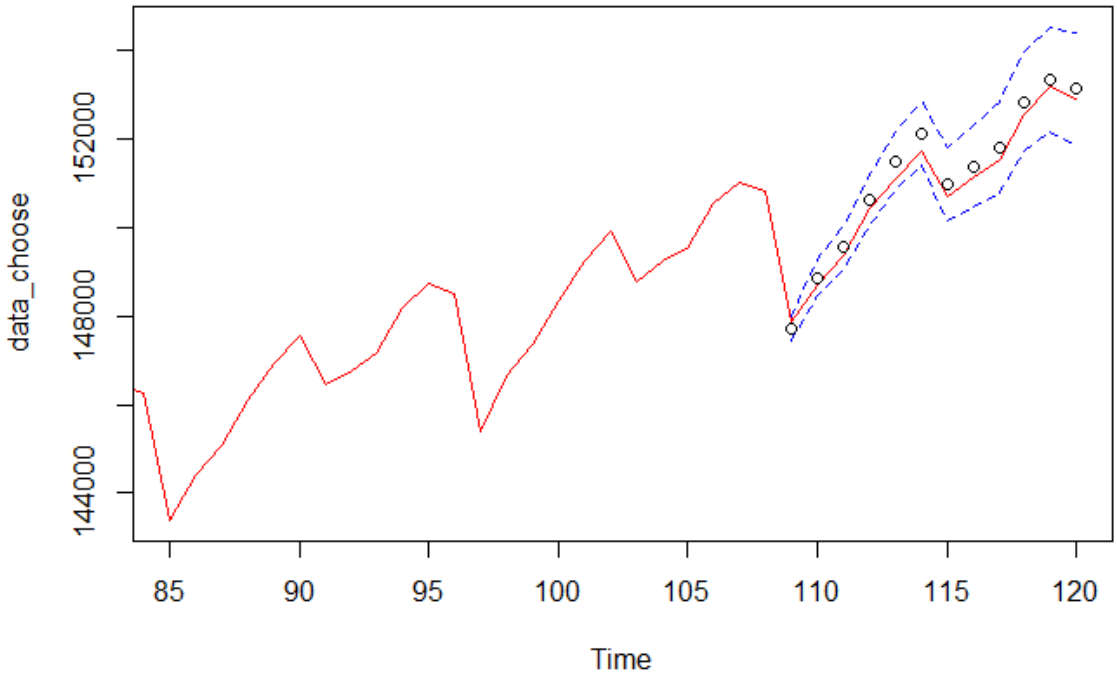
And we forecasted original data using model B:



To better see the result, we zoomed forecast of original data using model B, starting from entry 85:



Then we added the true value into the zoomed forecast plot:



The black circles are our predicted values, red line is the original data, and the blue dashed line are the confidence intervals. We can see from the above graph that our predictions are all within the confidence interval, and the true values are all within the confidence interval too. And our predicted values are all very close to the true values. Hence, I conclude that this is a successful prediction.

Conclusion

In summary, our final model for the Box-Cox transform of original data bc(Ut) follows SARIMA(6,1,0)×(0,1,1)$_{12}$ model with algebraic form:

$(1-0.4158_{(0.1162)}B^{6})(1-B)(1-B^{12})X_t = (1-0.5697_{(0.1105)}B^{12})Z_t$, where $Z_t \sim WN(0,\ 3.352e\text{-}08)$.

Our goal in this project is to forecast the 12 months data in year 2019. Since our prediction is pretty close to the original data and are all within confidence intervals. We can summarize that we successfully forecasted the employment data from 01/01/2019 to 12/01/2019 by analyzing the monthly, not seasonally adjusted employment data from 01/01/2010 to 12/01/2018.

To acknowledge everyone who helped me with this project, I want to first send my appreciation to Dr.Feldman, who is also my professor in this course. I am truly grateful to her passion of teaching this course and her patience of helping everyone to success in this course and this project. Thank you. And I also want to say thank you to both of the TAs: Nicole Yang and Jimin Lin. Thank you for your generous help in this project and throughout the quarter. I am also

grateful to all my classmates in this quarter. We went through a really complicated pandemic year together. Finally, thanks U.S. Bureau of Labor Statistics for collecting all these data and FRED, Federal Reserve Bank of St. Louis for providing this dataset so that I can use it to complete this project.

Reference

U.S. Bureau of Labor Statistics, All Employees, Total Nonfarm [PAYNSA], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/PAYNSA, December 4, 2020.

And all the PSTAT 174 course slides on GauchoSpace.

Appendix

```
employ_data <- read.csv("PAYNSA.csv")

View(employ_data)

length(employ_data[,2])

# We'll only take dataset from 01/01/2010 to 12/01/2019

data_choose <- employ_data[853:972,2]

l <- length(data_choose)

View(data_choose)

# Plot the raw employment data

plot.ts(data_choose, ylab = "Number of Employees")

fit <- lm(data_choose ~ as.numeric(1:l)); abline(fit, col="red")

abline(h=mean(data_choose), col="blue")

# Plot employment data with year on x-axis

emp <- ts(data_choose,start=c(2010,01,01),end=c(2019,12,01),frequency = 12)

ts.plot(emp,xlab="Years",ylab="Number of Employees", main="Raw Data")


# Seperate data into training set and test set

emp_train <- data_choose[c(1:108)]       # training dataset, to build model: 2010-01-01 to
                                         # 2018-12-01

emp_test <- data_choose[c(108:120)]      # test dataset: 2019-01-01 to 2019-12-01
```

```
# Plot training data
plot.ts(emp_train, ylab = "Number of Employees", main = "Training Data")
fit1 <- lm(emp_train ~ as.numeric(1:length(emp_train))); abline(fit1, col="red")
abline(h=mean(emp_train), col="blue")
var(emp_train)    # 41531786
# Plots histogram of trainsing data
hist(emp_train, col="light blue", xlab="", main="histogram: Number of employee")
# Plots ACF of training data
acf(emp_train,lag.max=40, main="ACF of Employment Stats")

# Perform transformations, plot transformed data, histograms:
library(MASS)
bcTransform <- boxcox(emp_train ~ as.numeric(1:length(emp_train)))
bcTransform$x[which(bcTransform$y == max(bcTransform$y))]    # lambda = -0.1414141
lambda <- bcTransform$x[which(bcTransform$y == max(bcTransform$y))]

# Perform transformations, plot transformed data, histograms:
emp.bc <- (1/lambda)*(emp_train^lambda-1)
plot.ts(emp.bc,main="Box-Cox transformed data")
var(emp.bc)      # 7.509169-05

emp.log <- log(emp_train)
plot.ts(emp.log, main="Log transformed data")
var(emp.log)    # 0.0021139675

hist(emp.log, col="light blue", xlab="", main="histogram; ln(U_t)")
hist(emp.bc, col="light blue", xlab="", main="histogram; bc(U_t)")
# Not much difference between transforms

# Produce decomposition of bc(U_t):
```

```
y <- ts(as.ts(emp.bc), frequency = 12)
decomp <- decompose(y)
plot(decomp)

# difference at lag 12
emp.bc_12 <- diff(emp.bc, lag=12)
plot.ts(emp.bc_12, main="bc(U_t) differenced at lag 12")
fit <- lm(emp.bc_12 ~ as.numeric(1:length(emp.bc_12))); abline(fit, col="red")
abline(h=mean(emp.bc_12), col="blue")
var(emp.bc_12)      # 2.637454e-07 -- variance is lower!
mean(emp.bc_12)    # 0.003117272

# difference at lag 1
emp.stat <- diff(emp.bc_12, lag=1)
plot.ts(emp.stat, main="bc(U_t) differenced at lag 12 & lag 1")
fit <- lm(emp.stat ~ as.numeric(1:length(emp.stat))); abline(fit, col="red")
abline(h=mean(emp.stat), col="blue")
var(emp.stat)         # 4.486463e-08    -- lower!
mean(emp.stat)       # 1.539943e-05

# emp.stat is bc transformed truncated data, differenced at lags 12 and then 1.
acf(emp.bc, lag.max=40, main="ACF of the bc(U_t)")
acf(emp.bc_12, lag.max=40, main="ACF of the bc(U_t), differenced at lag 12")
acf(emp.stat, lag.max=40, main="ACF of the bc(U_t), differenced at lags 12 and 1")
pacf(emp.stat, lag.max=40, main="PACF of the bc(U_t), differenced at lags 12 and 1")
hist(emp.stat, col="light blue", xlab="", main="histogram; bc(U_t) differenced at lags 12 & 1")

# Histogram of transformed and differenced data with normal curve:
hist(emp.stat, density=20,breaks=20, col="blue", xlab="", prob=TRUE,
```

```
      main="histogram; bc(U_t) differenced at lags 12 & 1")
m<-mean(emp.stat)
std<- sqrt(var(emp.stat))
curve(dnorm(x,m,std), add=TRUE)

# Trying models.
# SMA models tried: Q=1, q=0, 1.
sarima(emp.bc, p=0,d=1,q=1,P=0,D=1,Q=1,S=12, details = FALSE)    # AICc = -12.68445
sarima(emp.bc, p=0,d=1,q=0,P=0,D=1,Q=1,S=12, details = FALSE)    # AICc = -12.687

# SAR models tried: P=1, p=0, 1,6
sarima(emp.bc, p=0,d=1,q=0,P=1,D=1,Q=0,S=12, details = FALSE)    # AICc = -12.6629
sarima(emp.bc, p=1,d=1,q=0,P=1,D=1,Q=0,S=12, details = FALSE)    # AICc = -12.66516
sarima(emp.bc, p=6,d=1,q=0,P=1,D=1,Q=0,S=12, details = FALSE)    # AICc = -12.71039

# SARIMA models tried: P=Q=1, p = 0, 1, 6, q = 0, 1
sarima(emp.bc, p=0,d=1,q=1,P=1,D=1,Q=1,S=12, details = FALSE)    # AICc = -12.66769
sarima(emp.bc, p=1,d=1,q=1,P=1,D=1,Q=1,S=12, details = FALSE)    # AICc = -12.65454
sarima(emp.bc, p=6,d=1,q=1,P=1,D=1,Q=1,S=12, details = FALSE)    # AICc = -12.70943
sarima(emp.bc, p=0,d=1,q=0,P=1,D=1,Q=1,S=12, details = FALSE)    # AICc = -12.67006
sarima(emp.bc, p=1,d=1,q=0,P=1,D=1,Q=1,S=12, details = FALSE)    # AICc = -12.66323
sarima(emp.bc, p=6,d=1,q=0,P=1,D=1,Q=1,S=12, details = FALSE)    # AICc = -12.72683 -- lowest so far

# Fix coefficient for the model wl lowest AICc
sarima(emp.bc, p=6,d=1,q=0,P=1,D=1,Q=1,S=12, details = FALSE,
          fixed = c(0,0,0,0,0,NA,NA,NA))
# AICc = -12.76346
sarima(emp.bc, p=6,d=1,q=0,P=0,D=1,Q=1,S=12, details = FALSE,
          fixed = c(NA,NA,NA,0,NA,NA,NA))
```

```
# AICc = -12.76906
sarima(emp.bc, p=6,d=1,q=0,P=0,D=1,Q=1,S=12, details = FALSE,
           fixed = c(NA,0,NA,0,NA,NA,NA))
# AICc = -12.77971
# Below are the two models with the lowest and the second lowest AICc value
# I also tried to fixed coefficients in other parameters
# but I deleted them since they are not necessary here because they gave higher AICc values
sarima(emp.bc, p=6,d=1,q=0,P=0,D=1,Q=1,S=12, details = FALSE,
           fixed = c(0,0,0,0,0,NA,NA))
# AICc = -12.78042      ---- second lowest aic
sarima(emp.bc, p=6,d=1,q=0,P=0,D=1,Q=1,S=12, details = FALSE,
           fixed = c(NA,0,0,0,0,NA,NA))
# AICc = -12.78164    --- lowest aic

# model 1:
fit1 <- arima(emp.bc, order=c(6,1,0), seasonal = list(order = c(0,1,1), period = 12),
                    transform.pars = TRUE, fixed = c(NA,0,0,0,0,NA,NA), method="ML")
# model 2:
fit2 <- arima(emp.bc, order=c(6,1,0), seasonal = list(order = c(0,1,1), period = 12),
                    transform.pars = TRUE, fixed = c(0,0,0,0,0,NA,NA), method="ML")

# Check if model 1 and 2 are stationary
plot.roots(NULL,polyroot(c(1, -0.1514, 0, 0, 0, 0, 0.4280)),
                main="model 1: roots of AR part non-seasonal")
plot.roots(NULL,polyroot(c(1,0, 0, 0, 0, 0, 0.4158)),
                main="model 2: roots of AR part non-seasonal")

# Diagnostic checking for model 1:
res1<-residuals(fit1)
```

```
res1 <- res1[c(20:108)]    # opt out outliers
hist(res1,density=20,breaks=20, col="blue", xlab="", prob=TRUE)
m <- mean(res1)
std <- sqrt(var(res1))
curve(dnorm(x,m,std), add=TRUE)
plot.ts(res1)
fitt <- lm(res1 ~ as.numeric(1:length(res1))); abline(fitt, col="red")
abline(h=mean(res1), col="blue")
qqnorm(res1,main= "")
qqline(res1,col="blue")
acf(res1, lag.max=40)      # passed
pacf(res1, lag.max=40)    # passed
shapiro.test(res1)        # passed
Box.test(res1, lag = 11, type = c("Box-Pierce"), fitdf = 3)
Box.test(res1, lag = 11, type = c("Ljung-Box"), fitdf = 3)
Box.test(res1^2, lag = 11, type = c("Ljung-Box"), fitdf = 0)      # Passed all
acf(res1^2, lag.max=40)
ar(res1, aic = TRUE, order.max = NULL, method = c("yule-walker"))    # chose AR(0) = WN

# Diagnostic checking for model 2:
res2<-residuals(fit2)
res2 <- res2[c(20:108)]    # opt out outliers
hist(res2,density=20,breaks=20, col="blue", xlab="", prob=TRUE)
m <- mean(res2)
std <- sqrt(var(res2))
curve(dnorm(x,m,std), add=TRUE)
plot.ts(res2)
fitt <- lm(res2 ~ as.numeric(1:length(res2))); abline(fitt, col="red")
abline(h=mean(res2), col="blue")
```

```
qqnorm(res2,main= "")
qqline(res2,col="blue")
acf(res2, lag.max=40)      # passed
pacf(res2, lag.max=40)    # passed
shapiro.test(res2)        # passed
Box.test(res2, lag = 11, type = c("Box-Pierce"), fitdf = 2)
Box.test(res2, lag = 11, type = c("Ljung-Box"), fitdf = 2)
Box.test(res2^2, lag = 11, type = c("Ljung-Box"), fitdf = 0)    # passed all
acf(res2^2, lag.max=40)
ar(res2, aic = TRUE, order.max = NULL, method = c("yule-walker"))    # chose AR(0) = WN

# Forecasting using model B:
install.packages("forecast")
library(forecast)
# Choose model B since it has less parameter
fit.A <- arima(emp.bc, order=c(6,1,0), seasonal = list(order = c(0,1,1), period = 12),
                        transform.pars = TRUE, fixed = c(0,0,0,0,0,NA,NA), method="ML")
forecast(fit.A)
# To produce graph with 12 forecasts on transformed data:
pred.tr <- predict(fit.A, 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(emp.bc, xlim=c(1,length(emp.bc)+12), ylim = c(min(emp.bc),max(U.tr)))
lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")
points((length(emp.bc)+1):(length(emp.bc)+12), pred.tr$pred, col="red")
# produce graph with forecasts on original data:
pred.orig <- (lambda*(pred.tr$pred)+1)^(1/lambda)    # problem: in ^lambda
U <- (lambda*(U.tr)+1)^(1/lambda)
```

```
L <- (lambda*(L.tr)+1)^(1/lambda)
ts.plot(emp_train, xlim=c(1,length(emp)+12), ylim = c(min(emp),max(U)))
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(emp_train)+1):(length(emp_train)+12), pred.orig, col="red")

# zoom the graph, starting from entry 85:
ts.plot(emp_train, xlim = c(85,length(emp_train)+12), ylim = c(143377,max(U)))
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(emp_train)+1):(length(emp_train)+12), pred.orig, col="red")

# plot zoomed forecasts and true values:
ts.plot(data_choose, xlim = c(85,length(emp_train)+12), ylim = c(143377,max(U)), col="red")
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(emp_train)+1):(length(emp_train)+12), pred.orig, col="green")
points((length(emp_train)+1):(length(emp_train)+12), pred.orig, col="black")
# Forecasts—black circles;    Original data– red line
```