University of California Santa Barbara

PSTAT274 FINAL PROJECT

Trends in Atmospheric Concentrations of CO_2

Submitted by:

Selin Karabulut

ID-6075253

December 2022

Abstract

Mauna Loa is the world's largest active volcano which recently erupted "for the first time in four decades" [1]. Researchers from the United States National Oceanic and Atmospheric Association (NOAA), before their work was interrupted by this recent eruption of the Mauna Loa volcano, were stationed there to measure how the world's atmospheric carbon dioxide has been changing over time. With their research, they have been contributing to the research on climate change. CO_2 as one of the greenhouse gases is the primary driver of climate change [2] [3]. Studying how the CO_2 level has been changing over time helps us understand the severity of the issue of climate change. It is equally important to be able to make predictions about how it will change in the future because with that, we can raise awareness among the public and inform policymakers about the issue so that they can make policies that make it easy to apply preventative measures. With the aim to model the atmospheric CO_2 level utilizing the monthly data collected at the Mauna Loa Observatory in Hawaii, between 2000 and 2019, and make validation for the model about the CO_2 level for the year 2020 with the 1 year ahead forecasting. Additionally, the 3-year ahead (2019-2022) forecast provides a visual representation of atmospheric CO_2 level in the hypothetical scenario where COVID-19 did not occur, and we have continued to emit the same level of CO_2. Forecast shows that unfortunately, the CO_2 level has continued to increase despite the pandemic forcing people to stay at home and consequently not emit CO_2 at the same rate as pre-pandemic.

[1] https://www.cnn.com/2022/11/29/weather/volcano-terms-meaning-xpn-trnd/index.html

[2] https://ourworldindata.org/co2-and-other-greenhouse-gas-emissions

[3] https://climate.nasa.gov/vital-signs/carbon-dioxide/

Contents

List of Figures

1 Introduction

In this project, I analyze a dataset called "The Mauna Loa data", sourced from the US Government's Earth System Research Laboratory, Global Monitoring Division [4]. This data contains monthly mean carbon dioxide measured at Mauna Loa Observatory, Hawaii.

This data is important because it shows how the CO_2 level has changed over time. What is also interesting about this data is that it helps us test the hypothesis that global lowdown due to COVID-19 led reduction in the CO_2 emission. The rationale is that since people had to stay at home, did neither drive their cars nor book flights etc for at least more than a year, the world's atmospheric carbon dioxide level would drop significantly [5] (1). Therefore, this project aims to forecast values using a time series model fit on the years 2010-2019.

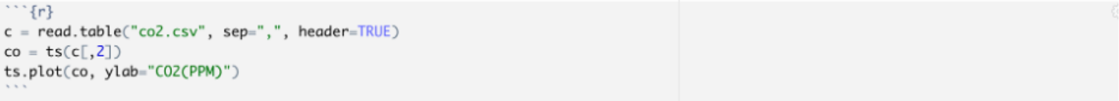
I predict values 1 year ahead (2019-2020) for validation. Then, I predict values 3-year ahead (2019-2022), to contrast my predictions with real-world CO_2 levels affected by COVID. Thus, I compare the model's predictions in a hypothetical case where COVID never occurred to COVID-affected CO_2 levels, to demonstrate the impact of COVID on the world's atmospheric CO_2.

In this study I fit models to original, box-cox, and log-transformed versions of the data. I find that log-transformed data is the most appropriate version and proceed with it directly. Then, comparing several candidate models, I find that SARIMA $(1-0.2766B)(1-B)(1-B^{12})X_t = (1-0.6986B)(1-0.9169B^{12})Z_t$ is the best model based on AICc, parsimony, and diagnostics results, including spectral analysis. The model is then used to perform forecasting.

As a conclusion, forecasts show that CO_2 level has not affected by the pandemic. It continues on its trajectory despite humans consumed less and emitted less $C0_2$ during the pandemic.

2 Time Series Analysis

2.1 Exploratory data analysis

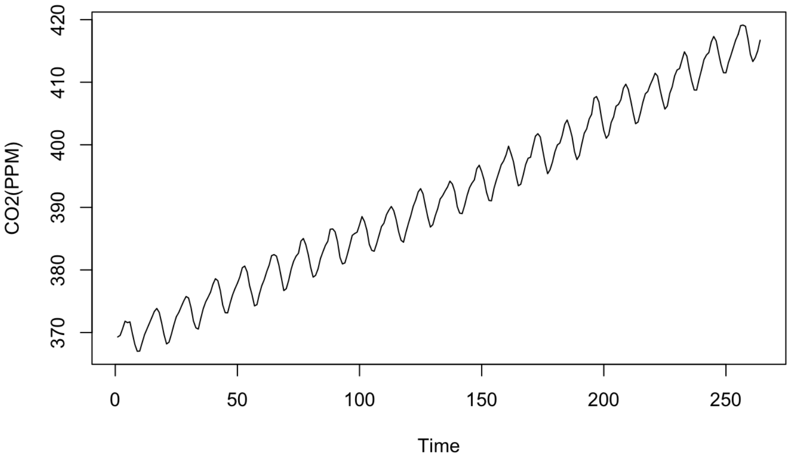


~~~
```{r}
c = read.table("co2.csv", sep=",", header=TRUE)
co = ts(c[,2])
ts.plot(co, ylab="CO2(PPM)")
```
~~~

Figure 1 below demonstrates how the entirety of the monthly time series data I am working with looks like. It covers from January 2000 to December 2021 (264 observations). The following observations can be made from Figure 1;
* There is a clear upwards trend
* There is a clear seasonal pattern (yearly)

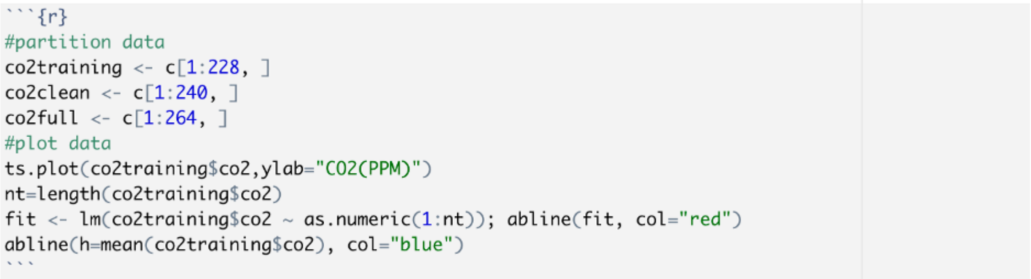
[4] https://gml.noaa.gov/ccgg/trends/index.html
[5] https://unfccc.int/sites/default/files/resource/1.GCP_.pdf

Figure 1: Monthly CO2(PPM), 01/2000–12/2021



* It does not look like the upward, seasonal trend has changed for the period after 2019

I will exclude COVID-19 times from my analysis to ensure that the underlying mechanism is consistent for the training data. I will use the first 19 years (January 2000-January 2019) as my training data and the data for the following year (January 2019-January 2020) as my testing data.



~~~
```{r}
#partition data
co2training <- c[1:228, ]
co2clean <- c[1:240, ]
co2full <- c[1:264, ]
#plot data
ts.plot(co2training$co2,ylab="CO2(PPM)")
nt=length(co2training$co2)
fit <- lm(co2training$co2 ~ as.numeric(1:nt)); abline(fit, col="red")
abline(h=mean(co2training$co2), col="blue")
```
~~~

Figure 2: Training data-Monthly CO2(PPM), 01/2000–1/2019

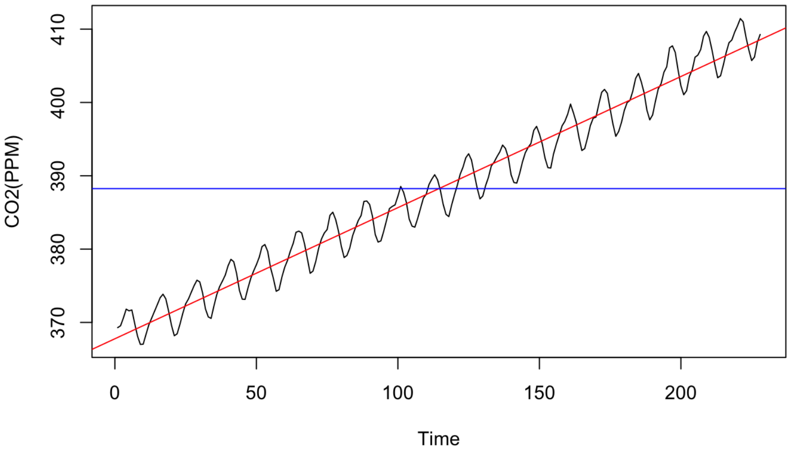
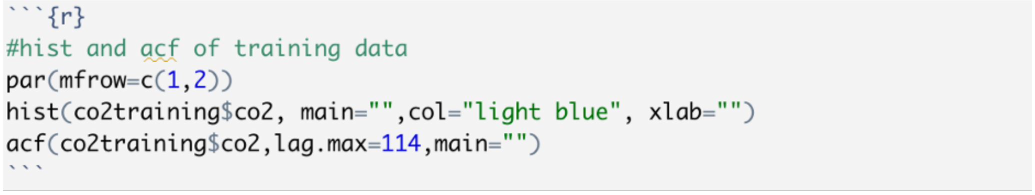


Figure 2 above demonstrates the training data with the trend and mean line added to it.

2.2 Transformations and stationarity

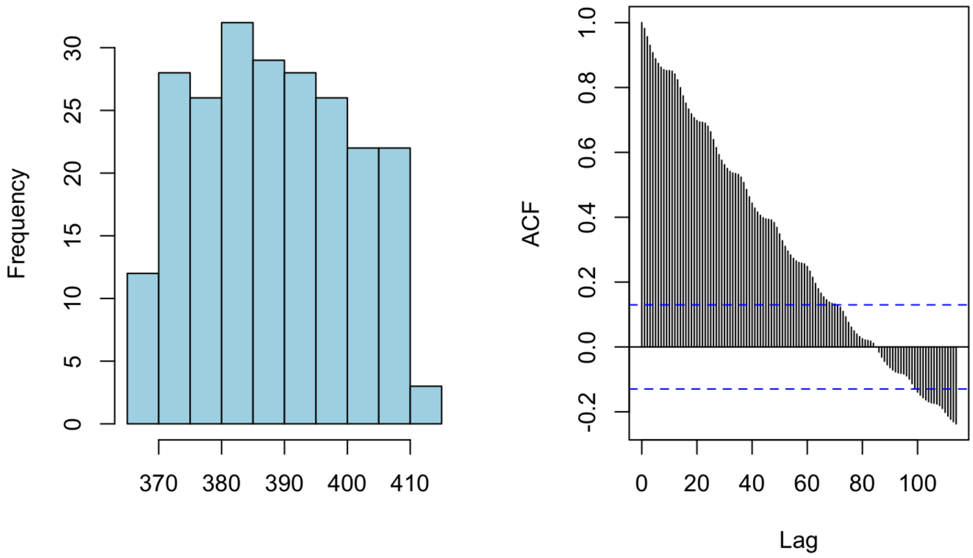
2.2.1 Transformation

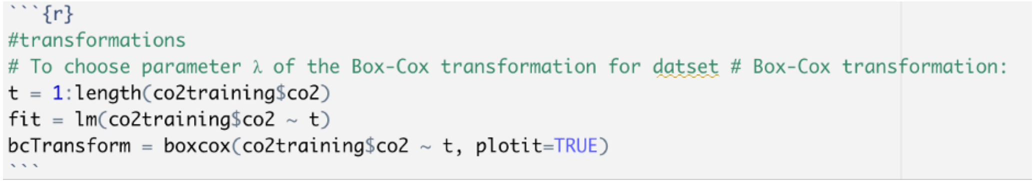
Figure 3 below illustrates the histogram and autocorrelation plot of CO2 data, respectively. While the histogram slightly deviates from the normal distribution, ACF indicates seasonal spikes. Along with the time series plot (Figure 2), this indicates that we need to adopt the differencing method. To eliminate the trend and seasonality we observe in Figures 2 and 3; I will first investigate whether the transformation is necessary and then perform differencing at multiple lags.



~~~
```{r}
#hist and acf of training data
par(mfrow=c(1,2))
hist(co2training$co2, main="",col="light blue", xlab="")
acf(co2training$co2,lag.max=114,main="")
```
~~~

Figure 3: Histogram and ACF of CO2 data



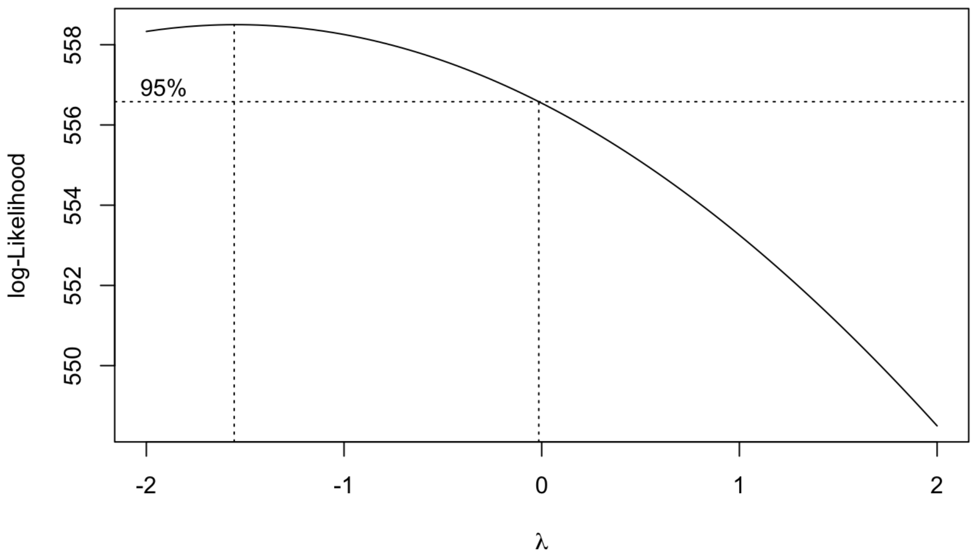


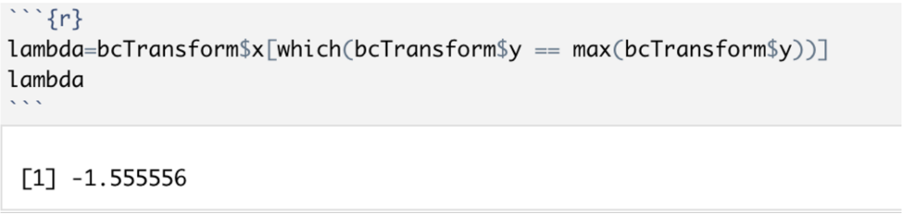
~~~
```{r}
#transformations
# To choose parameter λ of the Box-Cox transformation for datset # Box-Cox transformation:
t = 1:length(co2training$co2)
fit = lm(co2training$co2 ~ t)
bcTransform = boxcox(co2training$co2 ~ t, plotit=TRUE)
```
~~~

The Box-Cox transformation of the data (Figure 4) shows that confidence interval for lambda is wide and very close to 0 which suggests log transformation. Maximum lambda value is -1.55.

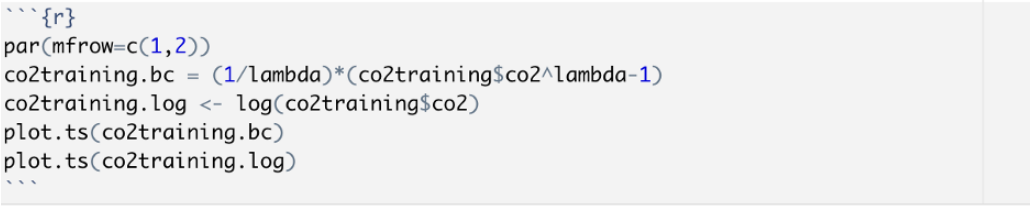
It looks like log transformation is the most appropriate method but before deciding on that, I will look at how variance and distribution on the histogram changes when we do log transformation and Box-Cox transformation.

Figure 4: Box-Cox transformation of CO2 data





~~~
```{r}
lambda=bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
lambda
```

[1] -1.555556
~~~

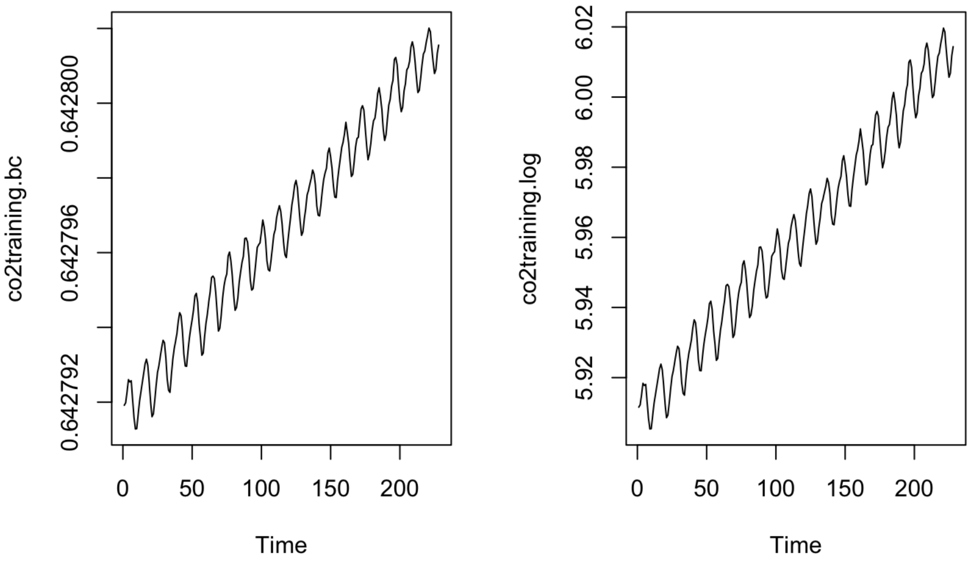


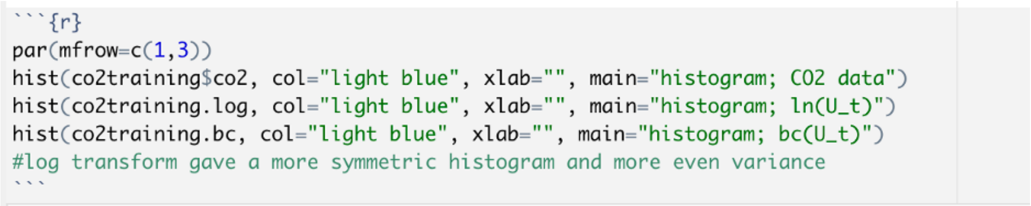
~~~
```{r}
par(mfrow=c(1,2))
co2training.bc = (1/lambda)*(co2training$co2^lambda-1)
co2training.log <- log(co2training$co2)
plot.ts(co2training.bc)
plot.ts(co2training.log)
```
~~~

Looking at the Box-Cox transformed and log transformed data suggests the following;

- Figure 5 demonstrates that variance is slightly reduced using log transformation.
- Based on Figure 6, Log transform gives a more symmetric histogram.
- Decomposition of $ln(U_t)$ in Figure 7 shows seasonality and almost linear trend. Hence, performing differencing is the step.

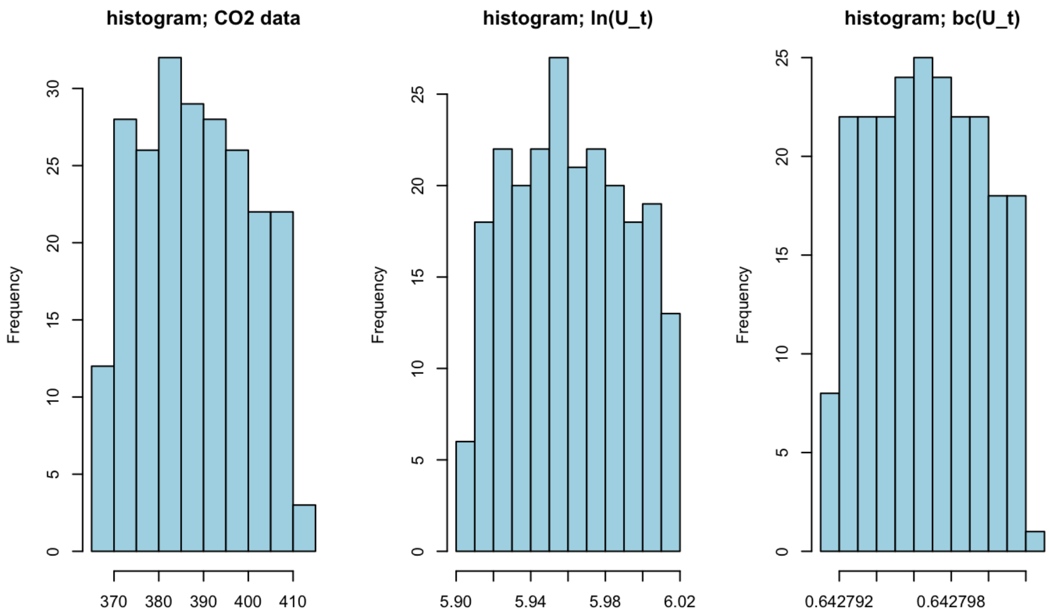
Figure 5: Plot of Box-Cox transformed data vs log transformed data

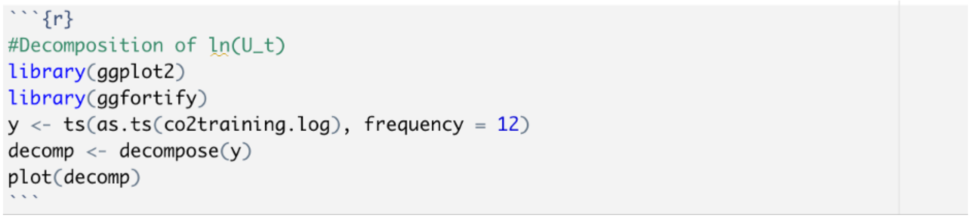




~~~
```{r}
par(mfrow=c(1,3))
hist(co2training$co2, col="light blue", xlab="", main="histogram; CO2 data")
hist(co2training.log, col="light blue", xlab="", main="histogram; ln(U_t)")
hist(co2training.bc, col="light blue", xlab="", main="histogram; bc(U_t)")
#log transform gave a more symmetric histogram and more even variance
```
~~~

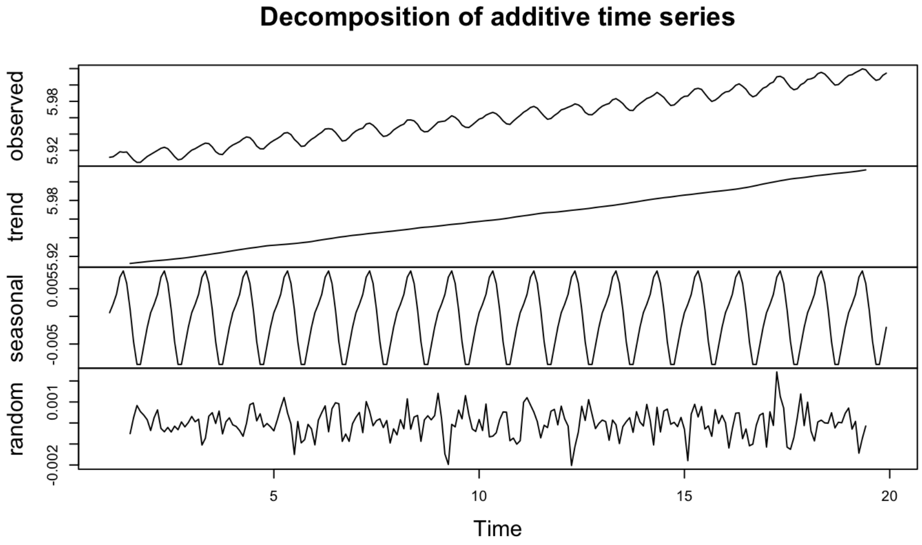
Figure 6: Histogram of U_t(training data), $ln(U_t)$, and Box-Cox (U_t)





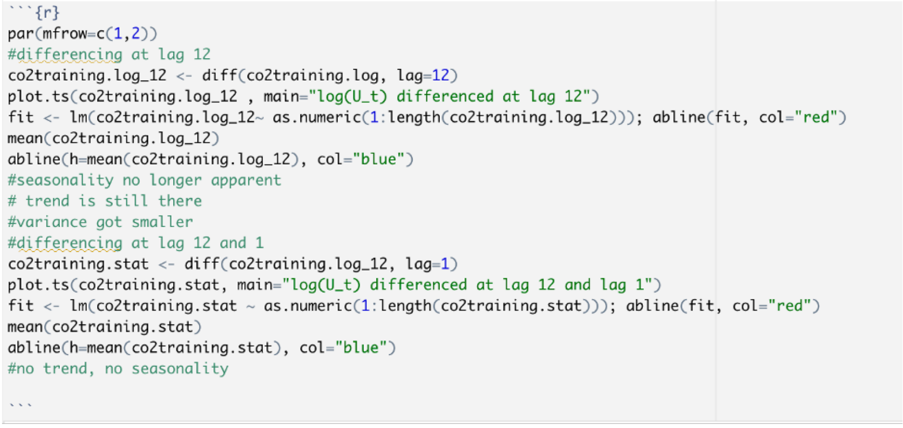
~~~
```{r}
#Decomposition of ln(U_t)
library(ggplot2)
library(ggfortify)
y <- ts(as.ts(co2training.log), frequency = 12)
decomp <- decompose(y)
plot(decomp)
```
~~~

Figure 7: Decomposition of $ln(U_t)$



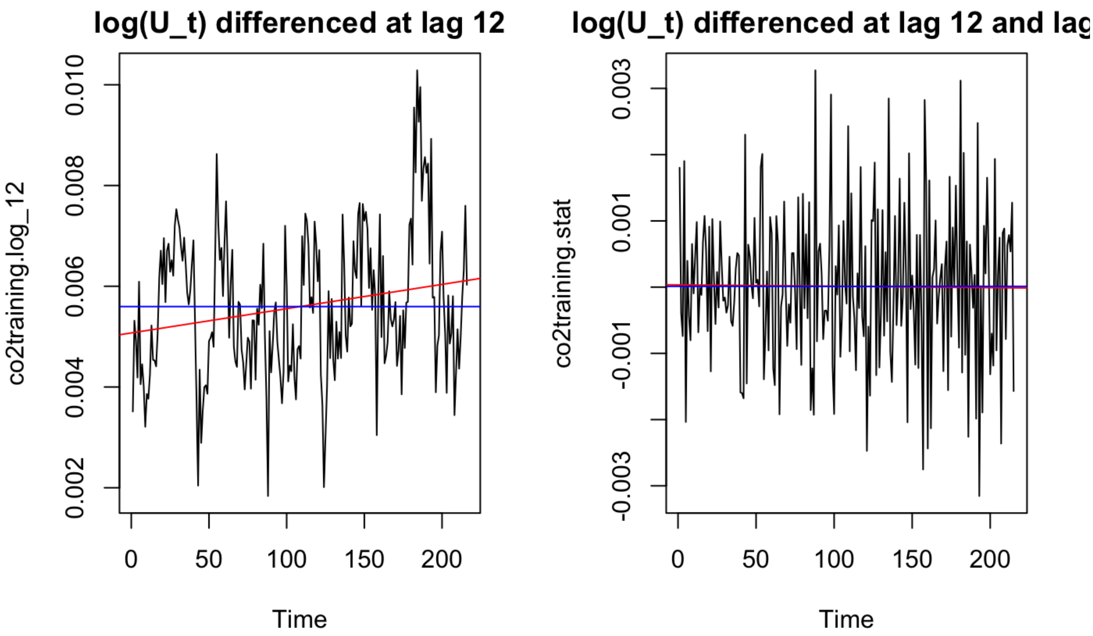
2.2.2 Differencing

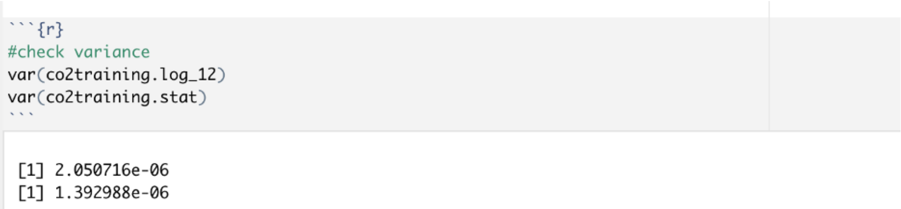
I difference at lag 1 once to remove trend and difference at lag 12 once to remove seasonality. I also inspect the variance of the data when additional differencing is performed. Performing differencing and checking variance reveals that differencing once at lags 1 and 12 is optimal (Figure 8).



~~~
```{r}
par(mfrow=c(1,2))
#differencing at lag 12
co2training.log_12 <- diff(co2training.log, lag=12)
plot.ts(co2training.log_12 , main="log(U_t) differenced at lag 12")
fit <- lm(co2training.log_12~ as.numeric(1:length(co2training.log_12))); abline(fit, col="red")
mean(co2training.log_12)
abline(h=mean(co2training.log_12), col="blue")
#seasonality no longer apparent
# trend is still there
#variance got smaller
#differencing at lag 12 and 1
co2training.stat <- diff(co2training.log_12, lag=1)
plot.ts(co2training.stat, main="log(U_t) differenced at lag 12 and lag 1")
fit <- lm(co2training.stat ~ as.numeric(1:length(co2training.stat))); abline(fit, col="red")
mean(co2training.stat)
abline(h=mean(co2training.stat), col="blue")
#no trend, no seasonality

```
~~~

Figure 8: Detrended/seasonalized Time Series

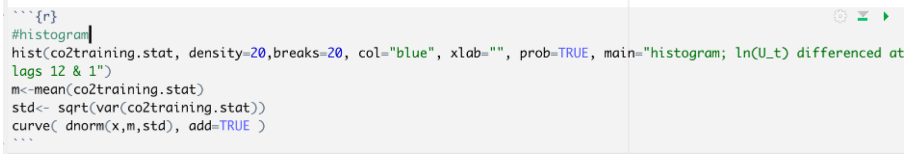




~~~
```{r}
#check variance
var(co2training.log_12)
var(co2training.stat)
```

[1] 2.050716e-06
[1] 1.392988e-06
~~~

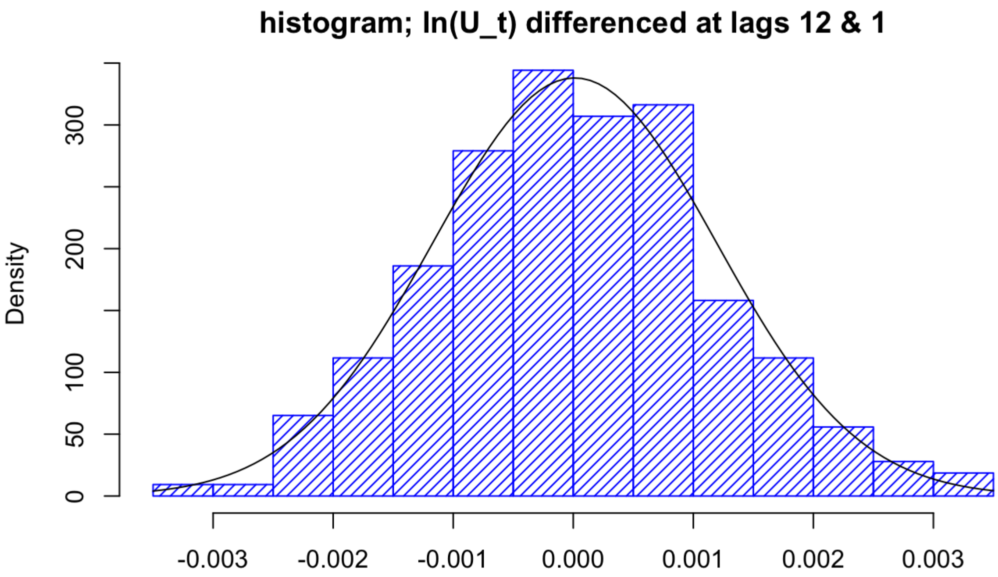
* Differencing the log transformed data at lag 1 and 12 helps lower the variance.

* Histogram (Figure 9) of the log-transformed, differenced at lags 1 and 12 data demonstrates that our data distribution looks more like Gaussian, compared to Figure 6.



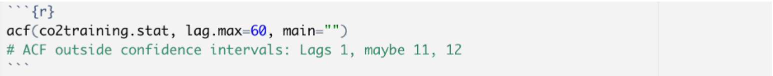
~~~~
```{r}
#histogram
hist(co2training.stat, density=20,breaks=20, col="blue", xlab="", prob=TRUE, main="histogram; ln(U_t) differenced at
lags 12 & 1")
m<-mean(co2training.stat)
std<- sqrt(var(co2training.stat))
curve( dnorm(x,m,std), add=TRUE )
```
~~~~

Figure 9: Histogram of detrended/seasonalized Time Series

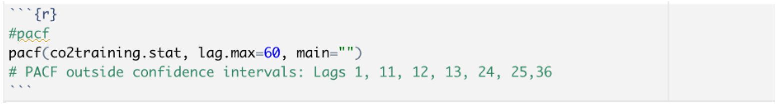


2.3 Preliminary model identification

Figure 10 below illustrates ACF of the data. At lags 1, 11, and 12, ACF outside confidence intervals. Due to significant spike at lag 12 and its surrounding lag, I conclude that Q, or seasonal MA component can be 1.



~~~~
```{r}
acf(co2training.stat, lag.max=60, main="")
# ACF outside confidence intervals: Lags 1, maybe 11, 12
```
~~~~



~~~~
```{r}
#pacf
pacf(co2training.stat, lag.max=60, main="")
# PACF outside confidence intervals: Lags 1, 11, 12, 13, 24, 25,36
```
~~~~

Figure 10: ACF of the $ln(U_t)$, differenced at lags 12 and 1

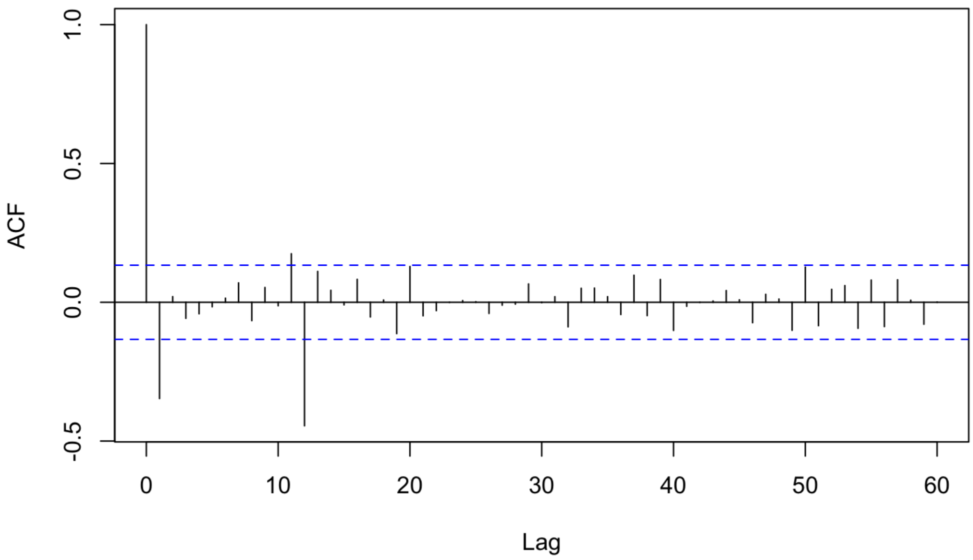
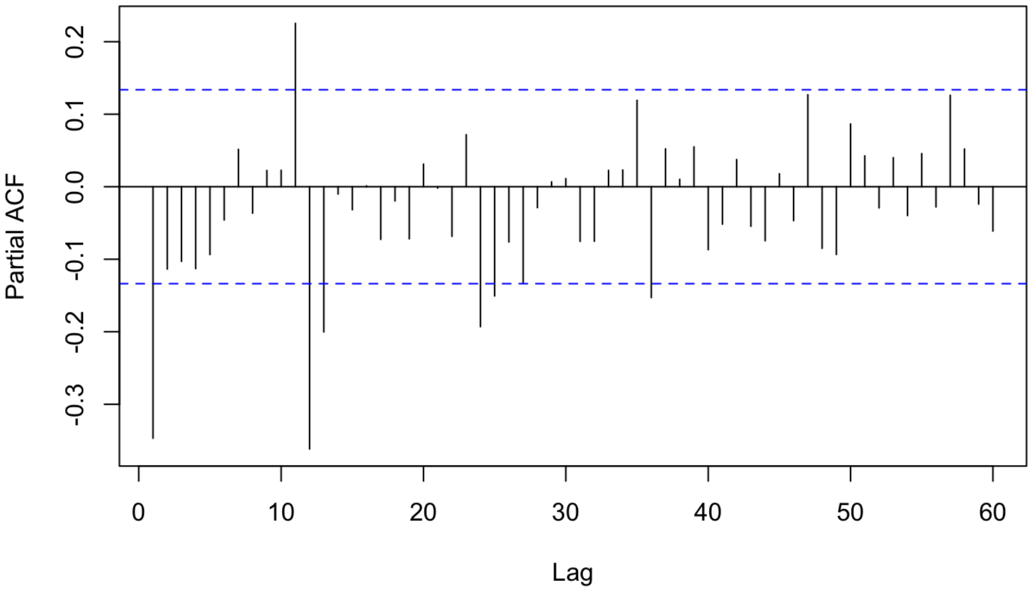
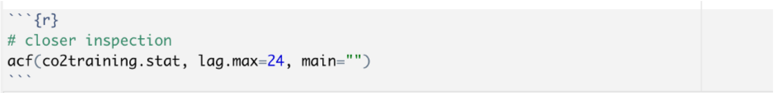


Figure 11 below illustrates PACF of the data. At lags 1, 11, 12, 13, 24, 25,36 PACF outside confidence intervals. Keeping the Bartley's formula in mind and considering the major significant spike at seasonal lag 12 and its surrounding lags, P, or seasonal AR component can be 1.

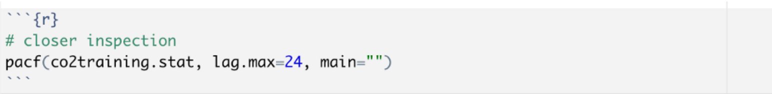
Figure 11: PACF of the $ln(U_t)$, differenced at lags 12 and 1





~~~
```{r}
# closer inspection
acf(co2training.stat, lag.max=24, main="")
```
~~~

Figure 12 provides closer inspection of ACF. There are spikes at lags 1 and 12. I then conclude that q, or non-seasonal MA component can be 1.



~~~
```{r}
# closer inspection
pacf(co2training.stat, lag.max=24, main="")
```
~~~

Figure 12: Closer look at ACF of the $ln(U_t)$, differenced at lags 12 and 1

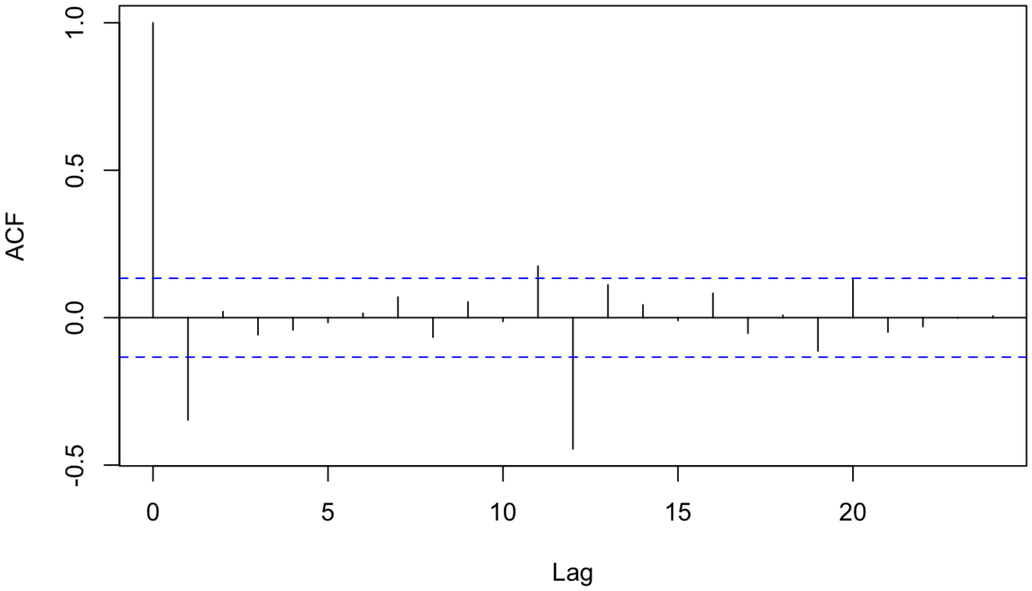


Figure 13: Closer look at PACF of the $ln(U_t)$, differenced at lags 12 and 1

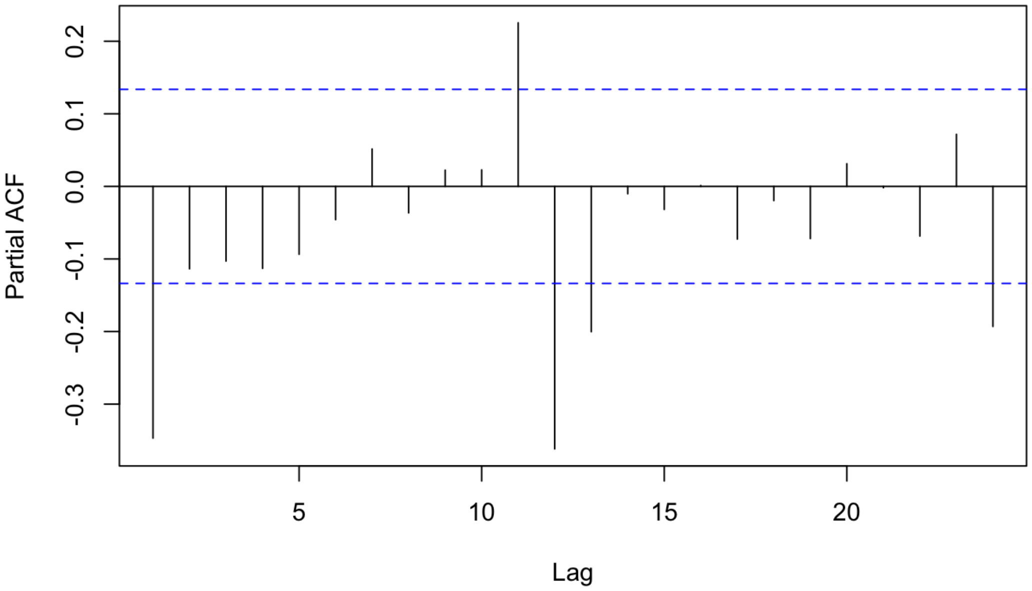


Figure 13 provides closer inspection of PACF. Spikes are present at lags 1,11,12, and 13. I then conclude that p, or non-seasonal AR component can be either 0 or 1.

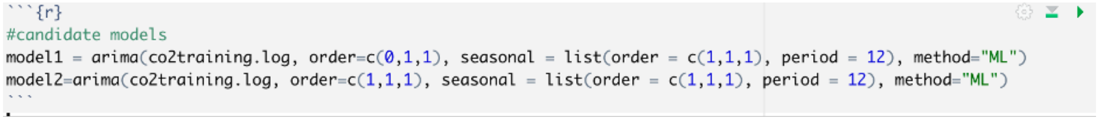
To summarize, preliminary model identification found the following:
d=1 D=1 p=0,1 P=1 q=1 Q=1
It therefore gives us two possible models:

* Model 1: $SARIMA(0,1,1)(1,1,1)_{12}$
* Model 2: $SARIMA(1,1,1)(1,1,1)_{12}$

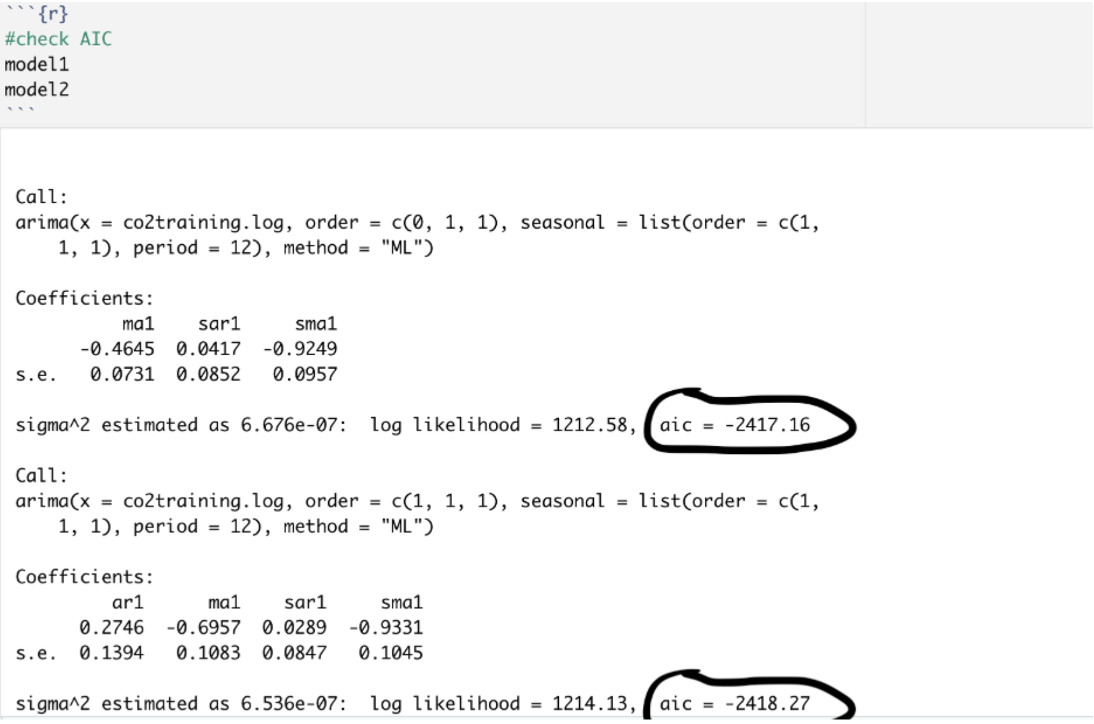
2.4 Model fitting, diagnostics, selection

2.4.1 Candidate models

I fit two candidate models specified at the end of the previous section and look at their AICc to evaluate the model performance.



~~~
```{r}
#candidate models
model1 = arima(co2training.log, order=c(0,1,1), seasonal = list(order = c(1,1,1), period = 12), method="ML")
model2=arima(co2training.log, order=c(1,1,1), seasonal = list(order = c(1,1,1), period = 12), method="ML")
```
~~~

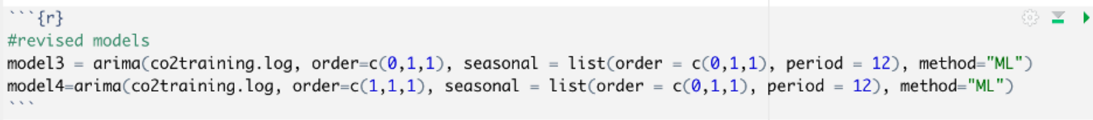


~~~
```{r}
#check AIC
model1
model2
```
~~~

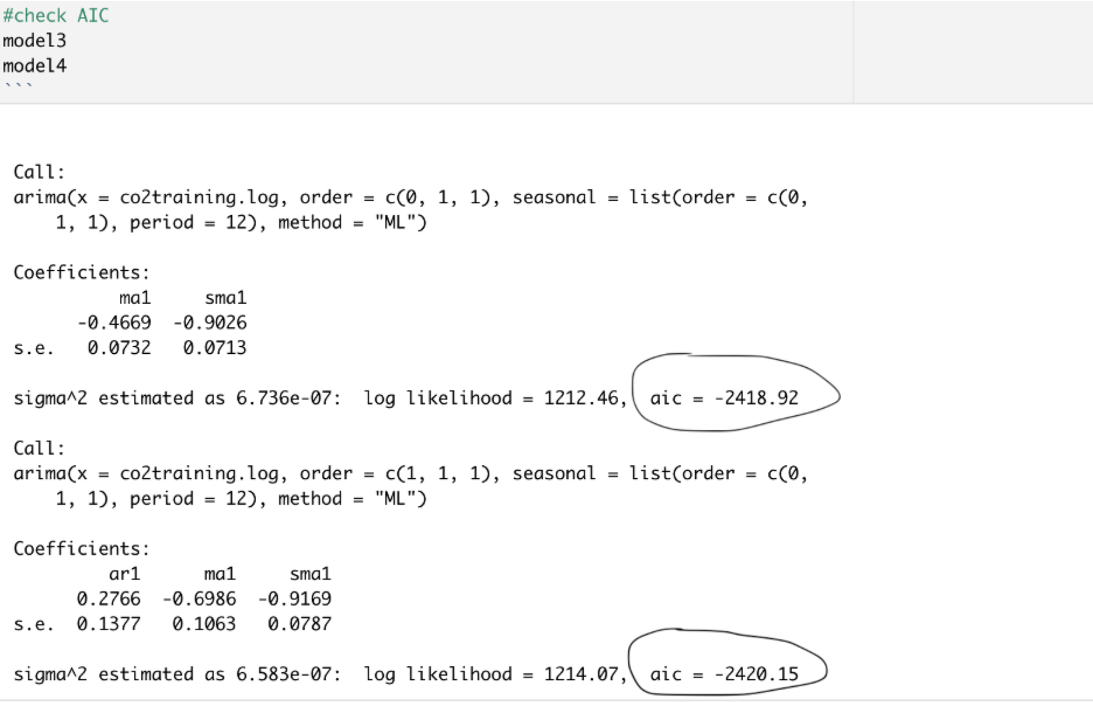
```

Call:
arima(x = co2training.log, order = c(0, 1, 1), seasonal = list(order = c(1,
    1, 1), period = 12), method = "ML")

Coefficients:
          ma1     sar1      sma1
      -0.4645   0.0417   -0.9249
s.e.   0.0731   0.0852    0.0957

sigma^2 estimated as 6.676e-07:  log likelihood = 1212.58,  aic = -2417.16

Call:
arima(x = co2training.log, order = c(1, 1, 1), seasonal = list(order = c(1,
    1, 1), period = 12), method = "ML")

Coefficients:
         ar1      ma1    sar1     sma1
      0.2746  -0.6957  0.0289  -0.9331
s.e.  0.1394   0.1083  0.0847   0.1045

sigma^2 estimated as 6.536e-07:  log likelihood = 1214.13,  aic = -2418.27
```

Model 2 has lower AICc but model 1 is also more favorable based on the principle of parsimony and there is not much difference between the AIC's of model 1 and model 2. As a next step, I examine coefficients of both models.

We can observe the following:
* Model 1: Confidence intervals include zero for coefficients SAR(1)
* Model 2: Confidence intervals include zero for coefficients SAR(1)
Hence, I set these coefficients to 0, obtaining two additional models:
* Model 3: $SARIMA(0,1,1)(0,1,1)_{12}$
* Model 4: $SARIMA(1,1,1)(0,1,1)_{12}$
I fit these two revised models and look at their AICc to evaluate the model performance.
Based on their AIC scores, Model 3 and Model 4 perform better.



~~~
```{r}
#revised models
model3 = arima(co2training.log, order=c(0,1,1), seasonal = list(order = c(0,1,1), period = 12), method="ML")
model4=arima(co2training.log, order=c(1,1,1), seasonal = list(order = c(0,1,1), period = 12), method="ML")
```
~~~



~~~
#check AIC
model3
model4
```


Call:
arima(x = co2training.log, order = c(0, 1, 1), seasonal = list(order = c(0,
    1, 1), period = 12), method = "ML")

Coefficients:
          ma1     sma1
      -0.4669  -0.9026
s.e.   0.0732   0.0713

sigma^2 estimated as 6.736e-07:  log likelihood = 1212.46,  aic = -2418.92

Call:
arima(x = co2training.log, order = c(1, 1, 1), seasonal = list(order = c(0,
    1, 1), period = 12), method = "ML")

Coefficients:
         ar1      ma1     sma1
      0.2766  -0.6986  -0.9169
s.e.  0.1377   0.1063   0.0787

sigma^2 estimated as 6.583e-07:  log likelihood = 1214.07,  aic = -2420.15
~~~

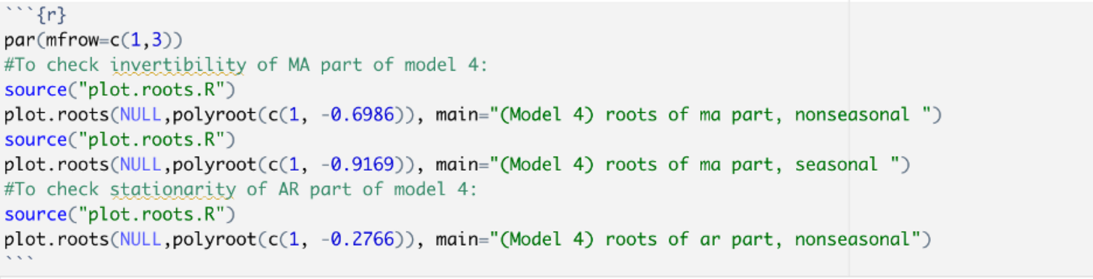
The following models are the selected primary models. :

* Model 3: $(1-B)(1-B^{12})X_t = (1-0.4669B)(1-0.9026B^{12})Z_t$

* Model 4: $(1-0.2766B)(1-B)(1-B^{12})X_t = (1-0.6986B)(1-0.9169B^{12})Z_t$

2.4.2 Diagnostic checking

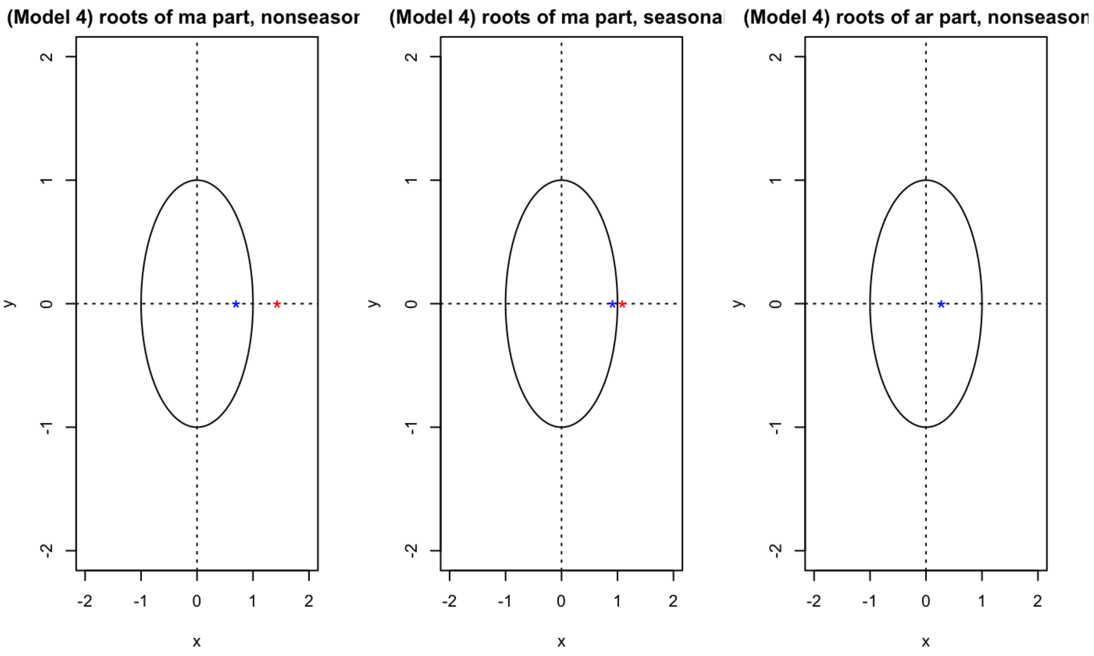
I will first examine the stationarity and invertibility of Model 4.

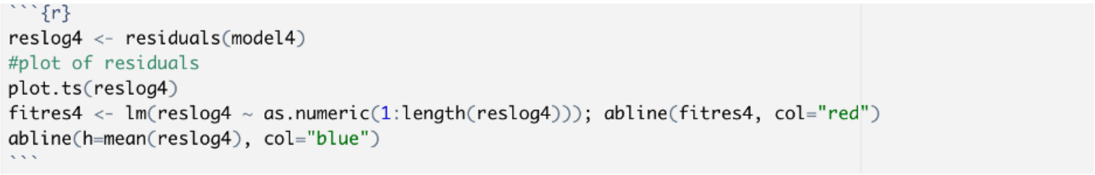


~~~
```{r}
par(mfrow=c(1,3))
#To check invertibility of MA part of model 4:
source("plot.roots.R")
plot.roots(NULL,polyroot(c(1, -0.6986)), main="(Model 4) roots of ma part, nonseasonal ")
source("plot.roots.R")
plot.roots(NULL,polyroot(c(1, -0.9169)), main="(Model 4) roots of ma part, seasonal ")
#To check stationarity of AR part of model 4:
source("plot.roots.R")
plot.roots(NULL,polyroot(c(1, -0.2766)), main="(Model 4) roots of ar part, nonseasonal")
```
~~~

Roots of AR (the root (in red) are outside of the unit circle but it does not show on the graph because it is outside the boundaries of this graph), MA and seasonal MA part lie outside the unit circle (Figure 14). Hence, model 4 is both stationary and invertible. I will follow with series of diagnostics and interpret their results below.

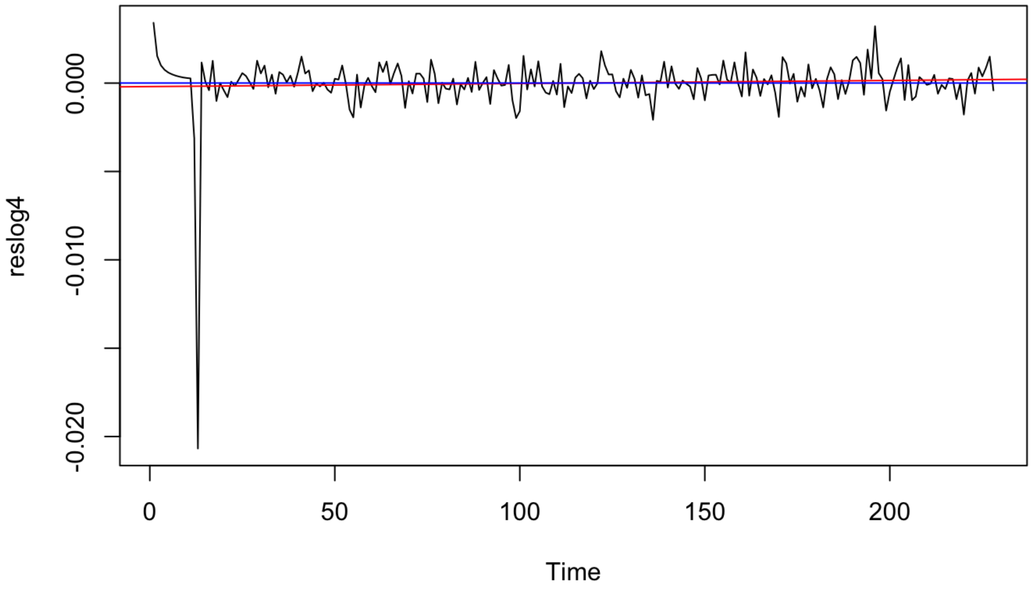
Figure 14: Polyroots of Model 4

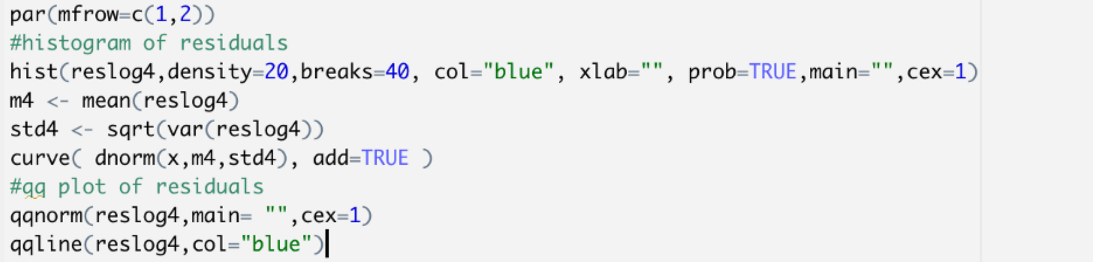




~~~
```{r}
reslog4 <- residuals(model4)
#plot of residuals
plot.ts(reslog4)
fitres4 <- lm(reslog4 ~ as.numeric(1:length(reslog4))); abline(fitres4, col="red")
abline(h=mean(reslog4), col="blue")
```
~~~

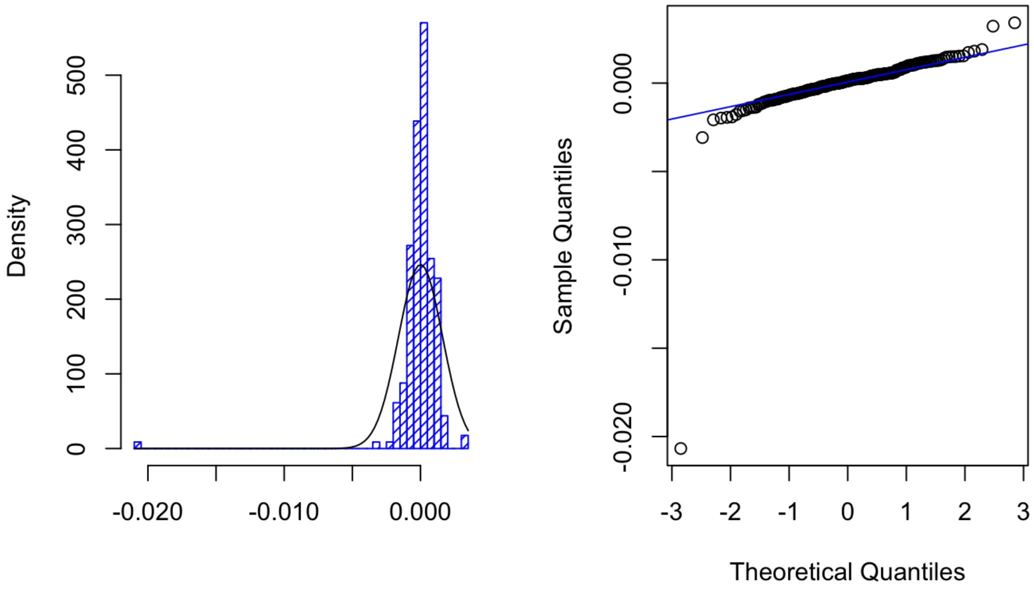
Figure 15: Plot of residuals (Model 4)

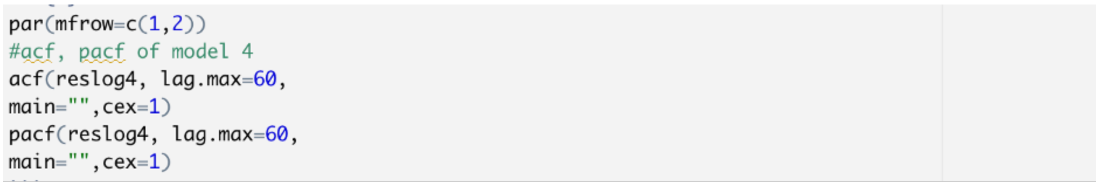




```
par(mfrow=c(1,2))
#histogram of residuals
hist(reslog4,density=20,breaks=40, col="blue", xlab="", prob=TRUE,main="",cex=1)
m4 <- mean(reslog4)
std4 <- sqrt(var(reslog4))
curve( dnorm(x,m4,std4), add=TRUE )
#qq plot of residuals
qqnorm(reslog4,main= "",cex=1)
qqline(reslog4,col="blue")
```

Figure 16: Histogram and Q-Q plot of residuals (Model 4)





```
par(mfrow=c(1,2))
#acf, pacf of model 4
acf(reslog4, lag.max=60,
main="",cex=1)
pacf(reslog4, lag.max=60,
main="",cex=1)
```

Figure 17: ACF and PACF of residuals (Model 4)

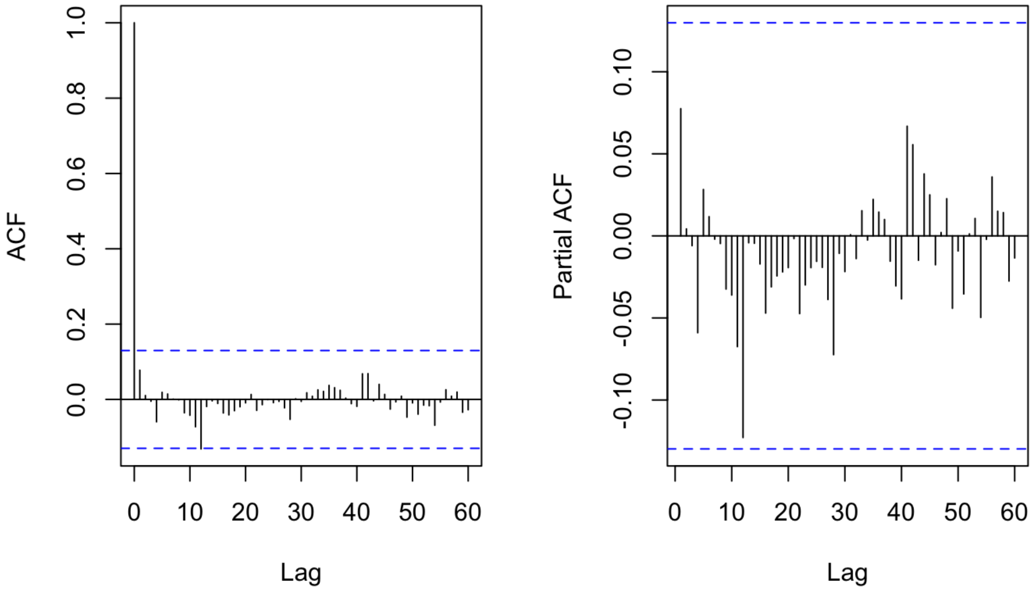
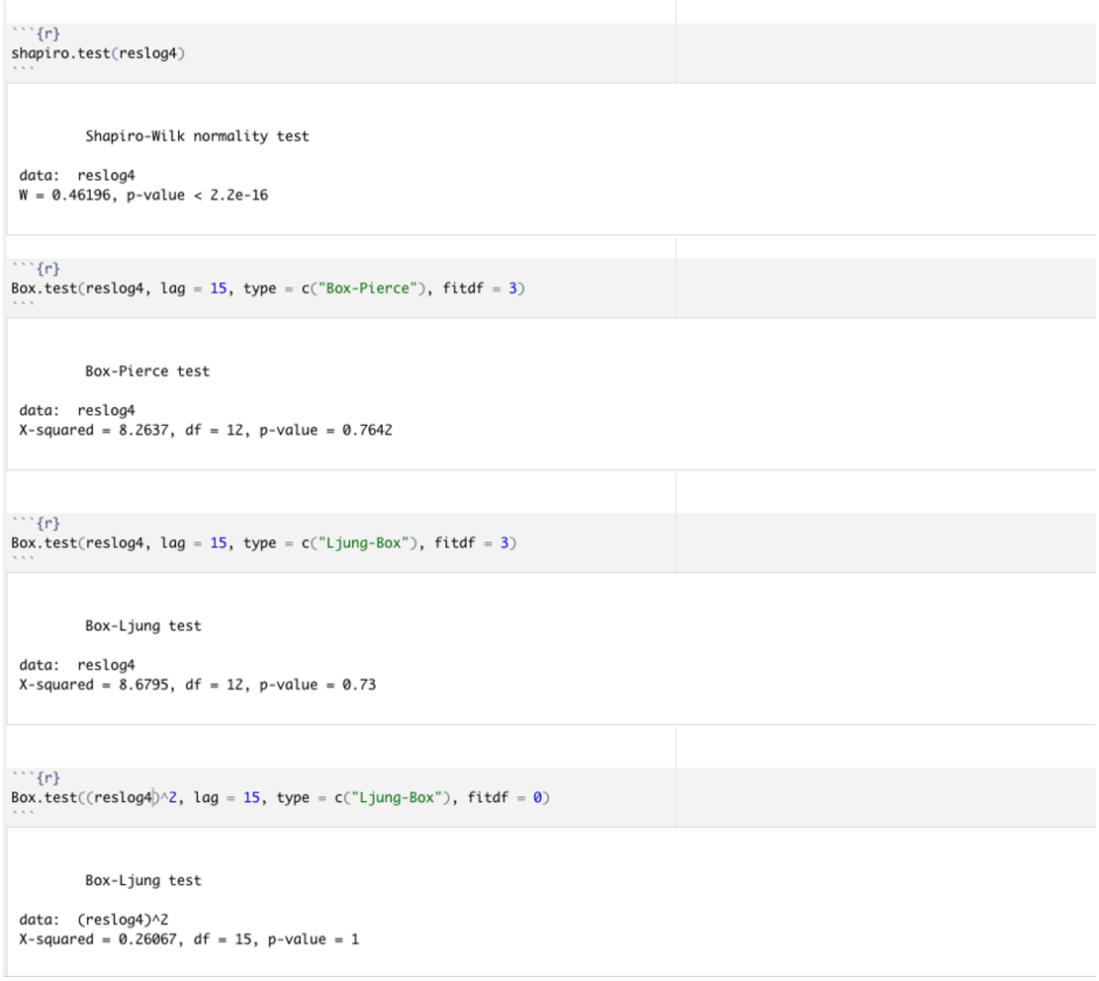


Figure 18: Diagnostic tests (Model 4)



~~~
```{r}
shapiro.test(reslog4)
~~~

        Shapiro-Wilk normality test

 data: reslog4
 W = 0.46196, p-value < 2.2e-16

```{r}
Box.test(reslog4, lag = 15, type = c("Box-Pierce"), fitdf = 3)
```

        Box-Pierce test

 data: reslog4
 X-squared = 8.2637, df = 12, p-value = 0.7642

```{r}
Box.test(reslog4, lag = 15, type = c("Ljung-Box"), fitdf = 3)
```

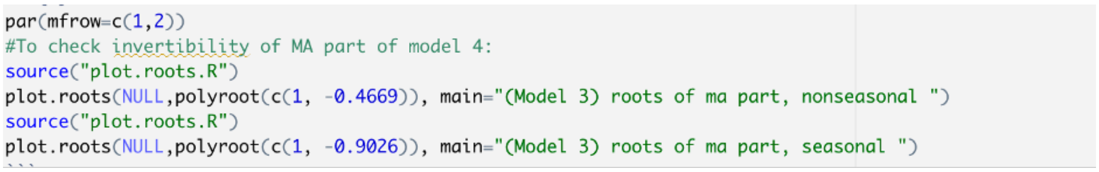
        Box-Ljung test

 data: reslog4
 X-squared = 8.6795, df = 12, p-value = 0.73

```{r}
Box.test((reslog4)^2, lag = 15, type = c("Ljung-Box"), fitdf = 0)
```

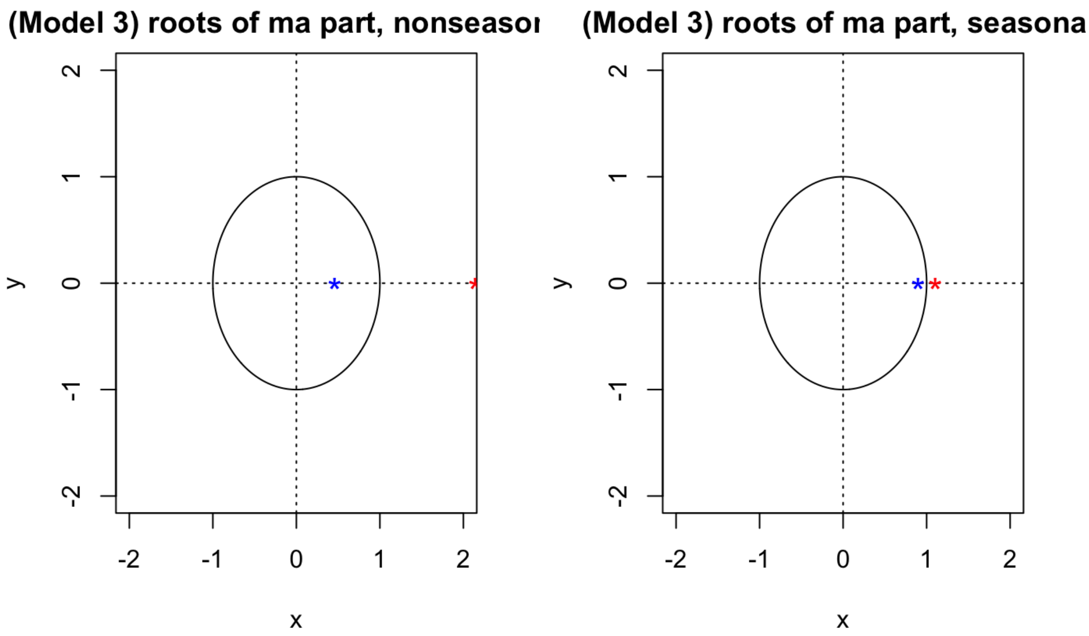
        Box-Ljung test

 data: (reslog4)^2
 X-squared = 0.26067, df = 15, p-value = 1
```

The results of a series of diagnostic test show:

- Plot of residuals demonstrates no trend, no visible change of variance, no seasonality
- Histogram and QQ plot suggest that residuals are not normal. Specifically, even though most residuals are on the line of QQ plot, there are several points that are not on the line, which indicates a heavy tail distribution.
- All ACF and PACF of residuals are within confidence intervals and can be counted as zeros - no additional coefficients are necessary.
- Shapiro-Wilk normality test failed to pass. Based on this test, I conclude that residuals are not normally distributed.
- Box-Pierce test is passed with p-value of 0.7642. Box-Ljung test is passed with p-value of 0.73. McLeod-Li test is passed with p-value of 1.

Therefore, Model 4 passed all the tests, except Shapiro.test and can be used for forecasting.

Next, I perform the same diagnostic tests for Model 3.

Roots of MA and seasonal MA part lie outside the unit circle(Figure 19) - hence model 3 is both stationary and invertible. I will follow with series of diagnostics and their interpretation below.
```



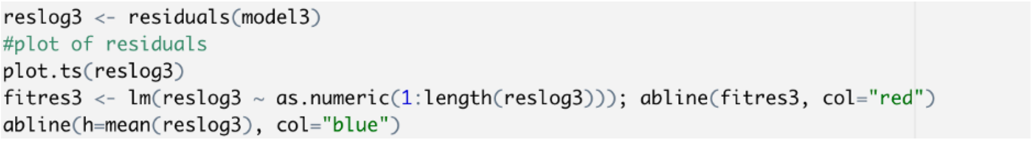
```
par(mfrow=c(1,2))
#To check invertibility of MA part of model 4:
source("plot.roots.R")
plot.roots(NULL,polyroot(c(1, -0.4669)), main="(Model 3) roots of ma part, nonseasonal ")
source("plot.roots.R")
plot.roots(NULL,polyroot(c(1, -0.9026)), main="(Model 3) roots of ma part, seasonal ")
```

Figure 19: Polyroots of Model 3



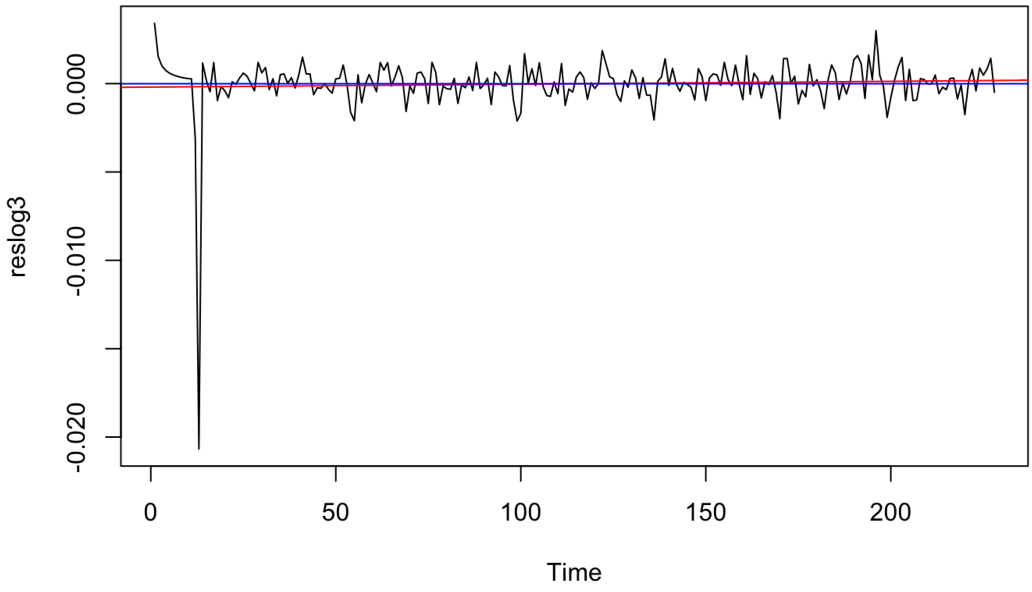
(Model 3) roots of ma part, nonseasoı

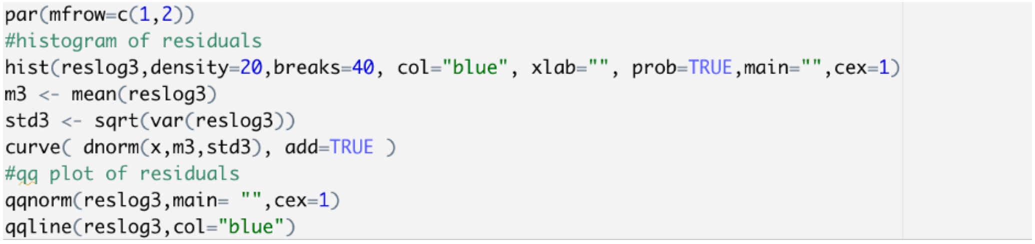
(Model 3) roots of ma part, seasona



```
reslog3 <- residuals(model3)
#plot of residuals
plot.ts(reslog3)
fitres3 <- lm(reslog3 ~ as.numeric(1:length(reslog3))); abline(fitres3, col="red")
abline(h=mean(reslog3), col="blue")
```

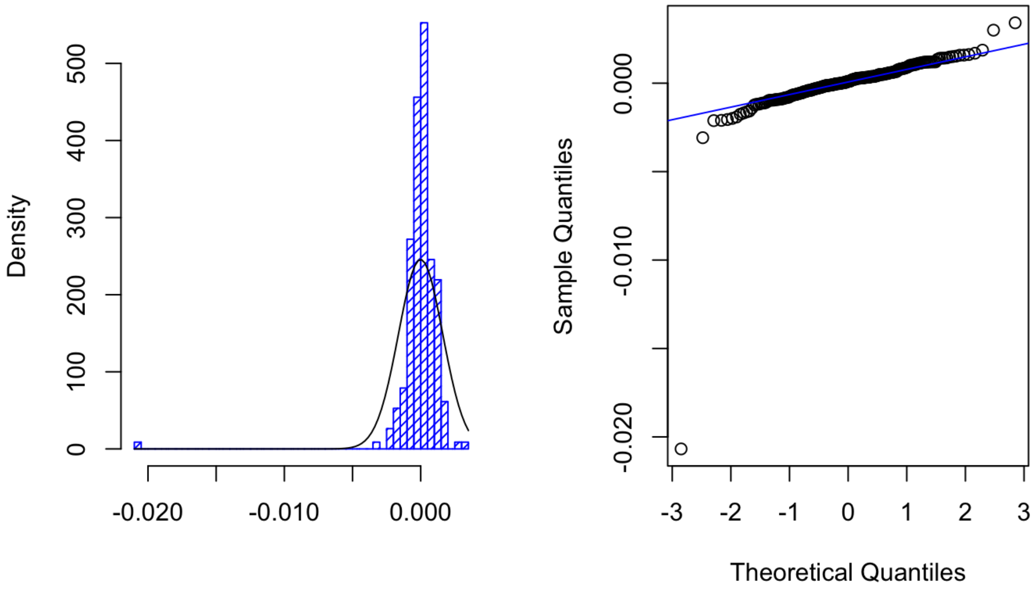
Figure 20: Plot of residuals (Model 3)

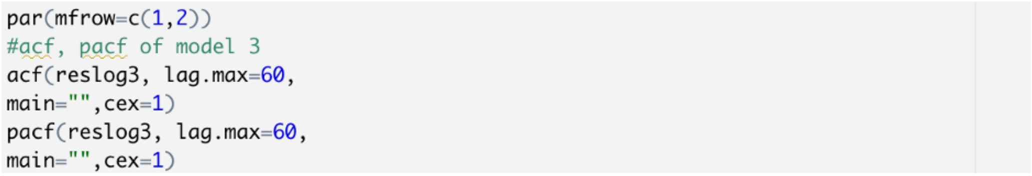




```
par(mfrow=c(1,2))
#histogram of residuals
hist(reslog3,density=20,breaks=40, col="blue", xlab="", prob=TRUE,main="",cex=1)
m3 <- mean(reslog3)
std3 <- sqrt(var(reslog3))
curve( dnorm(x,m3,std3), add=TRUE )
#qq plot of residuals
qqnorm(reslog3,main= "",cex=1)
qqline(reslog3,col="blue")
```

Figure 21: Histogram and Q-Q plot of residuals (Model 3)





```
par(mfrow=c(1,2))
#acf, pacf of model 3
acf(reslog3, lag.max=60,
main="",cex=1)
pacf(reslog3, lag.max=60,
main="",cex=1)
```

Figure 22: ACF and PACF of residuals (Model 3)

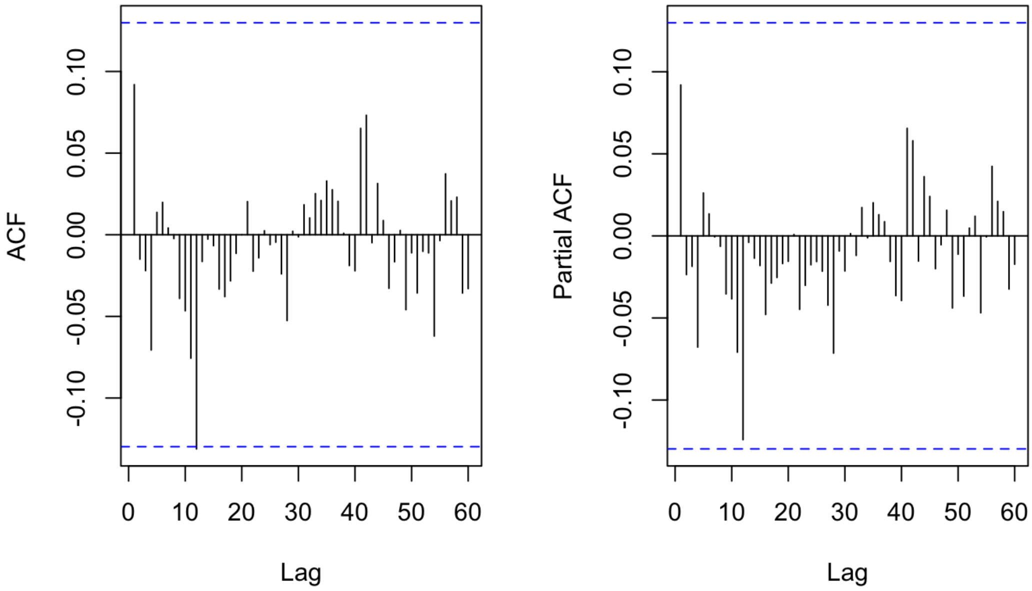
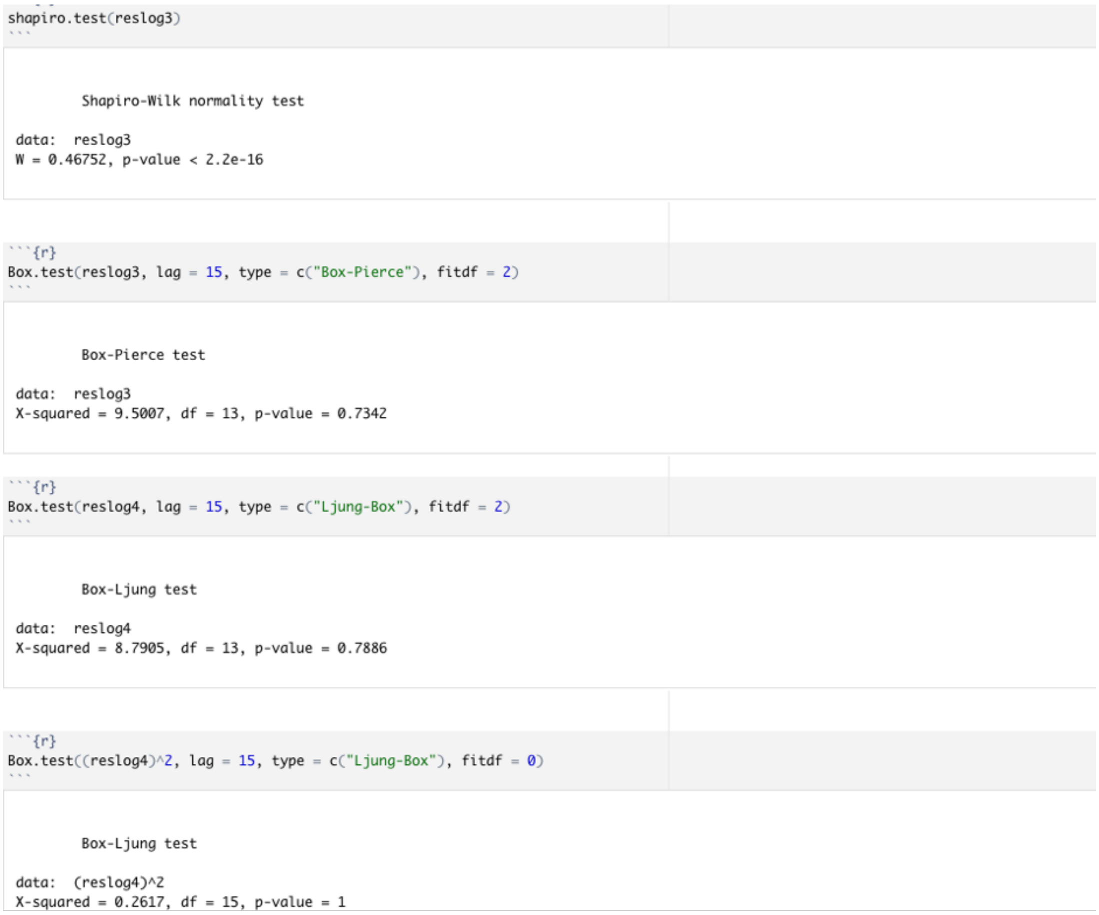


Figure 23: Diagnostic tests (Model 3)



```
shapiro.test(reslog3)
```

```

        Shapiro-Wilk normality test

 data:  reslog3
 W = 0.46752, p-value < 2.2e-16
```

```{r}
Box.test(reslog3, lag = 15, type = c("Box-Pierce"), fitdf = 2)
```

```

        Box-Pierce test

 data:  reslog3
 X-squared = 9.5007, df = 13, p-value = 0.7342
```

```{r}
Box.test(reslog4, lag = 15, type = c("Ljung-Box"), fitdf = 2)
```

```

        Box-Ljung test

 data:  reslog4
 X-squared = 8.7905, df = 13, p-value = 0.7886
```

```{r}
Box.test((reslog4)^2, lag = 15, type = c("Ljung-Box"), fitdf = 0)
```

```

        Box-Ljung test

 data:  (reslog4)^2
 X-squared = 0.2617, df = 15, p-value = 1
```

Diagnostic checkings shows:

- Plot of residuals demonstrates no trend, no visible change of variance, no seasonality
- Histogram and QQ plot suggest that residuals are not exactly normal. Specifically, even though most residuals are on the line of QQ plot, there are several points that are not on the line, which indicates a heavy tail distribution.
- All ACF and PACF of residuals are within confidence intervals and can be counted as zeros - no additional coefficients are necessary.
- Shapiro-Wilk normality test fail to pass. Based on this test, I conclude that residuals are not normally distributed.
- Box-Pierce test is passed with p-value of 0.7342. Box-Ljung test is passed with p-value of 0.7886. McLeod-Li test is passed with p-value of 1.

Therefore, similar to Model 4, Model 3 passed all the tests, except Shapiro.test and can be used for forecasting.

2.4.3 Model Selection

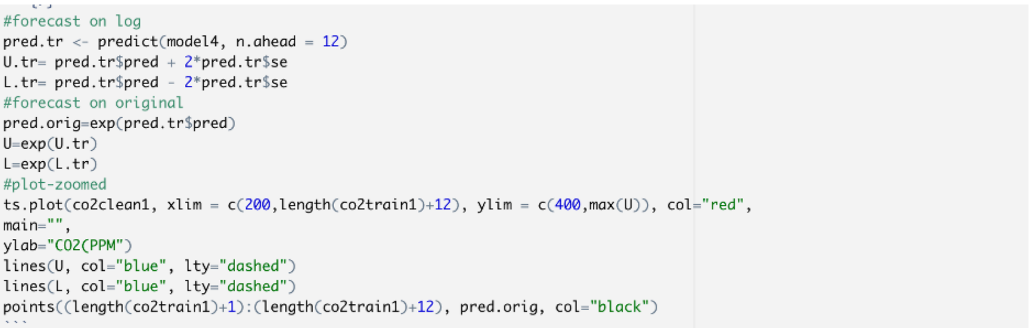
Both Model 3 and Model 4 passed all diagnostic checks, except the Shapiro-Wilk normality test. However, since Model 4 has lower AICc than Model3 (-2430.15 vs -2418.92), while Model 3 is better from a parsimony perspective (2 coefficients vs 3 coefficients),

Model selected with AICc is different from models suggested by ACF/PACF of the data, as it does not include a seasonal AR component and has one less SAR coefficient.
I will use Model 4 for forecasting.

$$\text{Model 4: } (1-0.2766B)(1-B)(1-B^{12})X_t = (1-0.6986B)(1-0.9169B^{12})Z_t$$

2.5 Forecasting

Figure 24 below demonstrates forecast on original data using the model. Red line corresponds to the original data, dots correspond to predictions from the model, and blue lines correspond to confidence intervals for predictions. From the figure we can see that original data is within confidence intervals for predictions, which means that the model performs well.



```
#forecast on log
pred.tr <- predict(model4, n.ahead = 12)
U.tr= pred.tr$pred + 2*pred.tr$se
L.tr= pred.tr$pred - 2*pred.tr$se
#forecast on original
pred.orig=exp(pred.tr$pred)
U=exp(U.tr)
L=exp(L.tr)
#plot-zoomed
ts.plot(co2clean1, xlim = c(200,length(co2train1)+12), ylim = c(400,max(U)), col="red",
main="",
ylab="CO2(PPM")
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(co2train1)+1):(length(co2train1)+12), pred.orig, col="black")
```

Figure 24: 1 year forecast with original data - zoomed

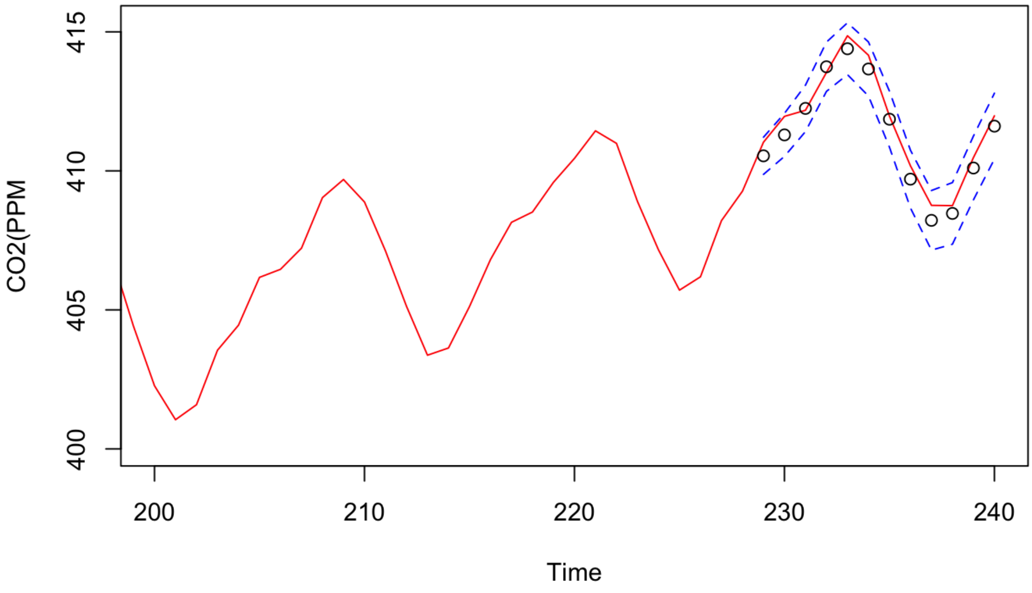
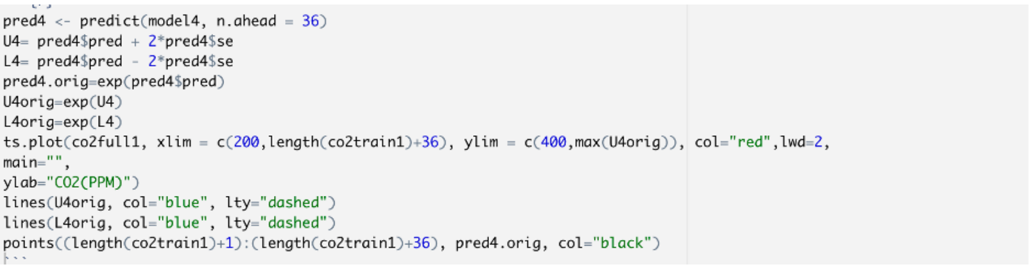
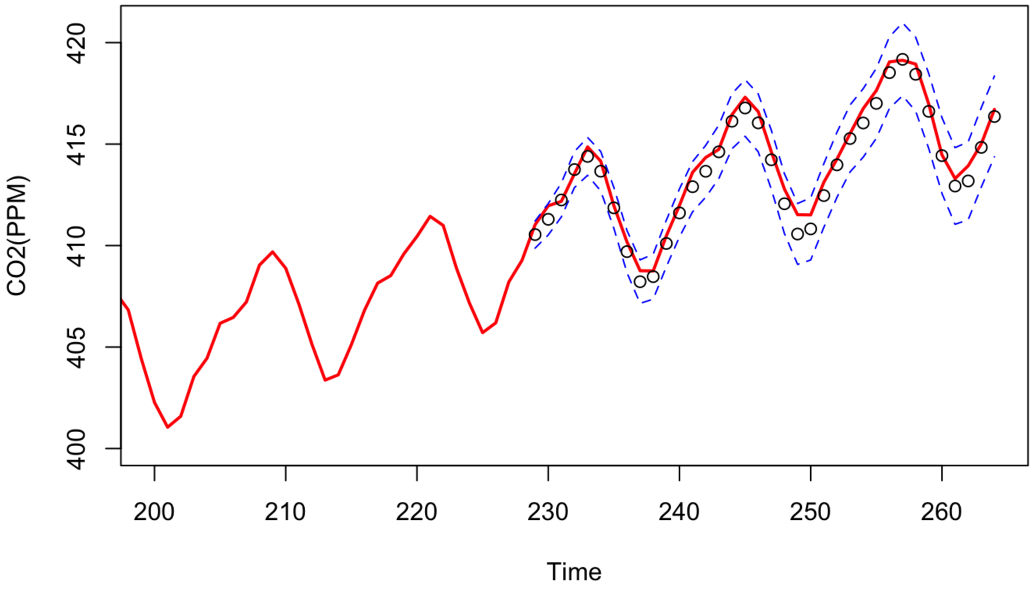


Figure 25 below demonstrates similar 3 year forecast.This graph illustrates the difference between predictions in a hypothetical situation when COVID did not occur vs the real situation where the world experienced the COVID-19 pandemic and its impacts.



```
pred4 <- predict(model4, n.ahead = 36)
U4= pred4$pred + 2*pred4$se
L4= pred4$pred - 2*pred4$se
pred4.orig=exp(pred4$pred)
U4orig=exp(U4)
L4orig=exp(L4)
ts.plot(co2full1, xlim = c(200,length(co2train1)+36), ylim = c(400,max(U4orig)), col="red",lwd=2,
main="",
ylab="CO2(PPM)")
lines(U4orig, col="blue", lty="dashed")
lines(L4orig, col="blue", lty="dashed")
points((length(co2train1)+1):(length(co2train1)+36), pred4.orig, col="black")
```

Figure 25: 3 year forecast with original data - zoomed



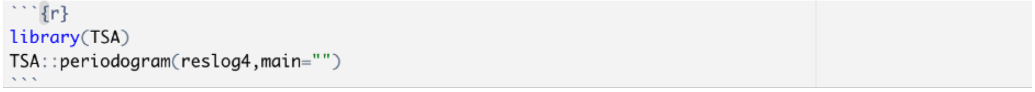
As we can see in the Figure 25, the trajectory of the rising level of CO_2 has not been decreased due to the pandemic which suggests that the expectation of possible positive impact of COVID-19 on the level of CO_2 was not actualized. It should be noted that as expected confidence intervals are increasing with longer forecasts, which means higher degree of uncertainty.

Overall, it shows that even though people had to lock down and consume less, emit less, the atmospheric CO_2 had continued to increase during the course of the COVID-19 pandemic.

2.6 Spectral Analysis

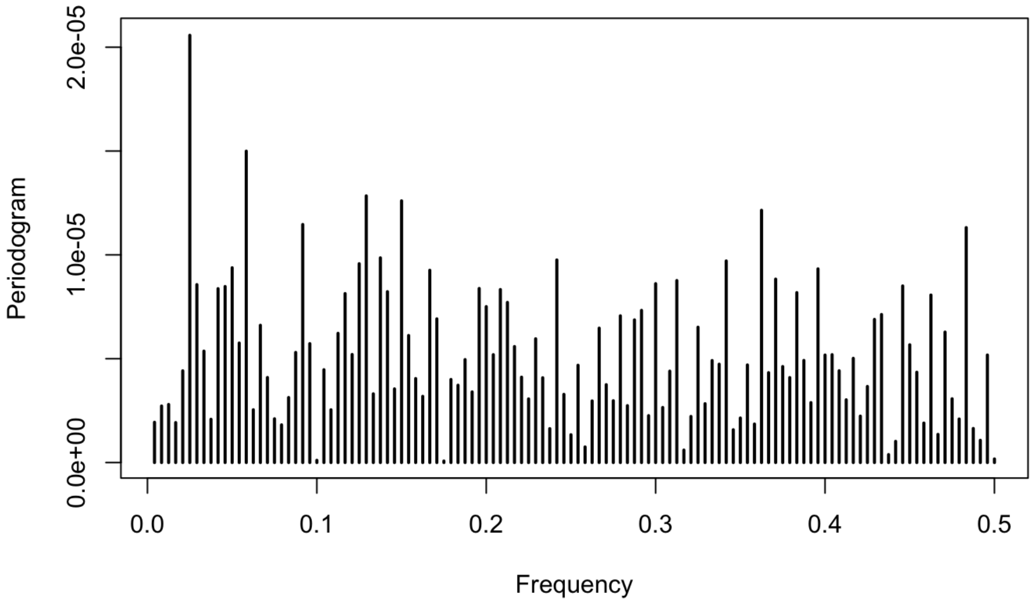
The spectral analysis of the residuals of my selected model. As evident from the periodogram of residuals below (Figure 26), there is no periodicity of residuals observed.

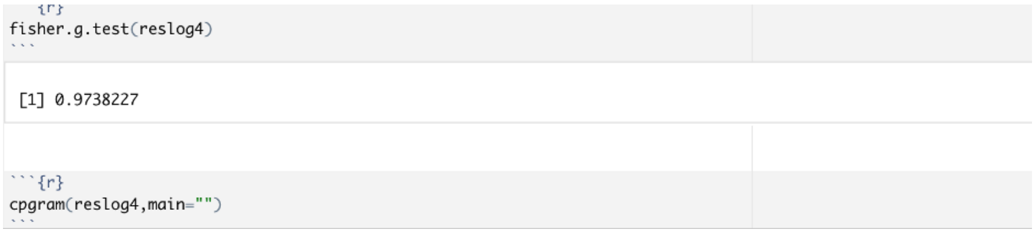
Additionally, I perform Fisher's and Kolmogorov-Smirnov's tests. Fisher's test yields p-value of 0.9738227, therefore we fail to reject the null hypothesis that residuals are Gaussian white noise. For Kolmogorov-Smirnov's test, the cumulative periodogram is always within the boundaries (Figure 27), hence we once again fail to reject the null hypothesis. We can conclude that residuals pass these tests.



~~~~
```{r}
library(TSA)
TSA::periodogram(reslog4,main="")
```
~~~~

Figure 26: Peridogram of residuals (Model 4)



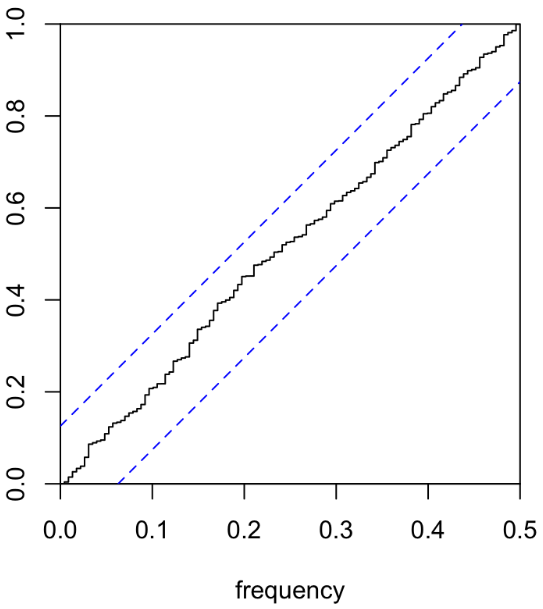


~~~~
{r}
fisher.g.test(reslog4)
```

[1] 0.9738227

```{r}
cpgram(reslog4,main="")
```
~~~~

Figure 27: Cumulative Peridogram of residuals (Model 4)



3 Conclusion

This project aims to fit an appropriate model for the forecasting of the atmospheric CO_2 data. This goal was achieved by using log-transformed version of the data between 2000/2019, and fitting a SARIMA model with the formula $(1-0.2766B)(1-B)(1-B^{12})X_t = (1-0.6986B)(1-0.9169B^{12})Z_t$. The model passed all diagnostics, expect the Shapiro normality test, including spectral analysis.

The model was used to predict 1 year ahead, where original data fell withing prediction intervals, validating the choice of the model. Then, model was used to predict 3 years ahead, this prediction goes against the expectation that the COVID-19 pandemic will change the course of climate change by helping reduce the $C0_2$ emission.

References

[1] B. Weir *et al.*, “Regional impacts of covid-19 on carbon dioxide detected worldwide from space,” *Science Advances*, vol. 7, no. 45, pp. 1–9, 2021.

4 Appendix

PSTAT274 Final Project

Selin Karabulut

```
setwd("~/Desktop/FALL 2022/PSTAT274/Final Project")
knitr::opts_chunk$set(
  error = TRUE, # do not interrupt generation in case of errors,
  echo = TRUE  # show R code
)
```

```
#load libraries
library(ggfortify)
```

```
## Loading required package: ggplot2
```

```
library(tseries)
```

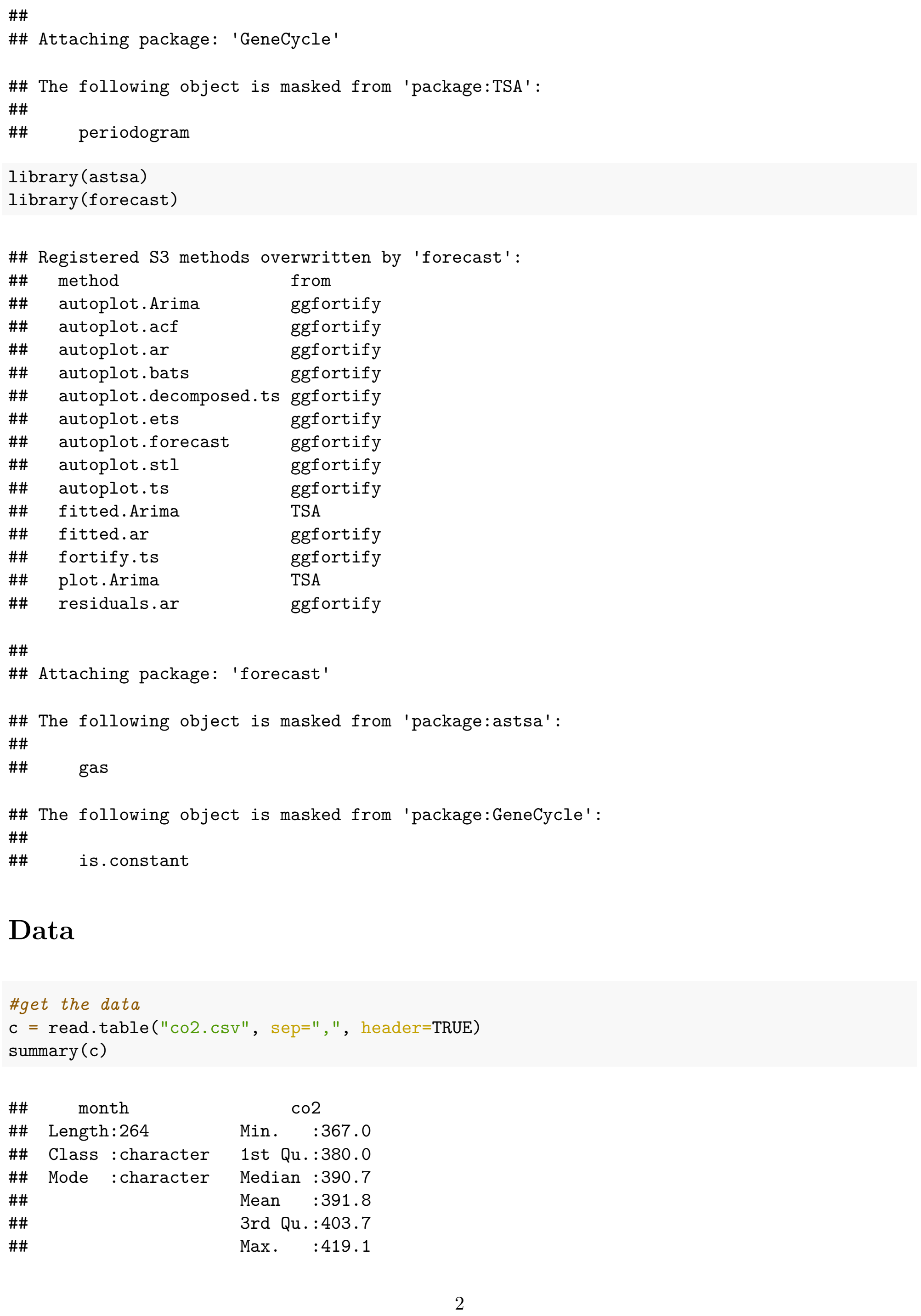
```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(TSA)
```

```
##
## Attaching package: 'TSA'

## The following objects are masked from 'package:stats':
##
##     acf, arima

## The following object is masked from 'package:utils':
##
##     tar
```

```
library(GeneCycle)
```

```
## Loading required package: MASS

## Loading required package: longitudinal

## Loading required package: corpcor

## Loading required package: fdrtool
```



```
##
## Attaching package: 'GeneCycle'

## The following object is masked from 'package:TSA':
##
##     periodogram
```

```
library(astsa)
library(forecast)
```

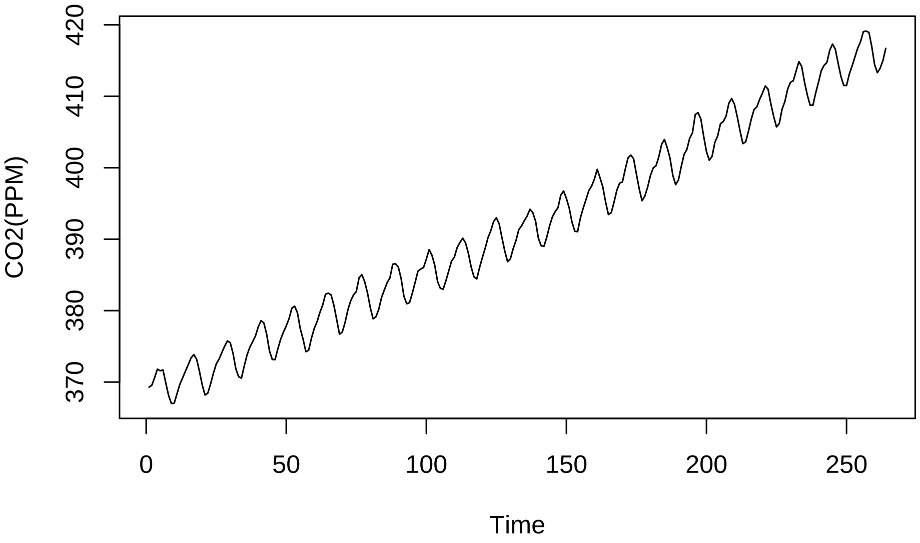
```
## Registered S3 methods overwritten by 'forecast':
##   method                 from
##   autoplot.Arima         ggfortify
##   autoplot.acf           ggfortify
##   autoplot.ar            ggfortify
##   autoplot.bats          ggfortify
##   autoplot.decomposed.ts ggfortify
##   autoplot.ets    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

##
## Attaching package: 'forecast'

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

## The following object is masked from 'package:GeneCycle':
##
##     is.constant
```

Data

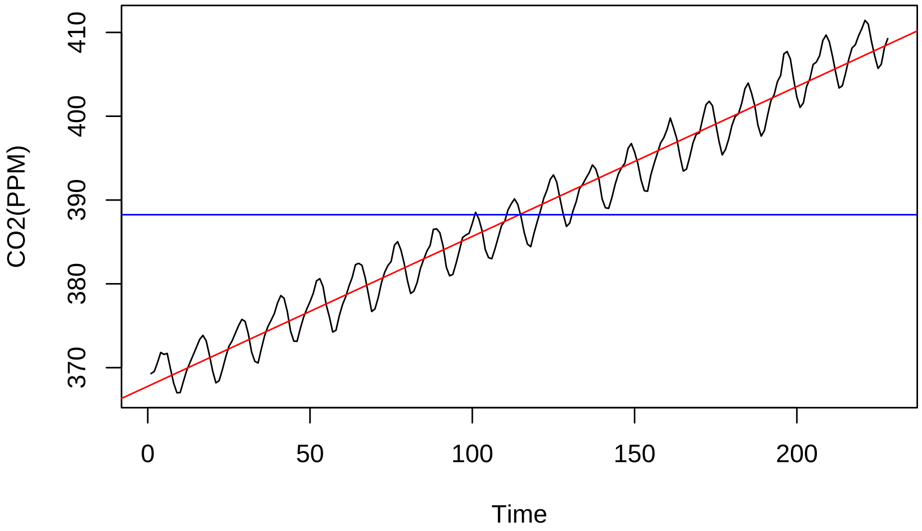
```
#get the data
c = read.table("co2.csv", sep=",", header=TRUE)
summary(c)
```

```
##     month                co2
##  Length:264         Min.   :367.0
##  Class :character   1st Qu.:380.0
##  Mode  :character   Median :390.7
##                     Mean   :391.8
##                     3rd Qu.:403.7
##                     Max.   :419.1
```

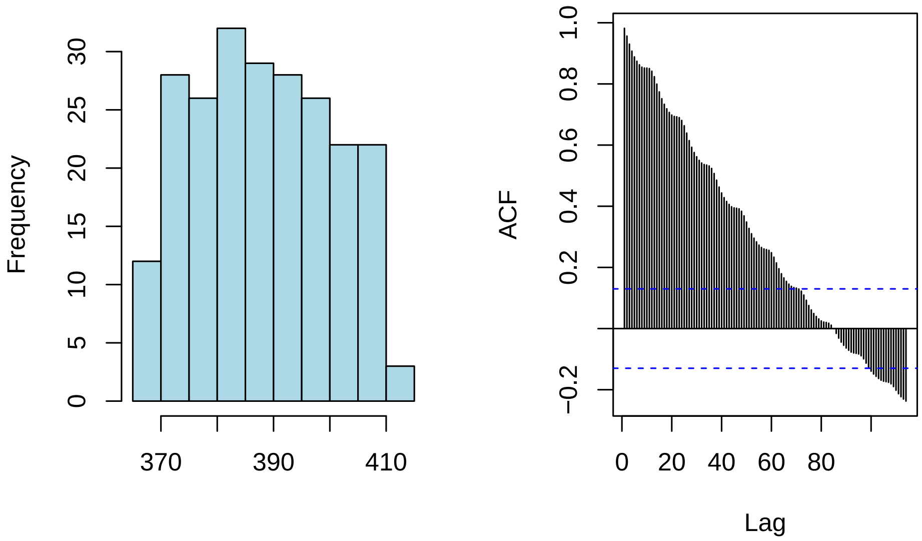
```
co = ts(c[,2])
ts.plot(co, ylab="CO2(PPM)")
```



```
#partition data
co2training <- c[1:228, ]
co2clean <- c[1:240, ]
co2full <- c[1:264, ]
#plot data
ts.plot(co2training$co2,ylab="CO2(PPM)")
nt=length(co2training$co2)
fit <- lm(co2training$co2 ~ as.numeric(1:nt)); abline(fit, col="red")
abline(h=mean(co2training$co2), col="blue")
```



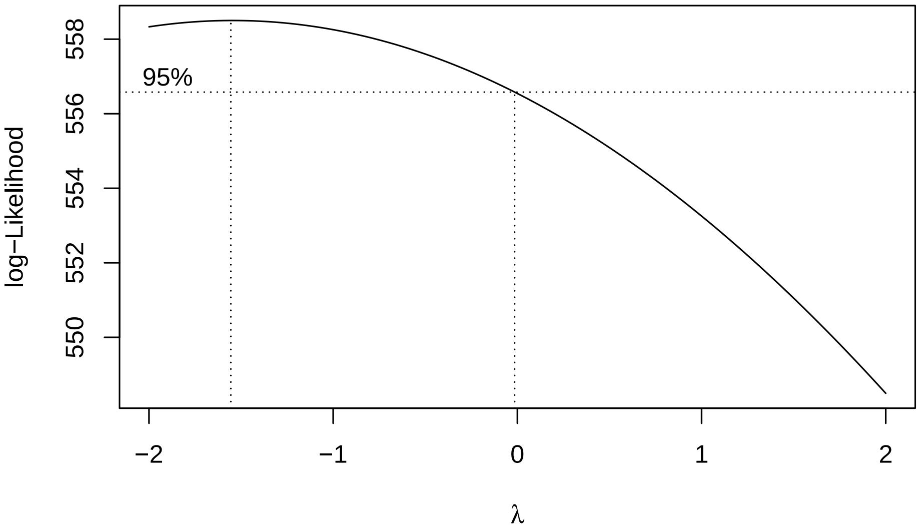
```
#hist and acf of training data
par(mfrow=c(1,2))
hist(co2training$co2, main="",col="light blue", xlab="")
acf(co2training$co2,lag.max=114,main="")
```



Transformations and Stationarity

Box-Cox

```
#transformations
# To choose parameter  of the Box-Cox transformation for datset # Box-Cox transformation:
t = 1:length(co2training$co2)
fit = lm(co2training$co2 ~ t)
bcTransform = boxcox(co2training$co2 ~ t, plotit=TRUE)
```

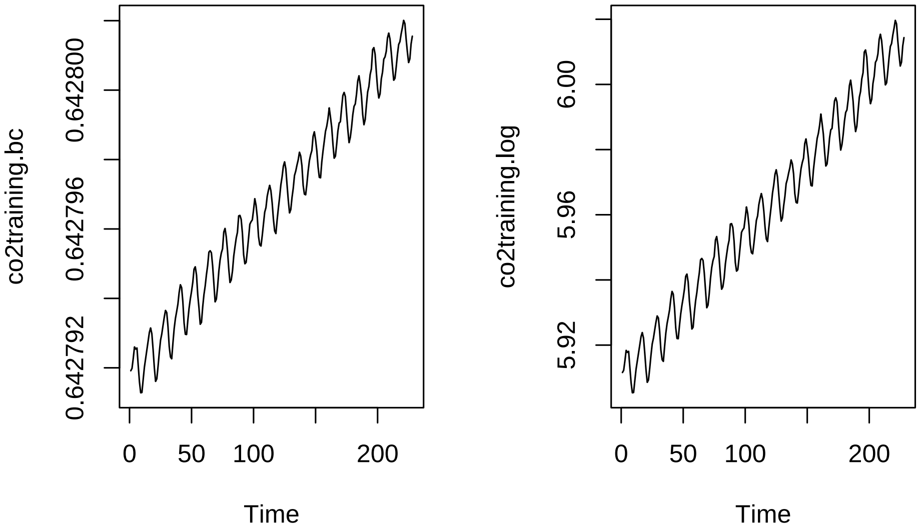


```
lambda=bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
lambda
```

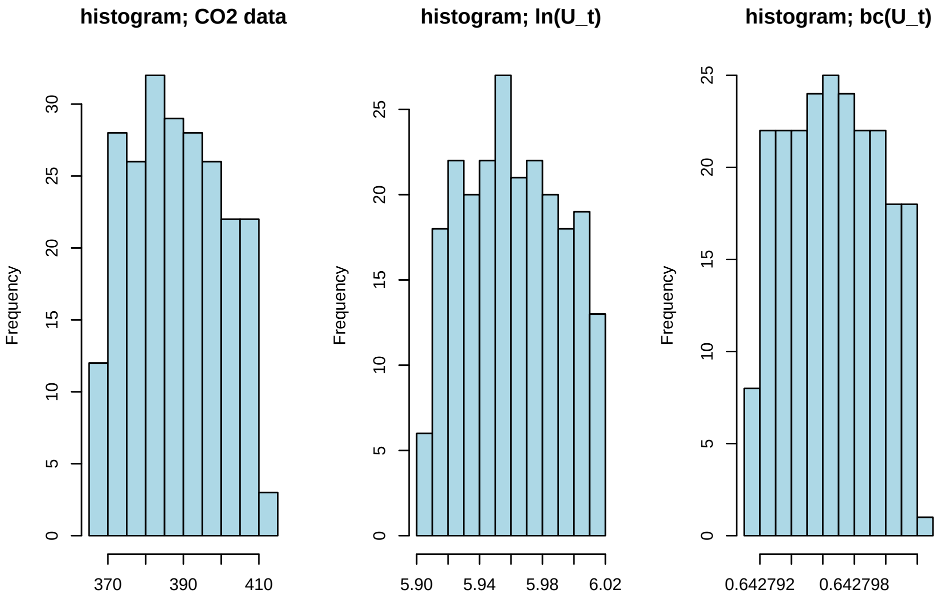
```
## [1] -1.555556
```

Log

```
par(mfrow=c(1,2))
co2training.bc = (1/lambda)*(co2training$co2^lambda-1)
co2training.log <- log(co2training$co2)
plot.ts(co2training.bc)
plot.ts(co2training.log)
```

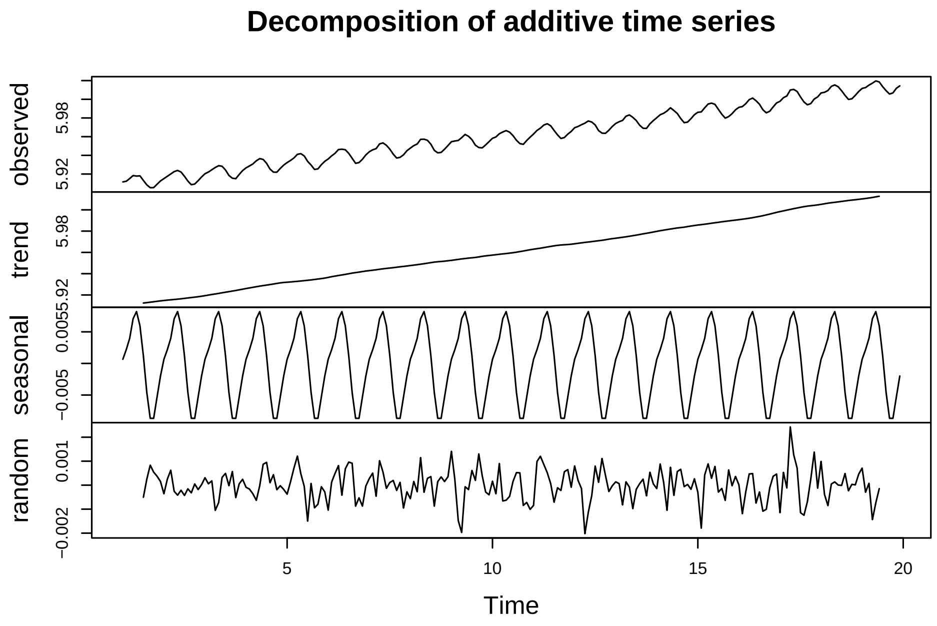


```
par(mfrow=c(1,3))
hist(co2training$co2, col="light blue", xlab="", main="histogram; CO2 data")
hist(co2training.log, col="light blue", xlab="", main="histogram; ln(U_t)")
hist(co2training.bc, col="light blue", xlab="", main="histogram; bc(U_t)")
```



```
#log transform gave a more symmetric histogram and more even variance
```

```
#Decomposition of ln(U_t)
library(ggplot2)
library(ggfortify)
y <- ts(as.ts(co2training.log), frequency = 12)
decomp <- decompose(y)
plot(decomp)
```



```
#check variance
var(co2training.bc)
```

```
## [1] 8.418503e-12
```

```
var(co2training.log)
```

```
## [1] 0.000955611
```

Differencing

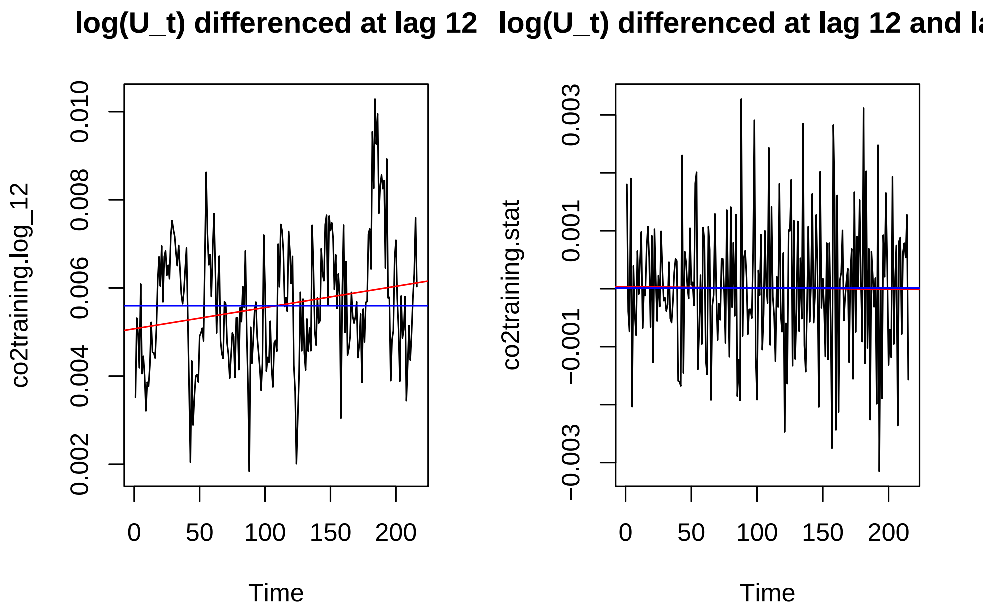
```
par(mfrow=c(1,2))
#differencing at lag 12
co2training.log_12 <- diff(co2training.log, lag=12)
plot.ts(co2training.log_12 , main="log(U_t) differenced at lag 12")
fit <- lm(co2training.log_12~ as.numeric(1:length(co2training.log_12))); abline(fit, col="red")
mean(co2training.log_12)
```

```
## [1] 0.00559652
```

```
abline(h=mean(co2training.log_12), col="blue")
#seasonality no longer apparent
# trend is still there
#variance got smaller
#differencing at lag 12 and 1
co2training.stat <- diff(co2training.log_12, lag=1)
plot.ts(co2training.stat, main="log(U_t) differenced at lag 12 and lag 1")
fit <- lm(co2training.stat ~ as.numeric(1:length(co2training.stat))); abline(fit, col="red")
mean(co2training.stat)
```

```
## [1] 1.169652e-05
```

```
abline(h=mean(co2training.stat), col="blue")
```



```
#no trend, no seasonality
```

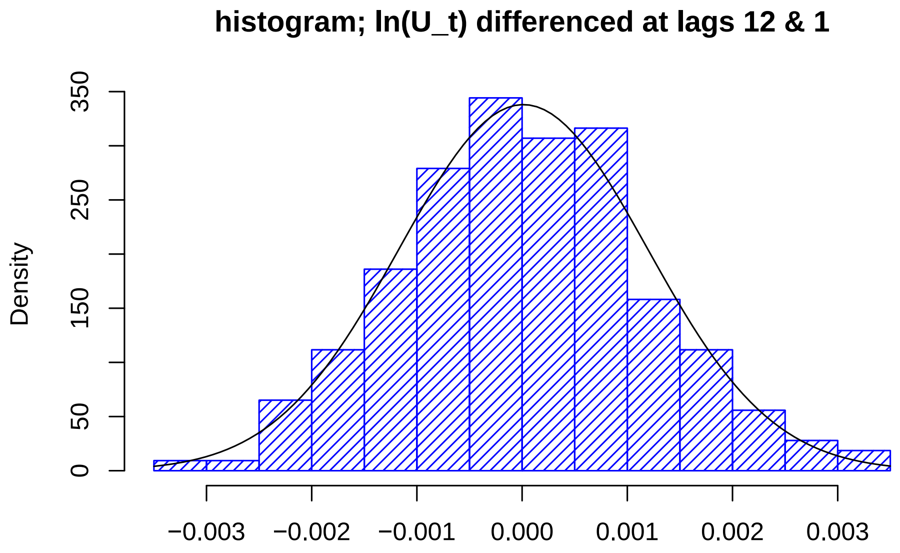
```
#check variance
var(co2training.log_12)
```

```
## [1] 2.050716e-06
```

```
var(co2training.stat)
```

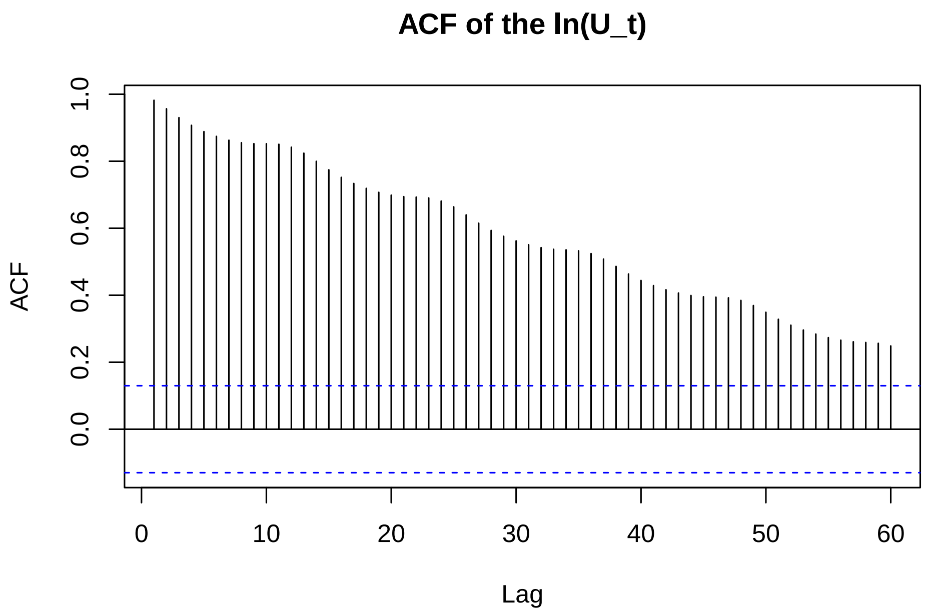
```
## [1] 1.392988e-06
```

```
#histogram
hist(co2training.stat, density=20,breaks=20, col="blue", xlab="", prob=TRUE, main="histogram; ln(U_t) d
m<-mean(co2training.stat)
std<- sqrt(var(co2training.stat))
curve( dnorm(x,m,std), add=TRUE )
```

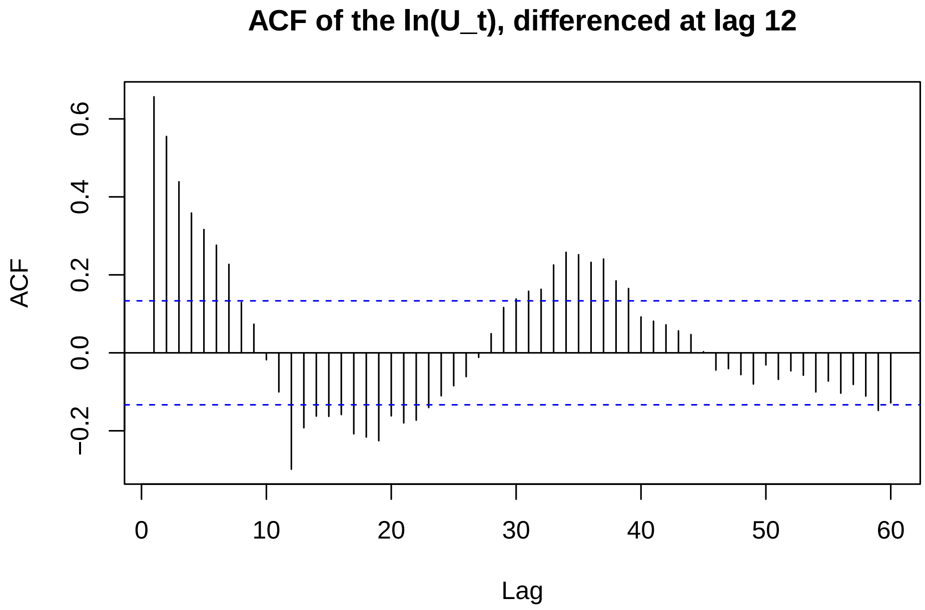


Preliminary model identification

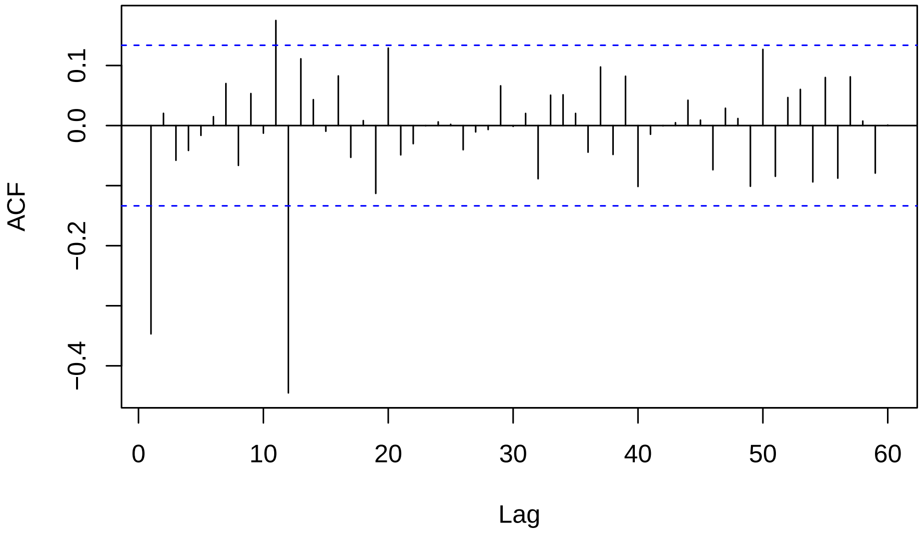
```
#check acf of the transformed and differenced
acf(co2training.log, lag.max=60, main="ACF of the ln(U_t)")
```


ACF of the ln(U_t)

```
acf(co2training.log_12, lag.max=60, main="ACF of the ln(U_t), differenced at lag 12")
```

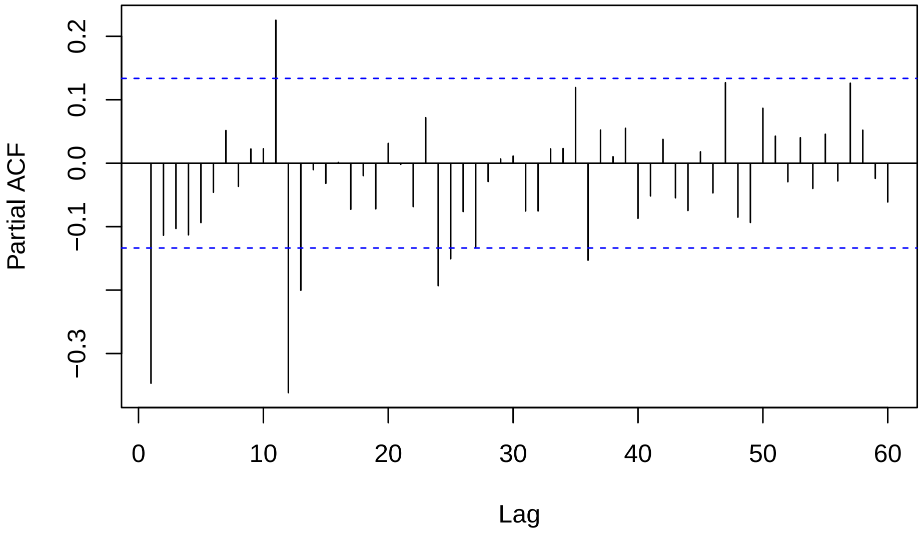

ACF of the ln(U_t), differenced at lag 12

```
acf(co2training.stat, lag.max=60, main="")
```



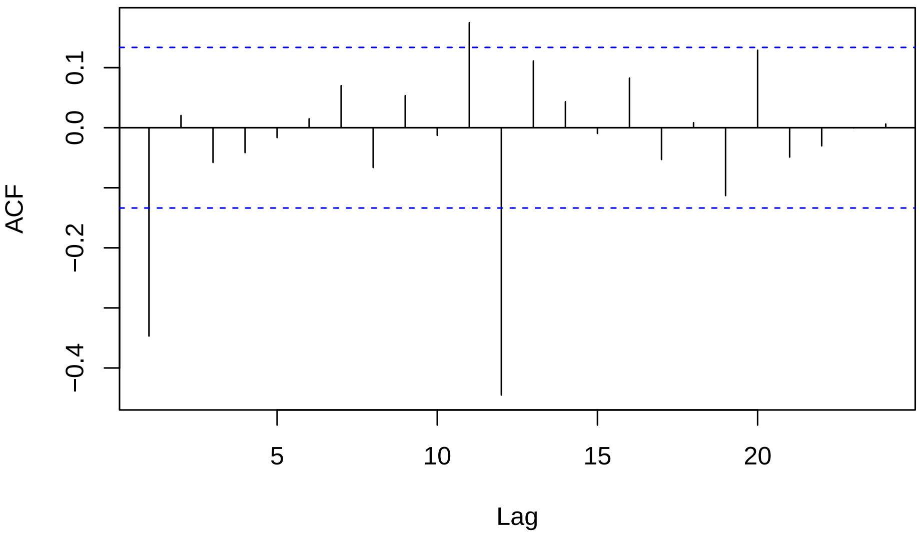
```
# ACF outside confidence intervals: Lags 1, maybe 11, 12
```

```
#pacf
pacf(co2training.stat, lag.max=60, main="")
```

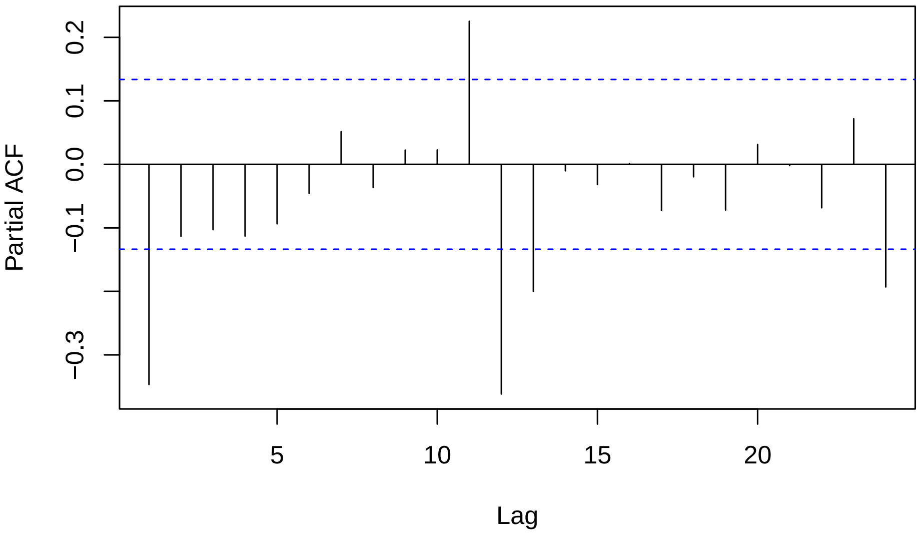


```
# PACF outside confidence intervals: Lags 1, 11, 12, 13, 24, 25,36
```

```
# closer inspection
acf(co2training.stat, lag.max=24, main="")
```



```
# closer inspection
pacf(co2training.stat, lag.max=24, main="")
```



#Model fitting

List of candidate models to try: s=12, d=1 D=1 p=0,1 P=1 q=1 Q=1

```
#candidate models
model1 = arima(co2training.log, order=c(0,1,1), seasonal = list(order = c(1,1,1), period = 12), method=
model2=arima(co2training.log, order=c(1,1,1), seasonal = list(order = c(1,1,1), period = 12), method="M
```

```
#check AIC
model1
```

```
##
## Call:
## arima(x = co2training.log, order = c(0, 1, 1), seasonal = list(order = c(1,
##     1, 1), period = 12), method = "ML")
##
## Coefficients:
##           ma1    sar1     sma1
##       -0.4645  0.0417  -0.9249
## s.e.   0.0731  0.0852   0.0957
##
## sigma^2 estimated as 6.676e-07:  log likelihood = 1212.58,  aic = -2419.16
```

```
model2
```

```
##
## Call:
## arima(x = co2training.log, order = c(1, 1, 1), seasonal = list(order = c(1,
##     1, 1), period = 12), method = "ML")
##
## Coefficients:
##          ar1      ma1    sar1     sma1
##       0.2746  -0.6957  0.0289  -0.9331
## s.e.  0.1394   0.1083  0.0847   0.1045
##
## sigma^2 estimated as 6.536e-07:  log likelihood = 1214.13,  aic = -2420.27
```

```
#revised models
model3 = arima(co2training.log, order=c(0,1,1), seasonal = list(order = c(0,1,1), period = 12), method=
model4=arima(co2training.log, order=c(1,1,1), seasonal = list(order = c(0,1,1), period = 12), method="M
```

```
#check AIC
model3
```

```
##
## Call:
## arima(x = co2training.log, order = c(0, 1, 1), seasonal = list(order = c(0,
##     1, 1), period = 12), method = "ML")
##
## Coefficients:
##           ma1     sma1
##       -0.4669  -0.9026
## s.e.   0.0732   0.0713
##
## sigma^2 estimated as 6.736e-07:  log likelihood = 1212.46,  aic = -2420.92
```

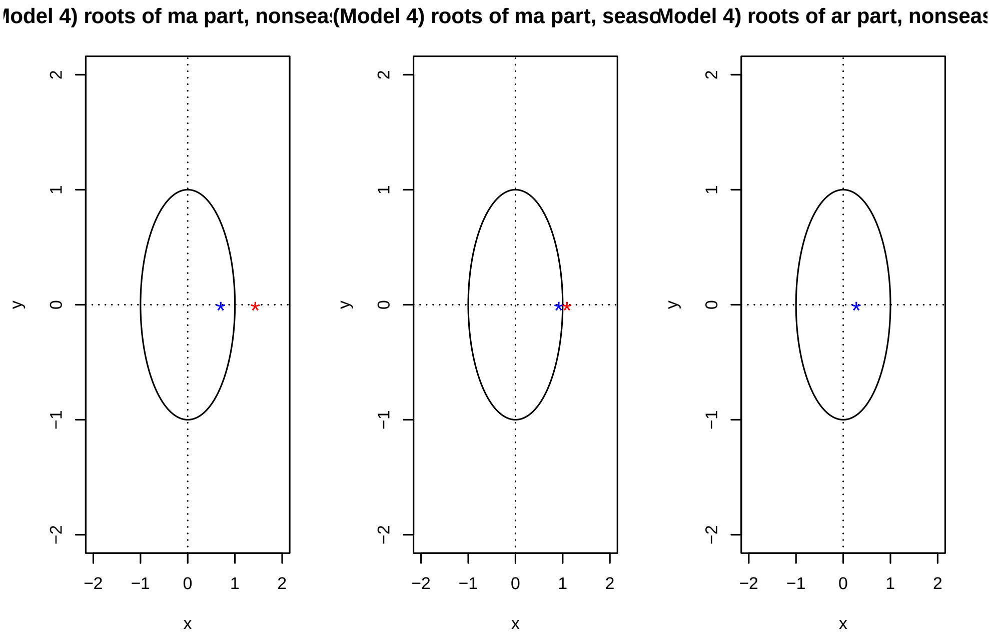
```
model4
```

```
##
## Call:
## arima(x = co2training.log, order = c(1, 1, 1), seasonal = list(order = c(0,
##     1, 1), period = 12), method = "ML")
##
## Coefficients:
##          ar1      ma1     sma1
##       0.2766  -0.6986  -0.9169
## s.e.  0.1377   0.1063   0.0787
##
## sigma^2 estimated as 6.583e-07:  log likelihood = 1214.07,  aic = -2422.15
```

Diagnostic

Model 4

```
par(mfrow=c(1,3))
#To check invertibility of MA part of model 4:
source("plot.roots.R")
plot.roots(NULL,polyroot(c(1, -0.6986)), main="(Model 4) roots of ma part, nonseasonal ")
source("plot.roots.R")
plot.roots(NULL,polyroot(c(1, -0.9169)), main="(Model 4) roots of ma part, seasonal ")
#To check stationarity of AR part of model 4:
source("plot.roots.R")
plot.roots(NULL,polyroot(c(1, -0.2766)), main="(Model 4) roots of ar part, nonseasonal")
```



```
#roots of AR
polyroot(c(1, -0.2766))
```

```
## [1] 3.615329+0i
```

```
#roots of MA
polyroot(c(1, -0.6986))
```

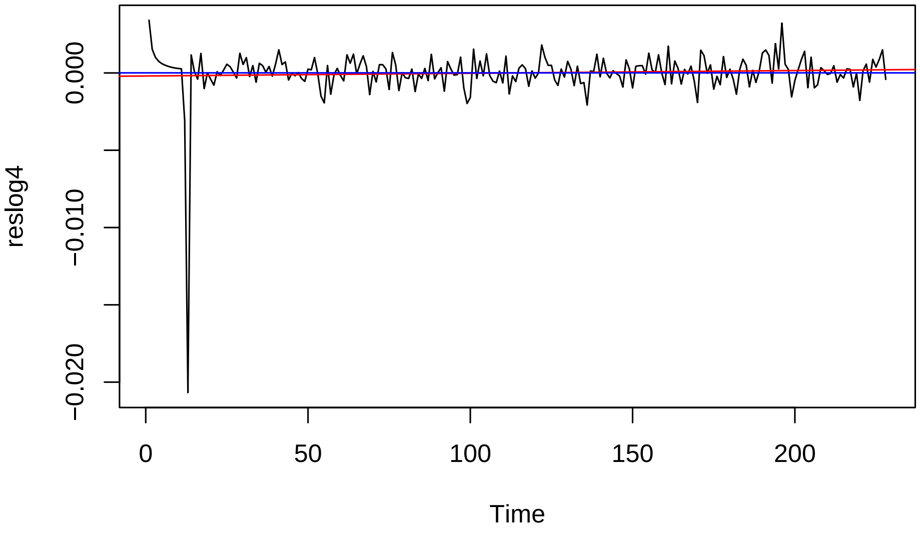
```
## [1] 1.431434+0i
```

```
#roots of seasonal MA
polyroot(c(1, -0.9169))
```

```
## [1] 1.090631+0i
```

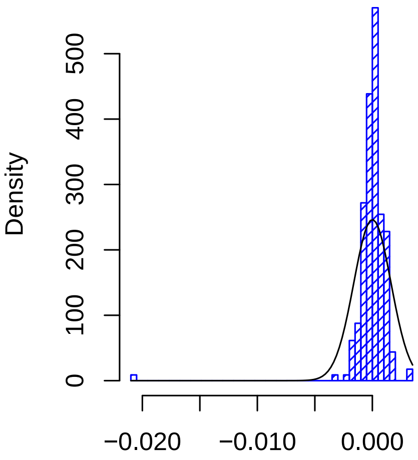
Roots of AR, MA and seasonal MA part lie outside the unit circle. Hence, model 4 is both stationary and invertible. I will follow with series of diagnostics and their interpretation below.

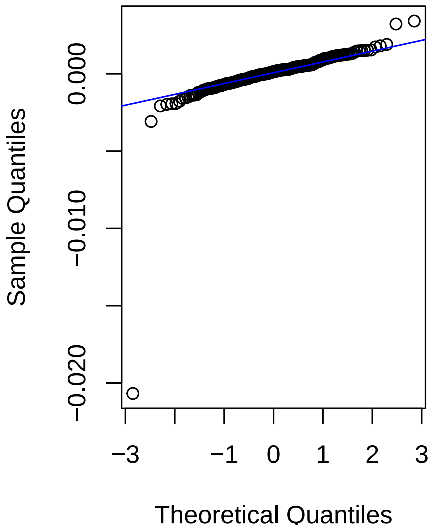
```
reslog4 <- residuals(model4)
#plot of residuals
plot.ts(reslog4)
fitres4 <- lm(reslog4 ~ as.numeric(1:length(reslog4))); abline(fitres4, col="red")
abline(h=mean(reslog4), col="blue")
```



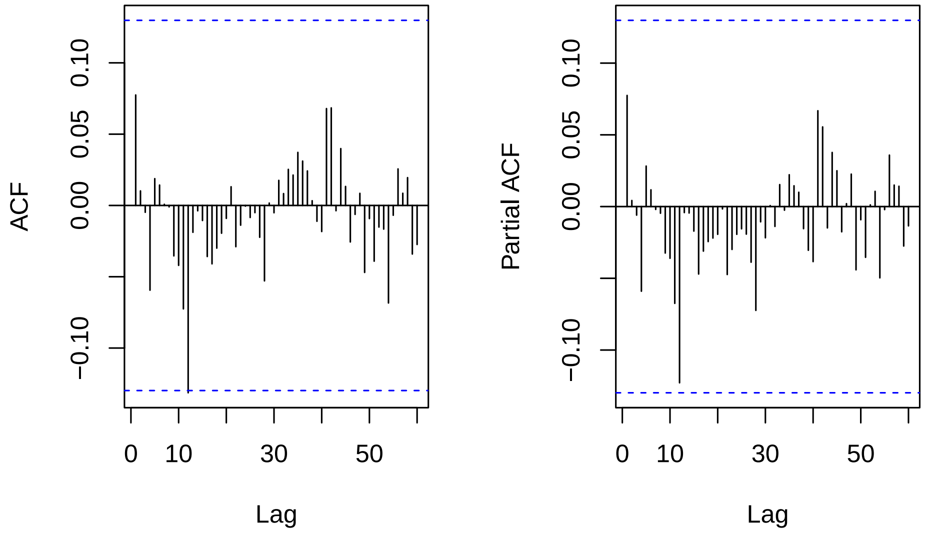
```
par(mfrow=c(1,2))
#histogram of residuals
hist(reslog4,density=20,breaks=40, col="blue", xlab="", prob=TRUE,main="",cex=1)
m4 <- mean(reslog4)
std4 <- sqrt(var(reslog4))
```

```
curve( dnorm(x,m4,std4), add=TRUE )
#qq plot of residuals
qqnorm(reslog4,main= "",cex=1)
qqline(reslog4,col="blue")
```





```
par(mfrow=c(1,2))
#acf, pacf of model 4
acf(reslog4, lag.max=60,
main="",cex=1)
pacf(reslog4, lag.max=60,
main="",cex=1)
```



```
shapiro.test(reslog4)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  reslog4
## W = 0.46196, p-value < 2.2e-16
```

```
Box.test(reslog4, lag = 15, type = c("Box-Pierce"), fitdf = 3)
```

```
##
##  Box-Pierce test
##
## data:  reslog4
## X-squared = 8.2637, df = 12, p-value = 0.7642
```

```
Box.test(reslog4, lag = 15, type = c("Ljung-Box"), fitdf = 3)
```

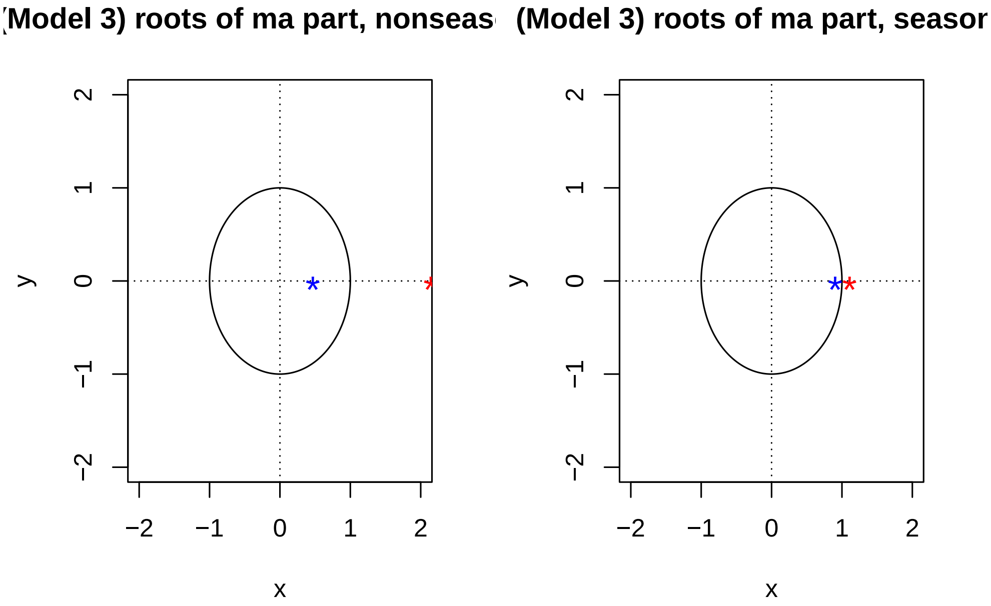
```
##
##  Box-Ljung test
##
## data:  reslog4
## X-squared = 8.6795, df = 12, p-value = 0.73
```

```
Box.test((reslog4)^2, lag = 15, type = c("Ljung-Box"), fitdf = 0)
```

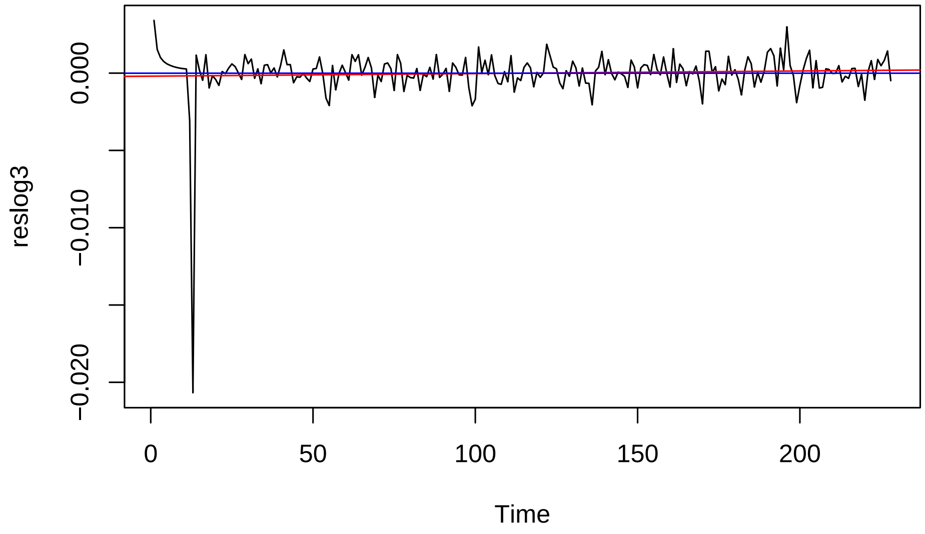
```
##
##  Box-Ljung test
##
## data:  (reslog4)^2
## X-squared = 0.26067, df = 15, p-value = 1
```

Model 3

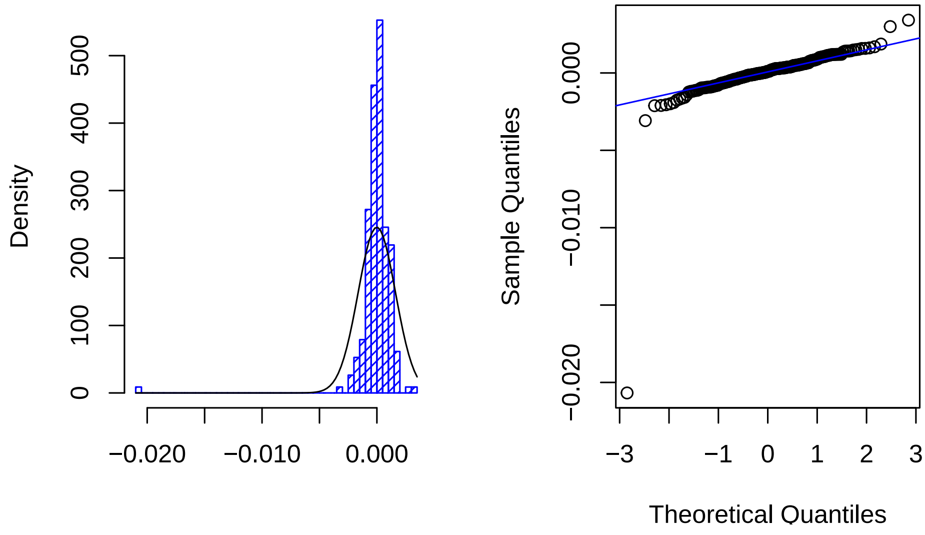
```
par(mfrow=c(1,2))
#To check invertibility of MA part of model 3:
source("plot.roots.R")
plot.roots(NULL,polyroot(c(1, -0.4669)), main="(Model 3) roots of ma part, nonseasonal ")
source("plot.roots.R")
plot.roots(NULL,polyroot(c(1, -0.9026)), main="(Model 3) roots of ma part, seasonal ")
```



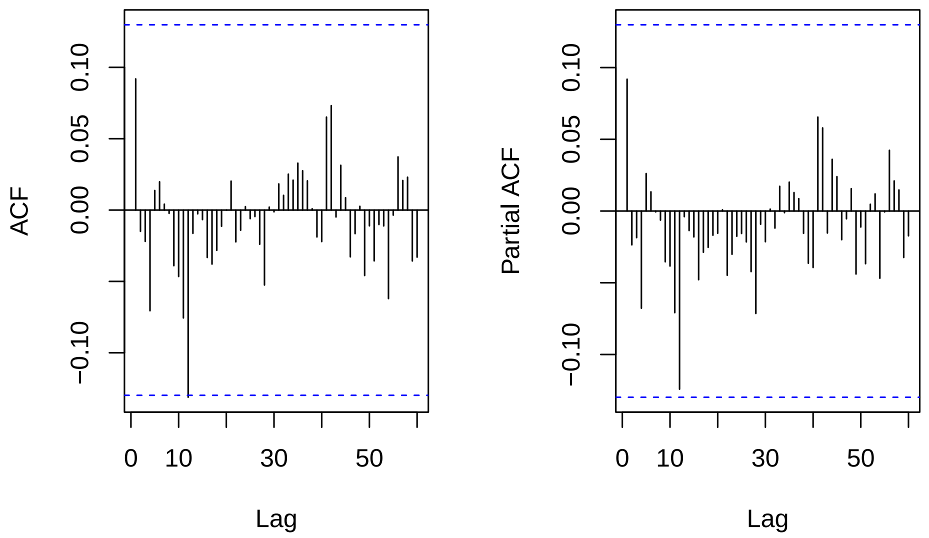
```
reslog3 <- residuals(model3)
#plot of residuals
plot.ts(reslog3)
fitres3 <- lm(reslog3 ~ as.numeric(1:length(reslog3))); abline(fitres3, col="red")
abline(h=mean(reslog3), col="blue")
```



```
par(mfrow=c(1,2))
#histogram of residuals
hist(reslog3,density=20,breaks=40, col="blue", xlab="", prob=TRUE,main="",cex=1)
m3 <- mean(reslog3)
std3 <- sqrt(var(reslog3))
curve( dnorm(x,m3,std3), add=TRUE )
#qq plot of residuals
qqnorm(reslog3,main= "",cex=1)
qqline(reslog3,col="blue")
```



```
par(mfrow=c(1,2))
#acf, pacf of model 3
acf(reslog3, lag.max=60,
main="",cex=1)
pacf(reslog3, lag.max=60,
main="",cex=1)
```



```
shapiro.test(reslog3)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  reslog3
## W = 0.46752, p-value < 2.2e-16
```

```
Box.test(reslog3, lag = 15, type = c("Box-Pierce"), fitdf = 2)
```

```
##
##  Box-Pierce test
##
## data:  reslog3
## X-squared = 9.5007, df = 13, p-value = 0.7342
```

```
Box.test(reslog4, lag = 15, type = c("Ljung-Box"), fitdf = 2)
```

```
##
##  Box-Ljung test
##
## data:  reslog4
## X-squared = 8.6795, df = 13, p-value = 0.7967
```

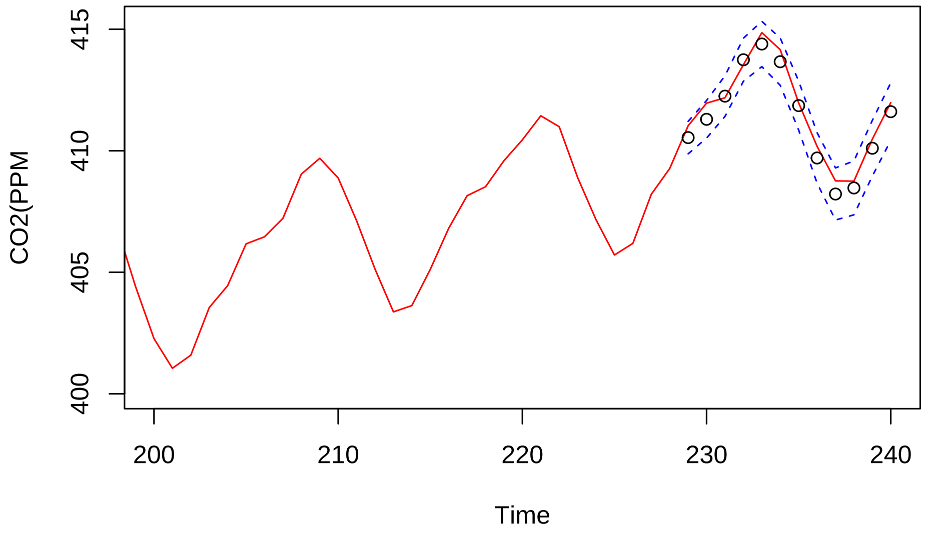
```
Box.test((reslog4)^2, lag = 15, type = c("Ljung-Box"), fitdf = 0)
```

```
##
##  Box-Ljung test
##
## data:  (reslog4)^2
## X-squared = 0.26067, df = 15, p-value = 1
```

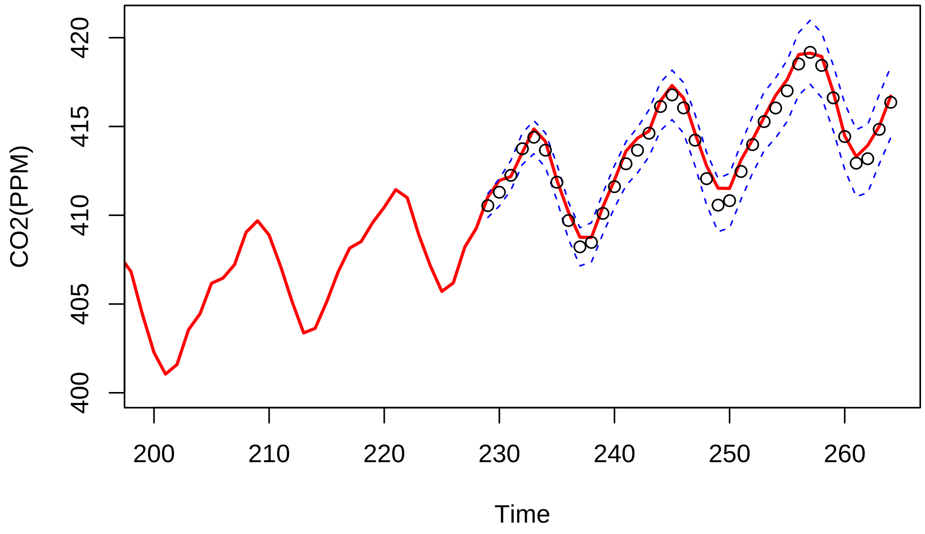
Forecasting

```
co = ts(c[,2])
co2train1 = co[c(1:228)]
co2clean1 =co[c(1:240)]
co2full1 = co[c(1:264)]
```

```
#forecast on log
pred.tr <- predict(model4, n.ahead = 12)
U.tr= pred.tr$pred + 2*pred.tr$se
L.tr= pred.tr$pred - 2*pred.tr$se
#forecast on original
pred.orig=exp(pred.tr$pred)
U=exp(U.tr)
L=exp(L.tr)
#plot-zoomed
ts.plot(co2clean1, xlim = c(200,length(co2train1)+12), ylim = c(400,max(U)), col="red",
main="",
ylab="CO2(PPM")
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(co2train1)+1):(length(co2train1)+12), pred.orig, col="black")
```

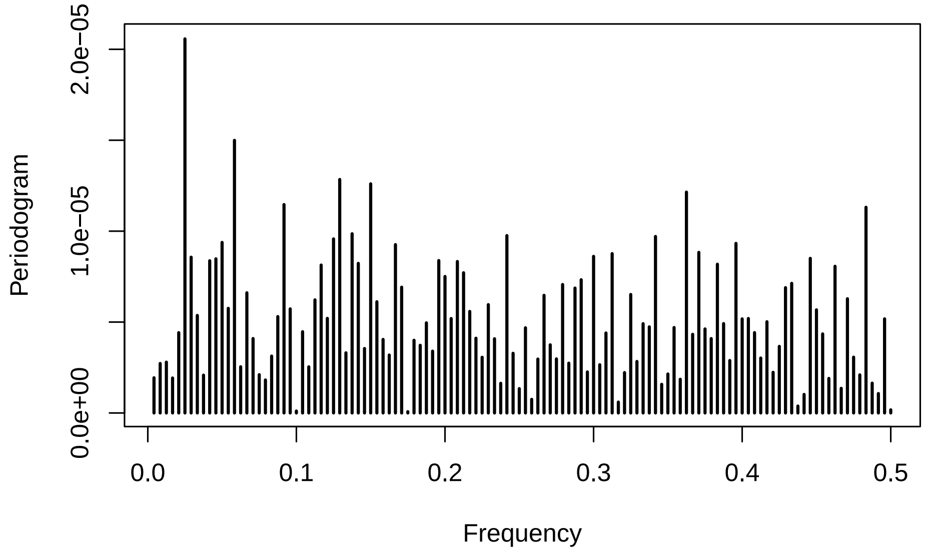


```
pred4 <- predict(model4, n.ahead = 36)
U4= pred4$pred + 2*pred4$se
L4= pred4$pred - 2*pred4$se
pred4.orig=exp(pred4$pred)
U4orig=exp(U4)
L4orig=exp(L4)
ts.plot(co2full1, xlim = c(200,length(co2train1)+36), ylim = c(400,max(U4orig)), col="red",lwd=2,
main="",
ylab="CO2(PPM)")
lines(U4orig, col="blue", lty="dashed")
lines(L4orig, col="blue", lty="dashed")
points((length(co2train1)+1):(length(co2train1)+36), pred4.orig, col="black")
```



Spectral Analysis

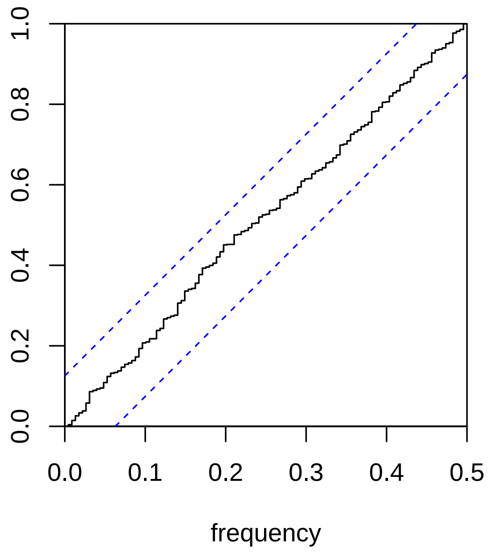
```
library(TSA)
TSA::periodogram(reslog4,main="")
```



```
fisher.g.test(reslog4)
```

```
## [1] 0.9738227
```

```
cpgram(reslog4,main="")
```


frequency