Peiyu Yang-174 final project

Peiyu Yang

2/25/2022

Introduction

The data set that I chose is about CO2 emissions that are produced by coal. This dataset contains the emission of CO2 from coal consumption in the United States from 1973 to 2007 monthly. I chose this dataset because I think CO2 emission is related to global warming, and global warming is related to our daily life. So I want to forecast the emission of CO2 caused by coal consumption for the following years, and hopefully, people can have a brief thoughts about how will the amount of CO2 emissions change in the future.

In this project, I obtained the data from U.S. Energy Information Administration and used RStudio to procedure my analysis and forecasting. I used techniques related to time series, such as plotting my data as time series, doing transformations to make my data stationary with no trend and no seasonality. I also used ACF and PACF graphs to determine the order of my model in order to do further actions. Besides, I also used diagnostic checkings to analyze my residuals to see whether my model is adequate. After ensuring my model fits my data well, I used forecasting skills to forecast future possible data, and then compare my forecasted data with real data to see whether the model I chose is good enough for the dataset.

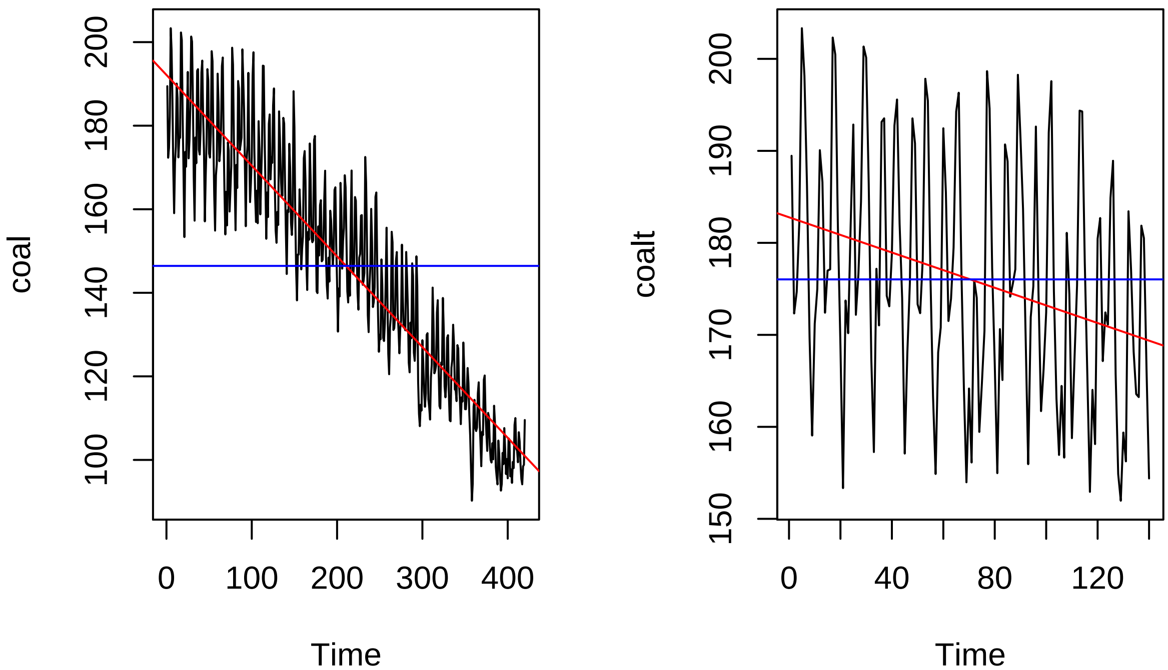
After doing these steps, I successfully obtained an ideal model that can be used to forecast future data points. Although some of the actual data points fit with my predicted points and they are all within the confidence interval of my model, there is still some difference between the actual data and the predicted data. But overall, I think the model that I identified for the data is good enough for forecasting.

Sections

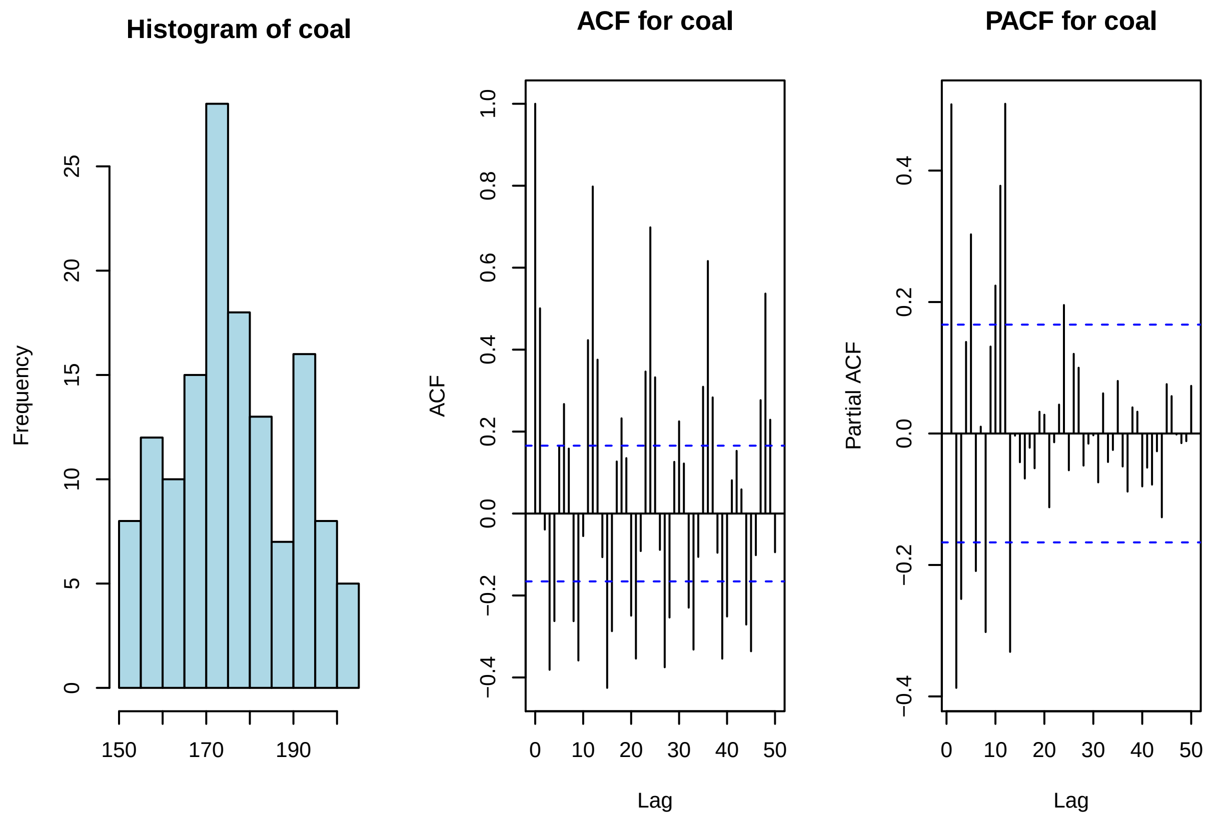
```r
par(mfrow=c(1,2))
coal.csv <- read.table("/Users/peiyu/Desktop/Coal_Including_Coal_Coke_Net_Imports_CO2_Emissions_Monthly
                        sep=',', skip=171)
head(coal.csv)
```

```
##         V1      V2
## 1 Dec 2007 189.467
## 2 Nov 2007 172.325
## 3 Oct 2007 174.684
## 4 Sep 2007 182.172
## 5 Aug 2007 203.334
## 6 Jul 2007 198.149
```

```r
coal <- ts(coal.csv$V2)
ts.plot(coal)
fit <- lm(coal~as.numeric(1:length(coal)))
abline(fit, col='red')
abline(h=mean(coal), col='blue')
coalt=coal[c(1:140)]
coal_test=coal[c(141:152)]
plot.ts(coalt)
fit <- lm(coalt~as.numeric(1:length(coalt)))
abline(fit, col='red')
abline(h=mean(coalt), col='blue')
```

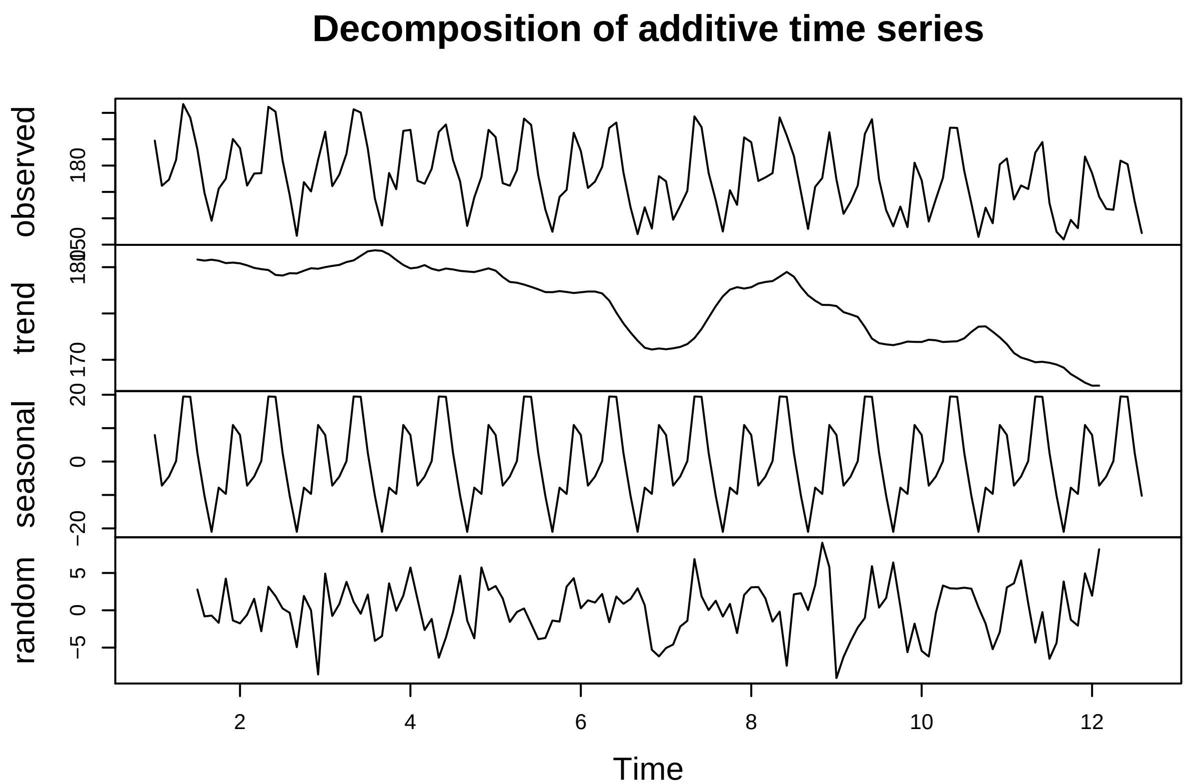



```r
par(mfrow=c(1,3))
hist(coalt, col='light blue', xlab='', main='Histogram of coal')
acf(coalt, lag.max=50, main='ACF for coal')
pacf(coalt, lag.max=50, main='PACF for coal')
```



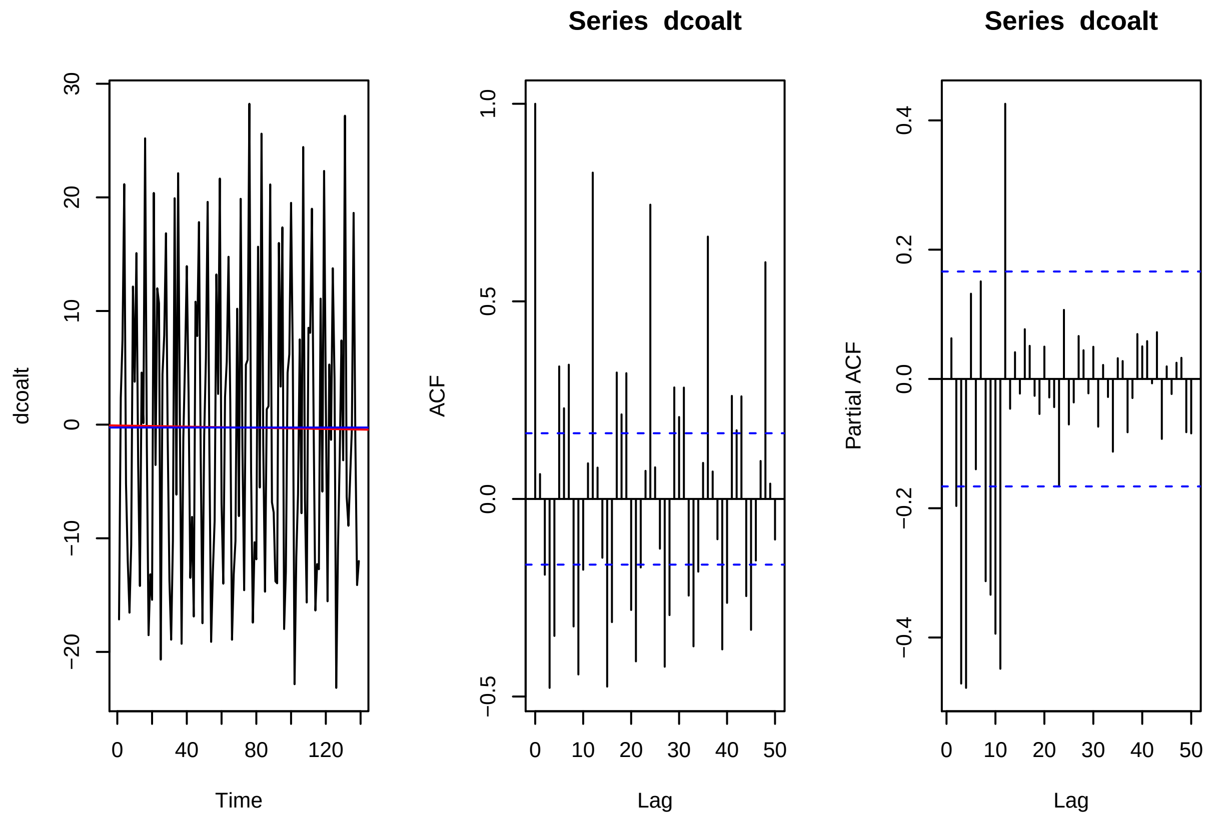
For this dataset, I chose 152 data as the total data I have. I divided the data into two parts. The first 140 data is used as training dataset, and the next 12 data is used as test dataset for future forecasting. From the plots, I think there is no sharp changes, but there is a decreasing trend and seasonal part. Therefore, I decide to decompose the data to see more details.

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

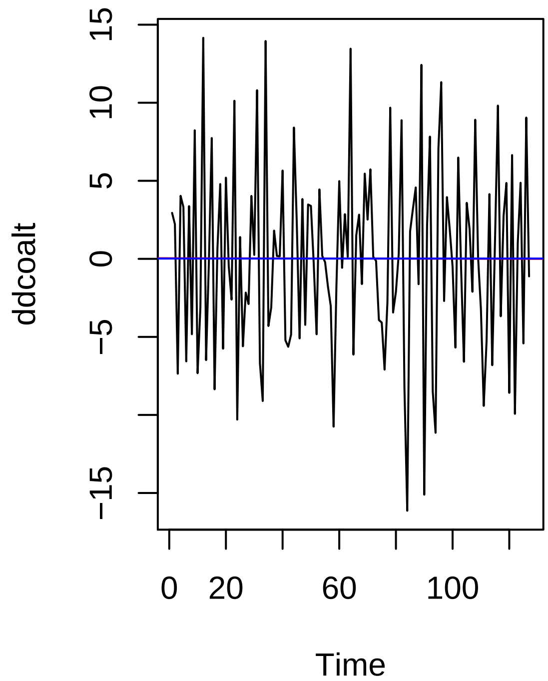


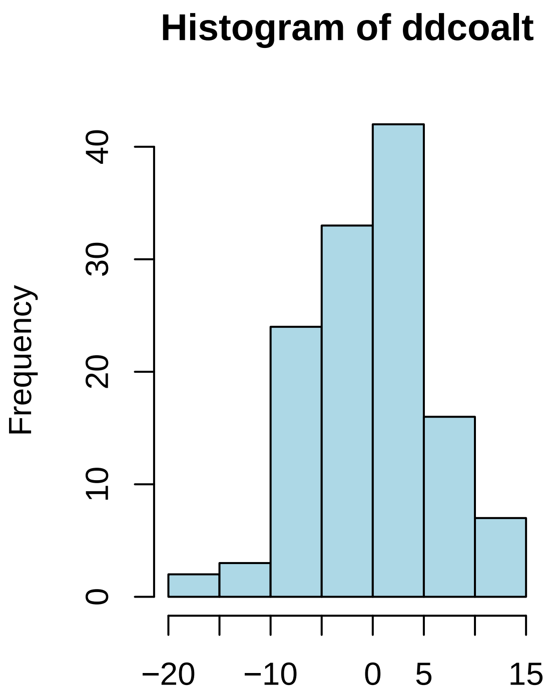
From the decomposition graphs of the data, it show seasonality and decreasing trend of the data. So the next step to do is to difference the data to remove seasonality and the decreasing trend. From the ACF graph, I can tell the seasonality component for this dataset is 12, so I decide to difference the data at lag 1 once to remove the trend, and difference it again at lag 12 to remove the seasonality.

```
par(mfrow=c(1,3))
dcoalt <- diff(coalt,1)
ts.plot(dcoalt)
fit <- lm(dcoalt~as.numeric(1:length(dcoalt)))
abline(fit, col='red')
abline(h=mean(dcoalt), col='blue')
acf(dcoalt, lag.max=50)
pacf(dcoalt, lag.max=50)
```

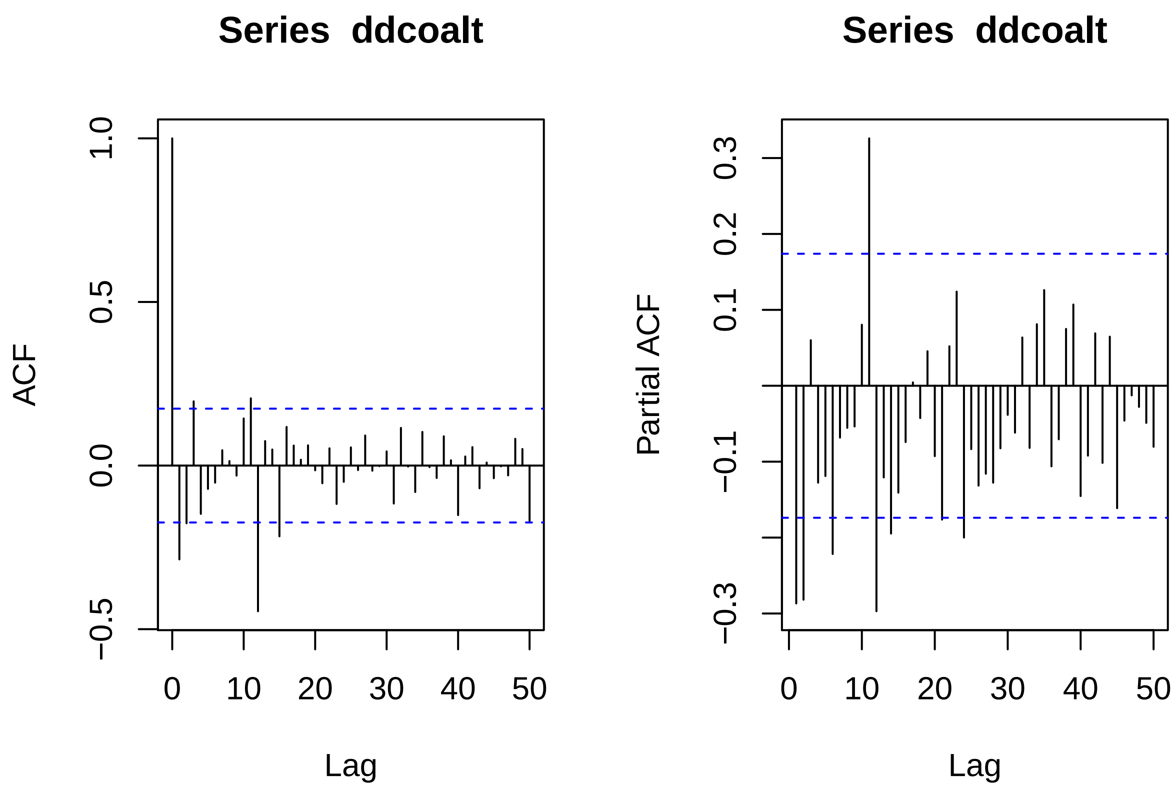



```
ddcoalt <- diff(dcoalt,lag=12, differences=1)
par(mfrow=c(1,2))
ts.plot(ddcoalt)
fit <- lm(ddcoalt~as.numeric(1:length(ddcoalt)))
abline(fit, col='red')
abline(h=mean(ddcoalt), col='blue')
hist(ddcoalt, col='light blue', xlab='', main='Histogram of ddcoalt')
```






```
acf(ddcoalt, lag.max=50)
pacf(ddcoalt, lag.max=50)
```




```r
var(coalt)
```

```
## [1] 172.2347
```

```r
var(dcoalt)
```

```
## [1] 168.4336
```

```r
var(ddcoalt)
```

```
## [1] 37.21625
```

After differencing the data at lag 1 once and at lag 12 once, there is no trend and no seasonality anymore, and the variance is lower than the original data. From the ACF and PACF graphs, they didn't show any indication of existence of trend and seasonality. And the histogram of ddcoalt looks more symmetric than the original data. Therefore, I think the data is stationary now, and good enough for me to try fit the model.

Trying models

From the ACF and PACF graphs, they indicate me to choose s=12, d=1, D=1, p=2, q=1 or 3, P=1, Q=1.

```r
library(qpcR)
```

```
## Loading required package: MASS

## Loading required package: minpack.lm

## Loading required package: rgl

## Loading required package: robustbase

## Loading required package: Matrix
```

```r
df <- expand.grid(p=0:2, q=0:3, P=0:1, Q=0:1)
df <- cbind(df, AICc=NA)
for (i in 1:nrow(df)) {
sarima.obj <- NULL
try(arima.obj <- arima(coalt, order=c(df$p[i], 1, df$q[i]),
seasonal=list(order=c(df$P[i], 1, df$Q[i]), period=12),
method="ML"))
if (!is.null(arima.obj)) { df$AICc[i] <- AICc(arima.obj) }
# print(df[i, ])
}
```

```
## Warning in log(s2): NaNs produced

## Warning in log(s2): NaNs produced
```

```r
df[which.min(df$AICc), ]
```

```
##    p q P Q     AICc
## 29 1 1 0 1 741.5262
```

```r
sort(df$AICc, decreasing=F)
```

```
##  [1] 741.5262 743.4787 744.1240 744.2410 744.6651 744.7767 746.0611 746.2057
##  [9] 746.3113 746.4116 746.7934 747.7160 748.1524 748.2600 748.3366 749.7421
## [17] 749.7619 750.3017 750.4338 751.8727 756.1011 757.9494 765.5004 766.9988
## [25] 767.0770 768.3984 768.5222 768.6674 772.2639 774.2903 774.7459 776.6547
## [33] 777.7042 777.9123 783.5142 789.9211 792.2259 793.4292 794.8257 798.9160
## [41] 800.3452 801.6692 803.3908 803.6487 804.2131 804.6794 811.9240 820.7343
```

```r
df
```

```
##    p q P Q     AICc
## 1  0 0 0 0 820.7343
## 2  1 0 0 0 811.9240
## 3  2 0 0 0 803.3908
## 4  0 1 0 0 804.6794
## 5  1 1 0 0 794.8257
```

```
## 6   2 1 0 0 804.2131
## 7   0 2 0 0 801.6692
## 8   1 2 0 0 798.9160
## 9   2 2 0 0 793.4292
## 10  0 3 0 0 803.6487
## 11  1 3 0 0 800.3452
## 12  2 3 0 0 792.2259
## 13  0 0 1 0 789.9211
## 14  1 0 1 0 783.5142
## 15  2 0 1 0 776.6547
## 16  0 1 1 0 777.7042
## 17  1 1 1 0 767.0770
## 18  2 1 1 0 777.9123
## 19  0 2 1 0 772.2639
## 20  1 2 1 0 768.6674
## 21  2 2 1 0 768.5222
## 22  0 3 1 0 774.2903
## 23  1 3 1 0 774.7459
## 24  2 3 1 0 768.3984
## 25  0 0 0 1 765.5004
## 26  1 0 0 1 756.1011
## 27  2 0 0 1 748.3366
## 28  0 1 0 1 748.2600
## 29  1 1 0 1 741.5262
## 30  2 1 0 1 749.7421
## 31  0 2 0 1 744.1240
## 32  1 2 0 1 746.3113
## 33  2 2 0 1 744.7767
## 34  0 3 0 1 746.2057
## 35  1 3 0 1 747.7160
## 36  2 3 0 1 744.2410
## 37  0 0 1 1 766.9988
## 38  1 0 1 1 757.9494
## 39  2 0 1 1 750.4338
## 40  0 1 1 1 750.3017
## 41  1 1 1 1 743.4787
## 42  2 1 1 1 751.8727
## 43  0 2 1 1 746.0611
## 44  1 2 1 1 744.6651
## 45  2 2 1 1 746.7934
## 46  0 3 1 1 748.1524
## 47  1 3 1 1 749.7619
## 48  2 3 1 1 746.4116
```

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

```
## Warning in log(s2): NaNs produced
```

```
##
## Call:
## arima(x = coalt, order = c(1, 1, 1), seasonal = list(order = c(0, 1, 1), period = 12),
##     method = "ML")
##
```

```
## Coefficients:
##          ar1      ma1     sma1
##       0.5147  -0.9657  -0.8246
## s.e.  0.0995   0.0737   0.1003
##
## sigma^2 estimated as 16.53:  log likelihood = -366.67,  aic = 741.35
```

```r
AICc(arima(coalt, order=c(1,1,1), seasonal=list(order=c(0,1,1), period=12), method="ML"))
```

```
## Warning in log(s2): NaNs produced
```

```
## [1] 741.5262
```

```r
# Model A
arima(coalt, order=c(1,1,1), seasonal=list(order=c(1,1,1), period=12), method="ML")
```

```
##
## Call:
## arima(x = coalt, order = c(1, 1, 1), seasonal = list(order = c(1, 1, 1), period = 12),
##     method = "ML")
##
## Coefficients:
##          ar1      ma1     sar1     sma1
##       0.5136  -0.9610  -0.0490  -0.7955
## s.e.  0.1026   0.0743   0.1186   0.1169
##
## sigma^2 estimated as 16.62:  log likelihood = -366.59,  aic = 743.18
```

```r
AICc(arima(coalt, order=c(1,1,1), seasonal=list(order=c(1,1,1), period=12), method="ML"))
```

```
## [1] 743.4787
```

```r
# Model B
arima(coalt, order=c(0,1,2), seasonal=list(order=c(0,1,1), period=12), method="ML")
```

```
##
## Call:
## arima(x = coalt, order = c(0, 1, 2), seasonal = list(order = c(0, 1, 1), period = 12),
##     method = "ML")
##
## Coefficients:
##           ma1      ma2     sma1
##       -0.4381  -0.2940  -0.8568
## s.e.   0.0924   0.1114   0.1124
##
## sigma^2 estimated as 16.92:  log likelihood = -367.97,  aic = 743.95
```

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

```
## [1] 744.124
```

For the model choosing, I ran a for loop to estimate which model produce the lowest AICc value. I chose models that have the lowest to estimate the coefficients. Although this model has the lowest AICc value, it produces NaNs value. Therefore, I decide to model the coefficients with the second lowest AICc value and the third lowest AICc value.

$(A) : (1 - 0.5136B)(1 + 0.049B^{12})(1 - B)(1 - B^{12})X_t = (1 - 0.961B)(1 - 0.7955B^{12})Z_t$, $\sigma_Z^2 = 16.62$

$(B) : (1 - B)(1 - B^{12})X_t = (1 - 0.4381B - 0.294B^2)(1 - 0.8568B^{12})Z_t$, $\sigma_Z^2 = 16.92$

Check invertible and stationary

```
library(UnitCircle)
# For model A
par(mfrow=c(1,4))
uc.check(pol_=c(1,-0.5136), plot_output=T)
```

```
##      real complex outside
## 1 1.94704       0    TRUE
## *Results are rounded to 6 digits.
```

```
uc.check(pol_=c(1,0.049), plot_output=T)
```

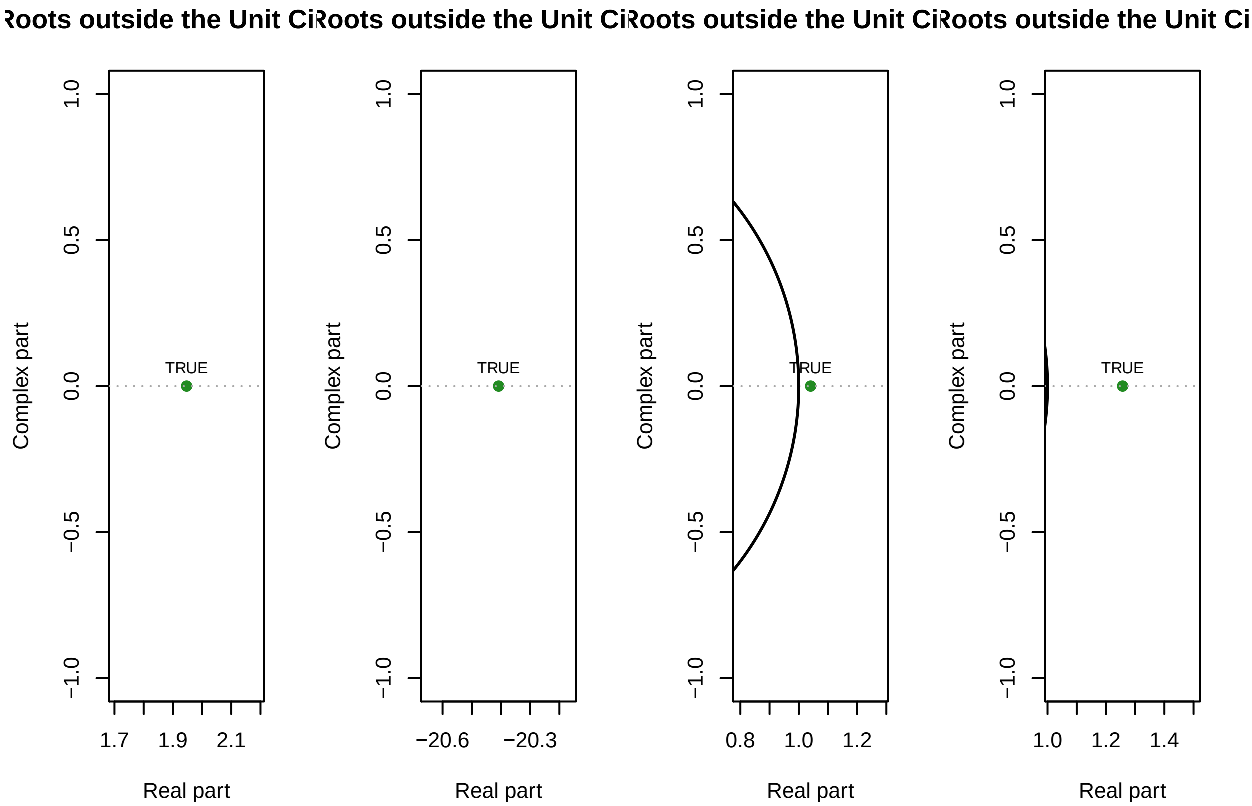
```
##        real complex outside
## 1 -20.40816       0    TRUE
## *Results are rounded to 6 digits.
```

```
uc.check(pol_=c(1,-0.961), plot_output=T)
```

```
##       real complex outside
## 1 1.040583       0    TRUE
## *Results are rounded to 6 digits.
```

```
uc.check(pol_=c(1,-0.7955), plot_output=T)
```

```
##       real complex outside
## 1 1.257071       0    TRUE
## *Results are rounded to 6 digits.
```

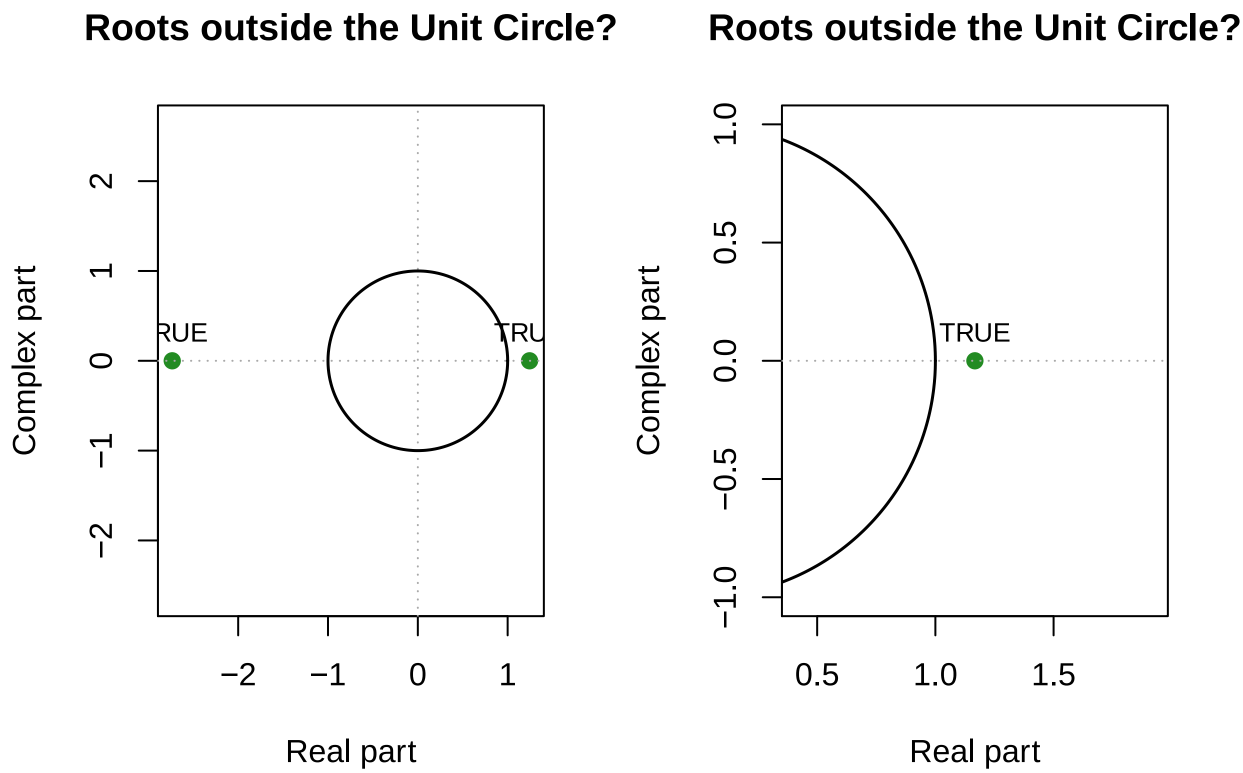



```r
# For model B
par(mfrow=c(1,2))
uc.check(pol_=c(1,-0.4381,-0.294), plot_output=T)
```

```
##        real complex outside
## 1  1.244024       0    TRUE
## 2 -2.734160       0    TRUE
## *Results are rounded to 6 digits.
```

```r
uc.check(pol_=c(1,-0.8568), plot_output=T)
```

```
##       real complex outside
## 1 1.167134       0    TRUE
## *Results are rounded to 6 digits.
```

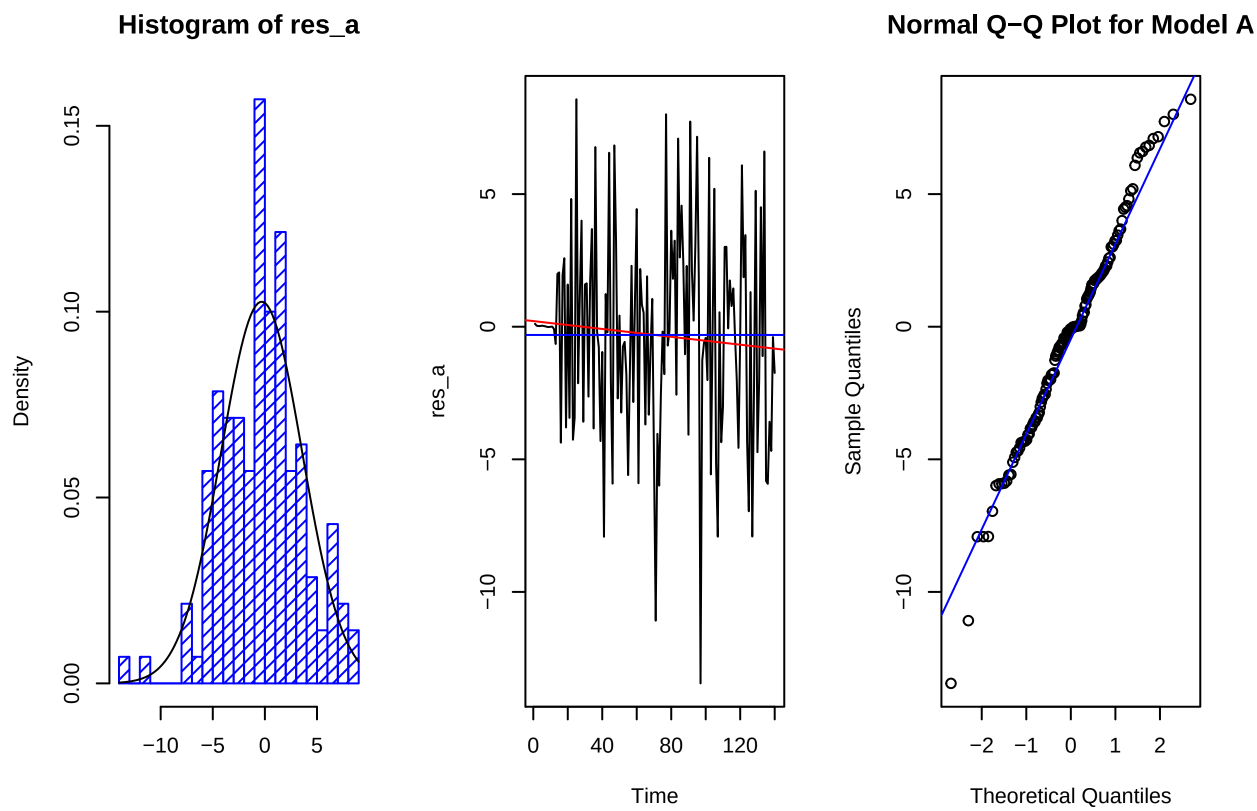


Model B is stationary since it is pure MA.

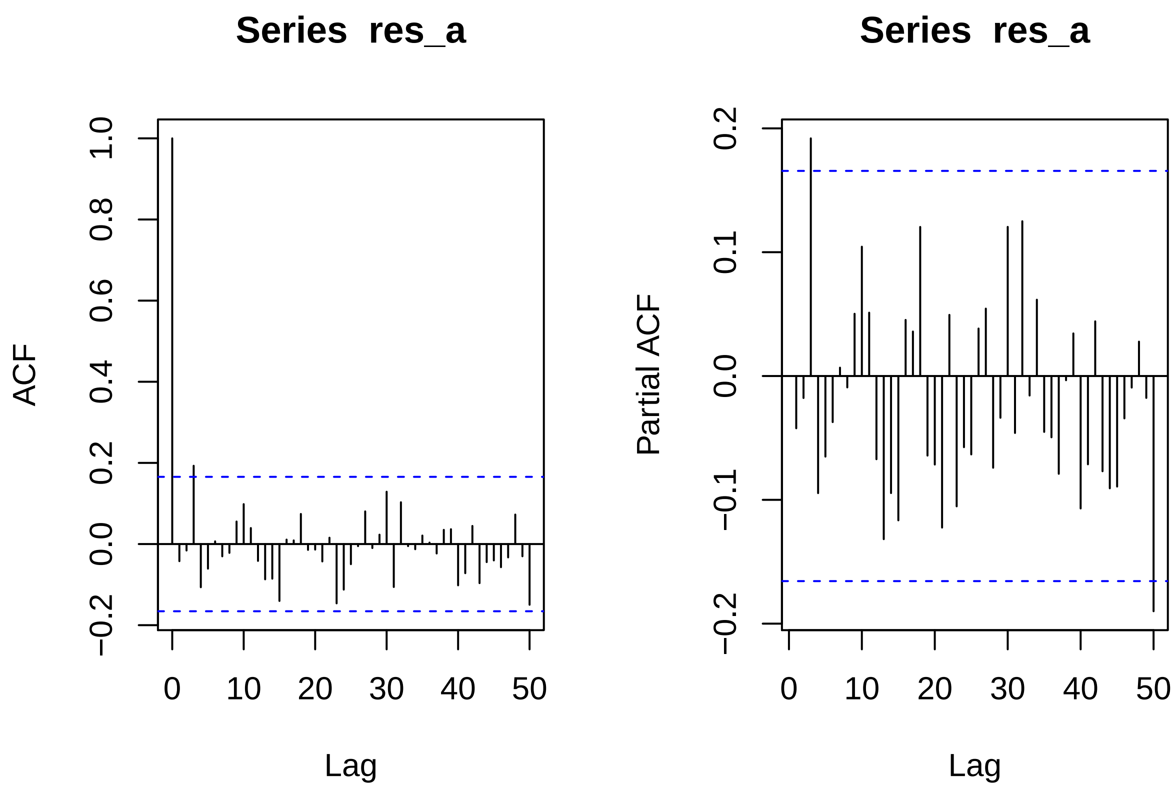
Both model A and model B are invertible and stationary.

Diagnostic checking for Model A

```
par(mfrow=c(1,3))
fit_a <- arima(coalt, order=c(1,1,1), seasonal=list(order=c(1,1,1), period=12), method="ML")
res_a <- residuals(fit_a)
hist(res_a,density=20, breaks=20, col='blue', xlab='', prob=TRUE)
m_a <- mean(res_a)
std_a <- sqrt(var(res_a))
curve(dnorm(x,m_a,std_a), add=T)
plot.ts(res_a)
fitt_a <- lm(res_a~as.numeric(1:length(res_a)))
abline(fitt_a, col='red')
abline(h=mean(res_a), col='blue')
qqnorm(res_a, main="Normal Q-Q Plot for Model A")
qqline(res_a, col='blue')
```




```r
par(mfrow=c(1,2))
acf(res_a, lag.max=50)
pacf(res_a, lag.max=50)
```



From the graphs, we can tell there is no trend, no visible change of variance, and no seasonality. The sample mean is almost zero, and histogram and Q-Q plot look good. From the ACF and PACF grapgs, we can tell all ACF of residuals are within confidence intervals and can be counted as zeros. But for the PACF graph of residuals, there are some lags that are outside of the confidence interval.

```
shapiro.test(res_a)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res_a
## W = 0.98734, p-value = 0.229
```

```
Box.test(res_a, lag=12, type=c("Box-Pierce"), fitdf=4)
```

```
##
##  Box-Pierce test
##
## data:  res_a
## X-squared = 10.049, df = 8, p-value = 0.2616
```

```
Box.test(res_a, lag=12, type=c("Ljung-Box"), fitdf=4)
```

```
##
```

```
##  Box-Ljung test
##
## data:  res_a
## X-squared = 10.567, df = 8, p-value = 0.2275
```

```
Box.test((res_a)^2, lag=12, type=c("Ljung-Box"), fitdf=0)
```

```
##
##  Box-Ljung test
##
## data:  (res_a)^2
## X-squared = 10.81, df = 12, p-value = 0.5452
```

For these four tests, all p-value is greater than 0.05.

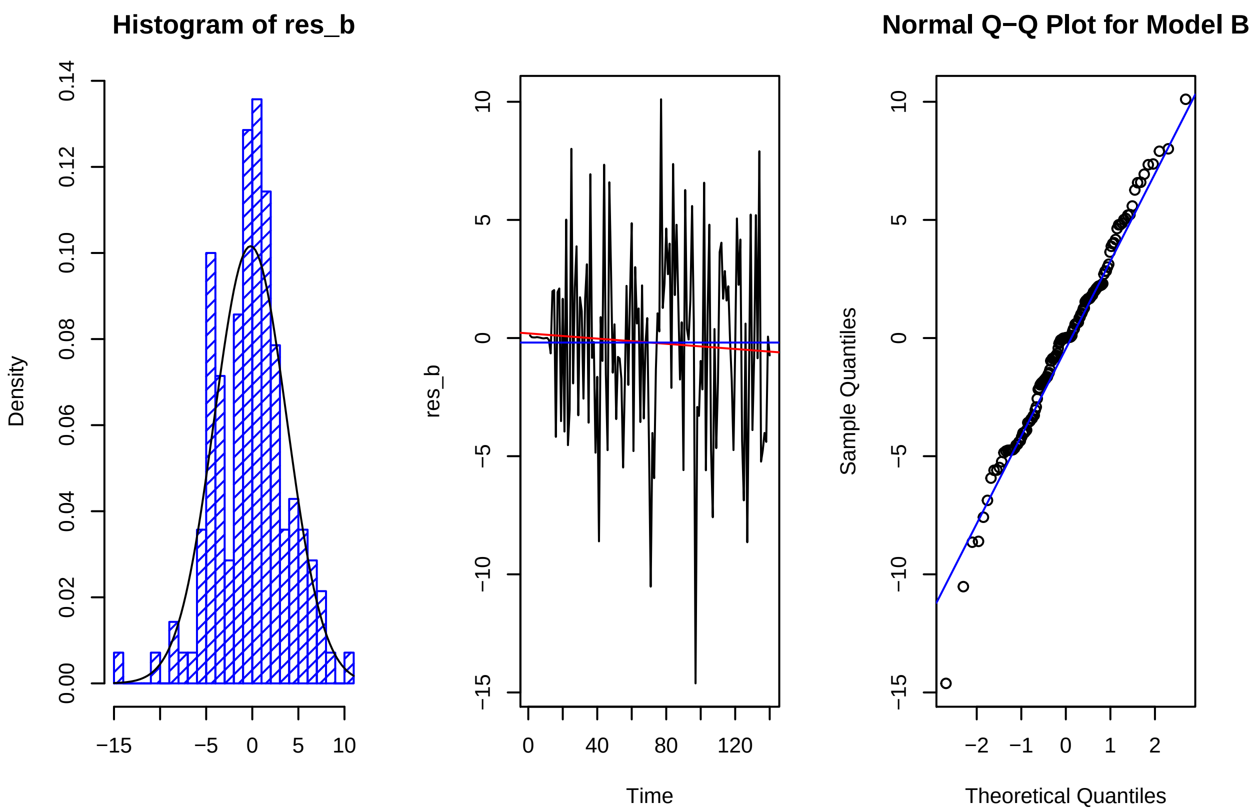
```
ar(res_a, aic=TRUE, order.max=NULL, method=c("yule-walker"))
```

```
##
## Call:
## ar(x = res_a, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  15.09
```

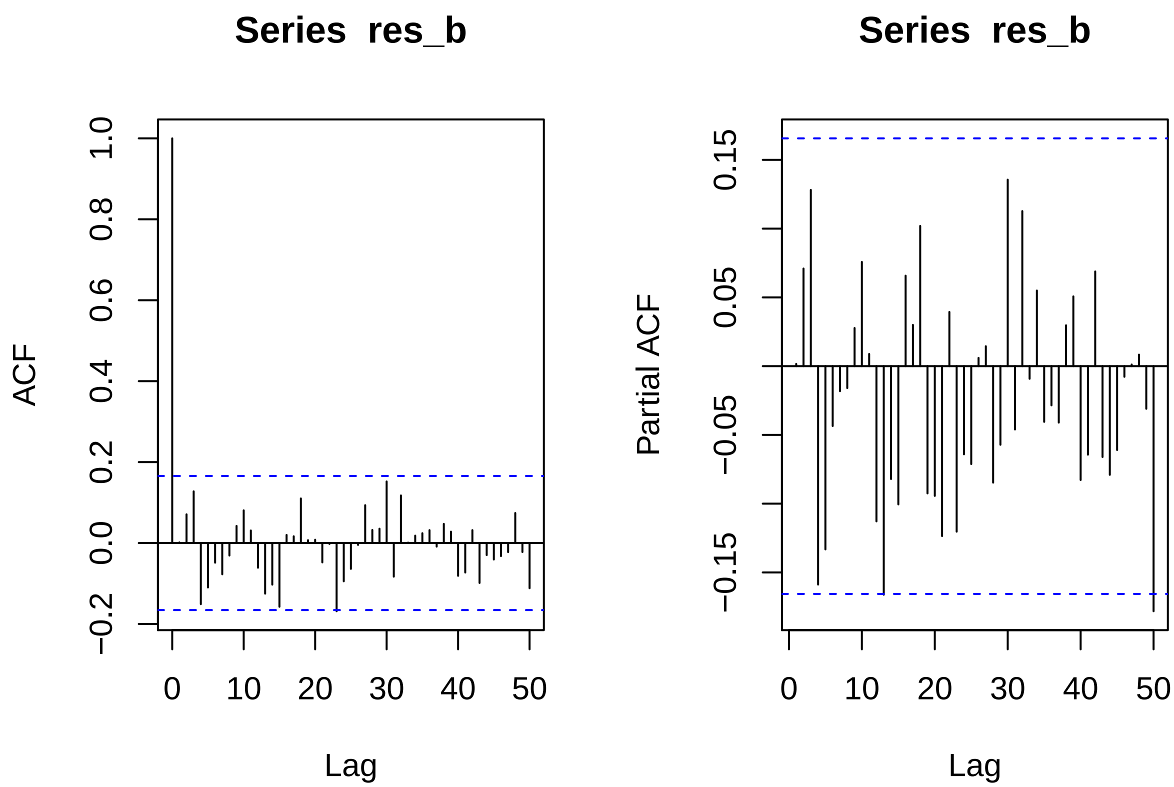
Fitted residuals to AR(0), which is WN.

Diagnostic checking for model B

```
par(mfrow=c(1,3))
fit_b <- arima(coalt, order=c(0,1,2), seasonal=list(order=c(0,1,1), period=12), method="ML")
res_b <- residuals(fit_b)
hist(res_b,density=20, breaks=20, col='blue', xlab='', prob=TRUE)
m_b <- mean(res_b)
std_b <- sqrt(var(res_b))
curve(dnorm(x,m_b,std_b), add=T)
plot.ts(res_b)
fitt_b <- lm(res_b~as.numeric(1:length(res_b)))
abline(fitt_b, col='red')
abline(h=mean(res_b), col='blue')
qqnorm(res_b, main="Normal Q-Q Plot for Model B")
qqline(res_b, col='blue')
```




```
par(mfrow=c(1,2))
acf(res_b, lag.max=50)
pacf(res_b, lag.max=50)
```



From the graphs, we can tell there is no trend, no visible change of variance, and no seasonality. The sample mean is almost zero, and histogram and Q-Q plot look good. All ACF and PACF of residuals are within confidence intervals and can be counted as zeros.

```
shapiro.test(res_b)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res_b
## W = 0.98609, p-value = 0.1699
```

```
Box.test(res_b, lag=12, type=c("Box-Pierce"), fitdf=3)
```

```
##
##  Box-Pierce test
##
## data:  res_b
## X-squared = 11.003, df = 9, p-value = 0.2755
```

```
Box.test(res_b, lag=12, type=c("Ljung-Box"), fitdf=3)
```

```
##
##  Box-Ljung test
```

```
##
## data:  res_b
## X-squared = 11.598, df = 9, p-value = 0.2369
```

```
Box.test((res_b)^2, lag=12, type=c("Ljung-Box"), fitdf=0)
```

```
##
##  Box-Ljung test
##
## data:  (res_b)^2
## X-squared = 9.2664, df = 12, p-value = 0.68
```

For these four tests, all p-value is greater than 0.05.

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

```
##
## Call:
## ar(x = res_b, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  15.42
```

Fitted residuals to AR(0), which is WN. It passes all diagnostic checking, so it is ready to be used for forecasting.

For model A, the PACF graph indicates there are some lags outside of the confidence interval. For model B, the ACF and PACF look better than model A. Also, model A estimates 4 coefficients and model B estimates 3 coefficients. Because of the principle of parsimony, it also suggests me to choose model B. Therefore I will choose model B to do further forecasting.

Therefore, the final model that can be used for forecasting is coalt follows SARIMA$(0,1,2)(0,1,1)_{12}$

$$(1-B)(1-B^{12})X_t = (1-0.4381B-0.294B^2)(1-0.8568B^{12})Z_t,\ \sigma_Z^2 = 16.92$$

Forecasting

```
# install.packages("forecast")
library(forecast)
```

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

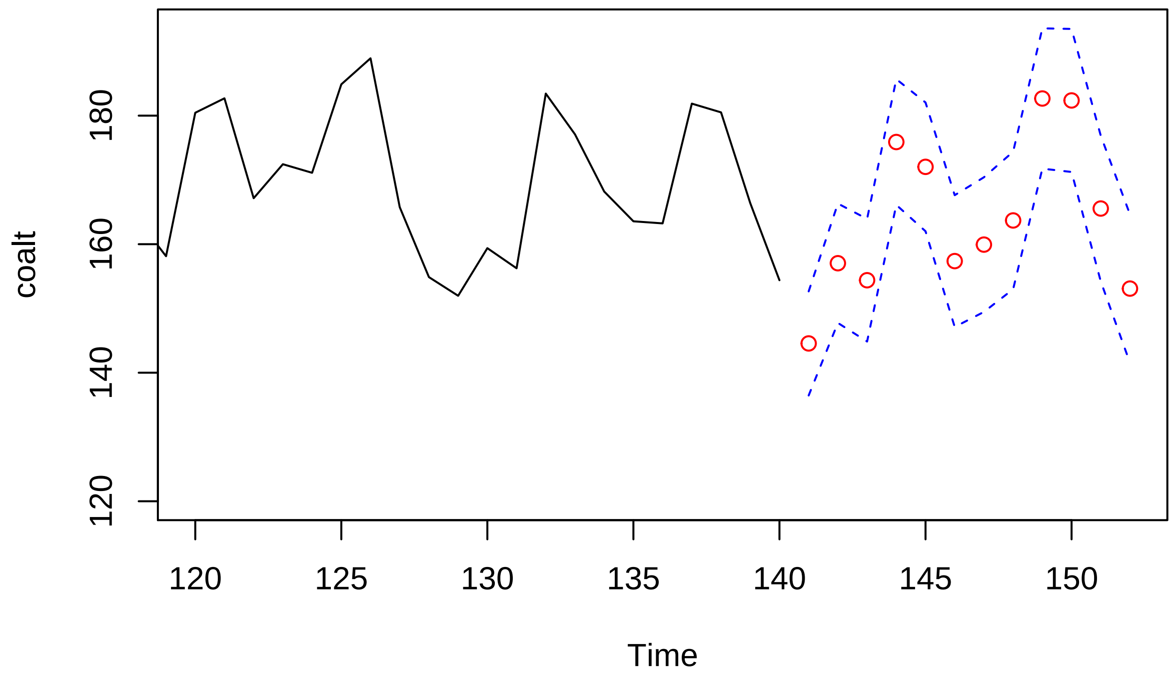
```
## Registered S3 methods overwritten by 'forecast':
##   method                 from
##   autoplot.Arima         ggfortify
##   autoplot.acf           ggfortify
##   autoplot.ar            ggfortify
##   autoplot.bats          ggfortify
##   autoplot.decomposed.ts ggfortify
```

```
##   autoplot.ets          ggfortify
##   autoplot.forecast     ggfortify
##   autoplot.stl          ggfortify
##   autoplot.ts           ggfortify
##   fitted.ar             ggfortify
##   fortify.ts            ggfortify
##   residuals.ar          ggfortify
```

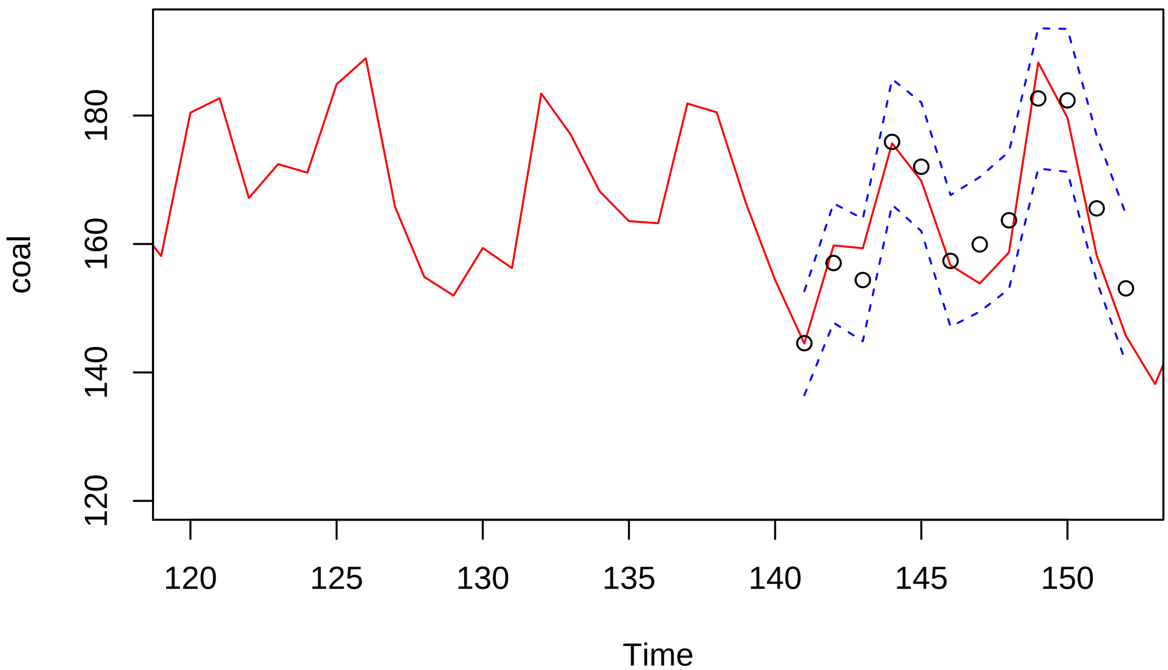
```r
forecast(fit_b)
```

```
##     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 141       144.5551 139.2599 149.8502 136.4568 152.6533
## 142       157.0346 150.9607 163.1085 147.7454 166.3238
## 143       154.3890 148.1517 160.6264 144.8498 163.9282
## 144       175.8928 169.4963 182.2894 166.1102 185.6755
## 145       172.0301 165.4800 178.5802 162.0125 182.0476
## 146       157.3693 150.6677 164.0709 147.1201 167.6185
## 147       159.9332 153.0835 166.7829 149.4575 170.4090
## 148       163.6913 156.6966 170.6860 152.9938 174.3888
## 149       182.6669 175.5302 189.8037 171.7522 193.5817
## 150       182.3634 175.0874 189.6395 171.2357 193.4912
## 151       165.5449 158.1322 172.9576 154.2082 176.8816
## 152       153.0877 145.5409 160.6346 141.5458 164.6297
## 153       142.9957 135.1298 150.8616 130.9658 155.0255
## 154       155.2909 147.2066 163.3753 142.9270 167.6549
## 155       152.6454 144.3981 160.8926 140.0322 165.2585
## 156       174.1492 165.7415 182.5568 161.2907 187.0076
## 157       170.2864 161.7216 178.8512 157.1877 183.3851
## 158       155.6256 146.9077 164.3435 142.2928 168.9584
## 159       158.1895 149.3212 167.0579 144.6266 171.7525
## 160       161.9476 152.9314 170.9639 148.1585 175.7368
## 161       180.9233 171.7615 190.0851 166.9115 194.9350
## 162       180.6198 171.3147 189.9249 166.3889 194.8507
## 163       163.8012 154.3550 173.2474 149.3545 178.2479
## 164       151.3441 141.7589 160.9293 136.6848 166.0034
```

```r
pred.tr <- predict(fit_b, 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(coalt, xlim=c(120, length(coalt)+12), ylim=c(120, max(U.tr)))
lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")
points((length(coalt)+1):(length(coalt)+12),pred.tr$pred, col="red")
```




```
ts.plot(coal, xlim = c(120,length(coalt)+12), ylim = c(120,max(U.tr)), col="red")
lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")
points((length(coalt)+1):(length(coalt)+12), pred.tr$pred, col="black")
```



The red line represents the original data, and the black circles represents the forecasted data. The test set is within prediction intervals.

Conclusion

For this project, the goal I intended to achieve is by using RStudio and skills about time series with previous data to forecast future values. After applying skills to make my selected data stationary and selecting models and coefficients, I obtained an ideal model that can be used to forecast future data points. Although some of the actual data points fit with my predicted points and they are all within the confidence interval of my model, there is still some difference between the actual data and the predicted data. Despite the difference between the actual data and predicted data, I still think I have achieved my goal of forecasting since the majority trend of data points matches. For this project, I got help from professor Feldman and TA Youhong Lee. The final model I chose for my data is SARIMA$(0,1,2)(0,1,1)_{12}$

$$(1-B)(1-B^{12})X_t = (1-0.4381B-0.294B^2)(1-0.8568B^{12})Z_t,\ \sigma_Z^2 = 16.92$$

Reference

Lecture notes, U.S. Energy Information Administration.

Appendix

```
knitr::opts_chunk$set(echo = TRUE)
par(mfrow=c(1,2))
coal.csv <- read.table("/Users/peiyu/Desktop/Coal_Including_Coal_Coke_Net_Imports_CO2_Emissions_Monthly
                         sep=',', skip=171)
head(coal.csv)
coal <- ts(coal.csv$V2)
ts.plot(coal)
fit <- lm(coal~as.numeric(1:length(coal)))
abline(fit, col='red')
abline(h=mean(coal), col='blue')
coalt=coal[c(1:140)]
coal_test=coal[c(141:152)]
plot.ts(coalt)
fit <- lm(coalt~as.numeric(1:length(coalt)))
abline(fit, col='red')
abline(h=mean(coalt), col='blue')
par(mfrow=c(1,3))
hist(coalt, col='light blue', xlab='', main='Histogram of coal')
acf(coalt, lag.max=50, main='ACF for coal')
pacf(coalt, lag.max=50, main='PACF for coal')
library(ggplot2)
library(ggfortify)
#install.packages("ggplot2")
#install.packages("ggfortify")
y <- ts(as.ts(coalt), frequency=12)
decomp <- decompose(y)
plot(decomp)
par(mfrow=c(1,3))
dcoalt <- diff(coalt,1)
ts.plot(dcoalt)
fit <- lm(dcoalt~as.numeric(1:length(dcoalt)))
abline(fit, col='red')
abline(h=mean(dcoalt), col='blue')
acf(dcoalt, lag.max=50)
pacf(dcoalt, lag.max=50)
ddcoalt <- diff(dcoalt,lag=12, differences=1)
par(mfrow=c(1,2))
ts.plot(ddcoalt)
fit <- lm(ddcoalt~as.numeric(1:length(ddcoalt)))
abline(fit, col='red')
abline(h=mean(ddcoalt), col='blue')
hist(ddcoalt, col='light blue', xlab='', main='Histogram of ddcoalt')
acf(ddcoalt, lag.max=50)
pacf(ddcoalt, lag.max=50)
var(coalt)
var(dcoalt)
var(ddcoalt)
library(qpcR)
df <- expand.grid(p=0:2, q=0:3, P=0:1, Q=0:1)
df <- cbind(df, AICc=NA)
for (i in 1:nrow(df)) {
```

```r
sarima.obj <- NULL
try(arima.obj <- arima(coalt, order=c(df$p[i], 1, df$q[i]),
seasonal=list(order=c(df$P[i], 1, df$Q[i]), period=12),
method="ML"))
if (!is.null(arima.obj)) { df$AICc[i] <- AICc(arima.obj) }
# print(df[i, ])
}
df[which.min(df$AICc), ]
sort(df$AICc, decreasing=F)
df
arima(coalt, order=c(1,1,1), seasonal=list(order=c(0,1,1), period=12), method="ML")
AICc(arima(coalt, order=c(1,1,1), seasonal=list(order=c(0,1,1), period=12), method="ML"))
# Model A
arima(coalt, order=c(1,1,1), seasonal=list(order=c(1,1,1), period=12), method="ML")
AICc(arima(coalt, order=c(1,1,1), seasonal=list(order=c(1,1,1), period=12), method="ML"))
# Model B
arima(coalt, order=c(0,1,2), seasonal=list(order=c(0,1,1), period=12), method="ML")
AICc(arima(coalt, order=c(0,1,2), seasonal=list(order=c(0,1,1), period=12), method="ML"))
library(UnitCircle)
# For model A
par(mfrow=c(1,4))
uc.check(pol_=c(1,-0.5136), plot_output=T)
uc.check(pol_=c(1,0.049), plot_output=T)
uc.check(pol_=c(1,-0.961), plot_output=T)
uc.check(pol_=c(1,-0.7955), plot_output=T)
# For model B
par(mfrow=c(1,2))
uc.check(pol_=c(1,-0.4381,-0.294), plot_output=T)
uc.check(pol_=c(1,-0.8568), plot_output=T)
par(mfrow=c(1,3))
fit_a <- arima(coalt, order=c(1,1,1), seasonal=list(order=c(1,1,1), period=12), method="ML")
res_a <- residuals(fit_a)
hist(res_a,density=20, breaks=20, col='blue', xlab='', prob=TRUE)
m_a <- mean(res_a)
std_a <- sqrt(var(res_a))
curve(dnorm(x,m_a,std_a), add=T)
plot.ts(res_a)
fitt_a <- lm(res_a~as.numeric(1:length(res_a)))
abline(fitt_a, col='red')
abline(h=mean(res_a), col='blue')
qqnorm(res_a, main="Normal Q-Q Plot for Model A")
qqline(res_a, col='blue')
par(mfrow=c(1,2))
acf(res_a, lag.max=50)
pacf(res_a, lag.max=50)
shapiro.test(res_a)
Box.test(res_a, lag=12, type=c("Box-Pierce"), fitdf=4)
Box.test(res_a, lag=12, type=c("Ljung-Box"), fitdf=4)
Box.test((res_a)^2, lag=12, type=c("Ljung-Box"), fitdf=0)
ar(res_a, aic=TRUE, order.max=NULL, method=c("yule-walker"))
par(mfrow=c(1,3))
fit_b <- arima(coalt, order=c(0,1,2), seasonal=list(order=c(0,1,1), period=12), method="ML")
res_b <- residuals(fit_b)
```

```r
hist(res_b,density=20, breaks=20, col='blue', xlab='', prob=TRUE)
m_b <- mean(res_b)
std_b <- sqrt(var(res_b))
curve(dnorm(x,m_b,std_b), add=T)
plot.ts(res_b)
fitt_b <- lm(res_b~as.numeric(1:length(res_b)))
abline(fitt_b, col='red')
abline(h=mean(res_b), col='blue')
qqnorm(res_b, main="Normal Q-Q Plot for Model B")
qqline(res_b, col='blue')
par(mfrow=c(1,2))
acf(res_b, lag.max=50)
pacf(res_b, lag.max=50)
shapiro.test(res_b)
Box.test(res_b, lag=12, type=c("Box-Pierce"), fitdf=3)
Box.test(res_b, lag=12, type=c("Ljung-Box"), fitdf=3)
Box.test((res_b)^2, lag=12, type=c("Ljung-Box"), fitdf=0)
ar(res_b, aic=TRUE, order.max=NULL, method=c("yule-walker"))
# install.packages("forecast")
library(forecast)
forecast(fit_b)
pred.tr <- predict(fit_b, 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(coalt, xlim=c(120, length(coalt)+12), ylim=c(120, max(U.tr)))
lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")
points((length(coalt)+1):(length(coalt)+12),pred.tr$pred, col="red")
ts.plot(coal, xlim = c(120,length(coalt)+12), ylim = c(120,max(U.tr)), col="red")
lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")
points((length(coalt)+1):(length(coalt)+12), pred.tr$pred, col="black")
```