Time Series Analysis of US Monthly Gas Supply(274 Final Project)

Bochen Wang

2022-12-9

1. Abstract

This times series project is about studying and predicting the data of US monthly gas supply from Jan. 1992 to Dec. 2007.In this data prediction, I have used the modeling processes of SARIMA models to fit the data and utilized it for predictions. By analyzing the acf/pacf, comparing AICcs, analysis of residuals, I selected the best model for prediction, and the results are looking great for the prediction outcomes. I also did the spectral analysis for the periodicity checking. The conclusion is that our model prediction power is fine with all the predicted two years of data lies inside the confidance interval.

2. Introduction

Project interest details:

Due to recent feelings about the increasing prices of gas money has brought to many people's attention. I decided to analyze the supply side of amount of gas data to study its trend and give people insights of supply side of data. In this project, I plan to analyze the monthly gas supply data in the US from 1992.1 to 2007.12 for the gas time series data prediction. In this case, I will use the data from 1992.1 to 2005.12 for training the model and use the 2006.1 to 2007.12 data for validation and prediction.

Techniques and Results:

In analyzing my time series data, I have used box-cox transformation for data to be more Gaussian, acf/pacf identification, AICc compare for model identification and checking model unit roots, analysis of residuals, Shapiro-Wilk normality test, Box-Pierce test, Box-Ljung test, for diagnostic checking, h-steps ahead predictions. In my result, my model successfully pass the test and the residuals plot reassembles normal distribution, my prediction of 24 observations are all within 95% confidence interval, positively speaking. Negatively speaking, in my residual plot's pacf, there are peaks in lag 1 and lag 2 that is outside the confidence interval. I tried to use AR(1) to my model which ends up with unit roots. However, I found the optimal one for prediction which is described as the following.

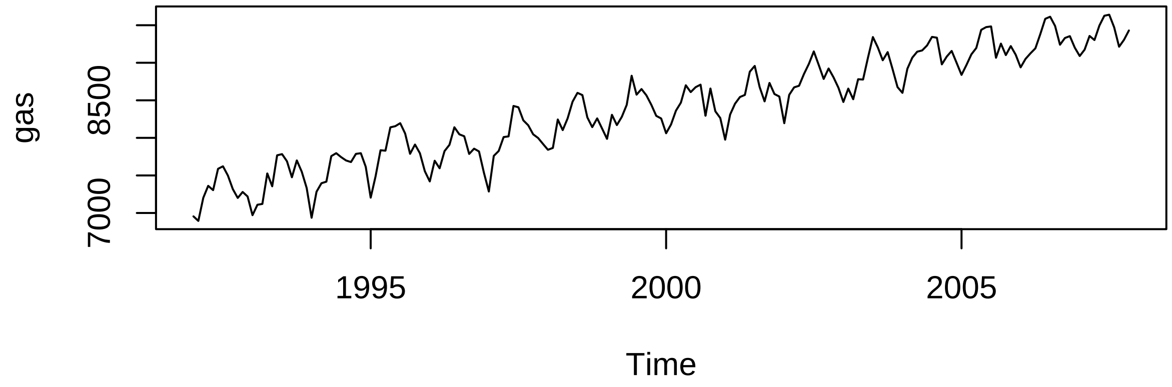
Sources:

Data collection : https://www.eia.gov/naturalgas/data.php Software: R Studio and R Markdown.

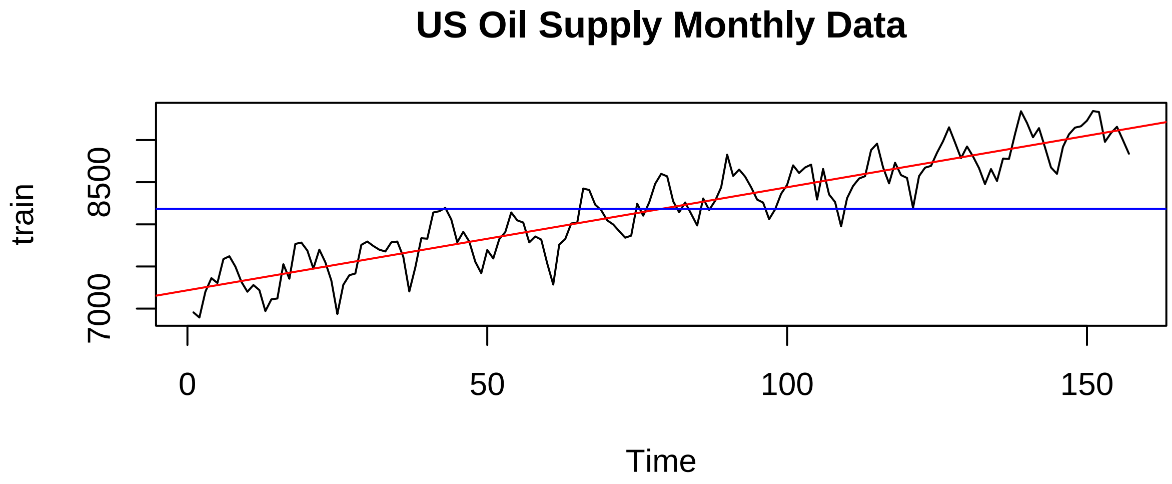
3. Model Building

Data Input:

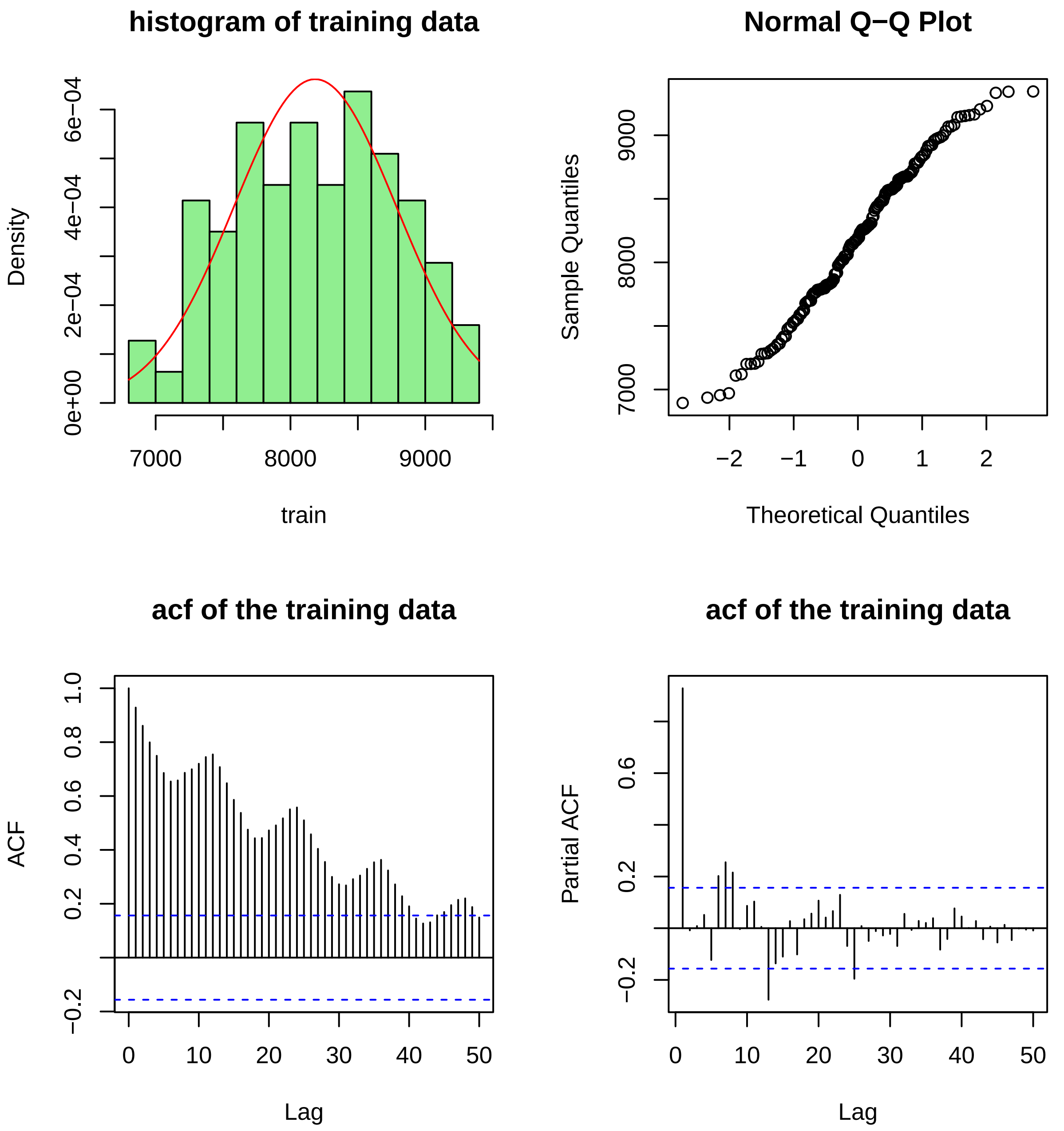
A little note: For my data, since the data I have is the weekly data from the origin website, I used python code to output a csv file that is the average of the monthly data using pandas dataframe. The code for data retrieval and transformation is included in the appendix.



Testing and Training split and added lines for trend and mean:

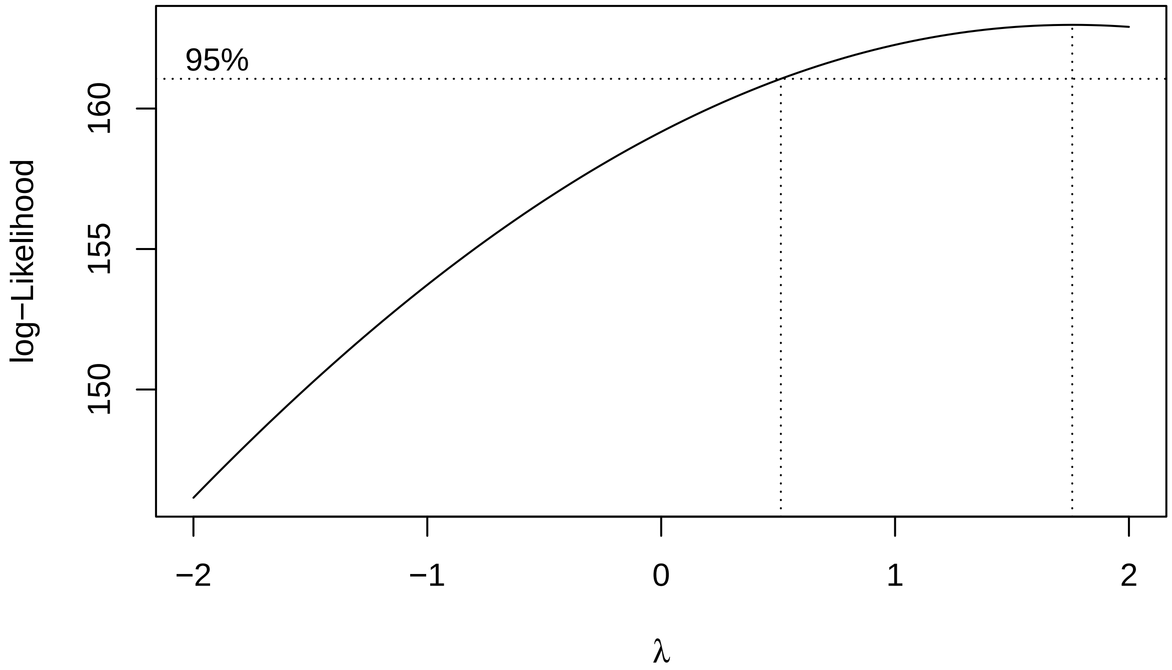


## Normality checking and acf/pacf:



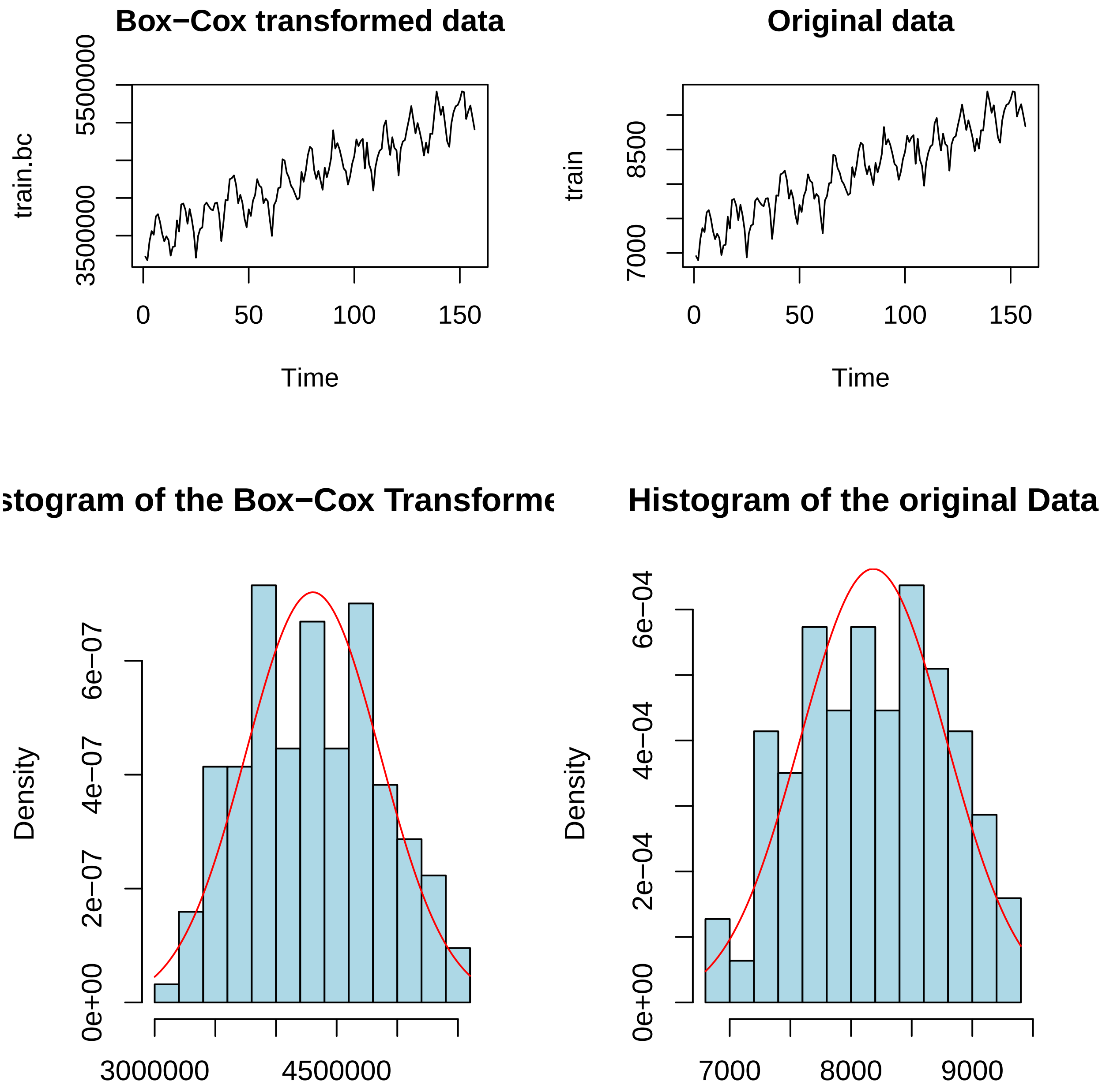
We can see from the graph that is is a little bit skewed to the left from the histogram. We can see that there is a seasonal trend for seasonal data from acf/pacf.

Because there is skewed data, I want to try to preform box-cox transformation and see how it goes:



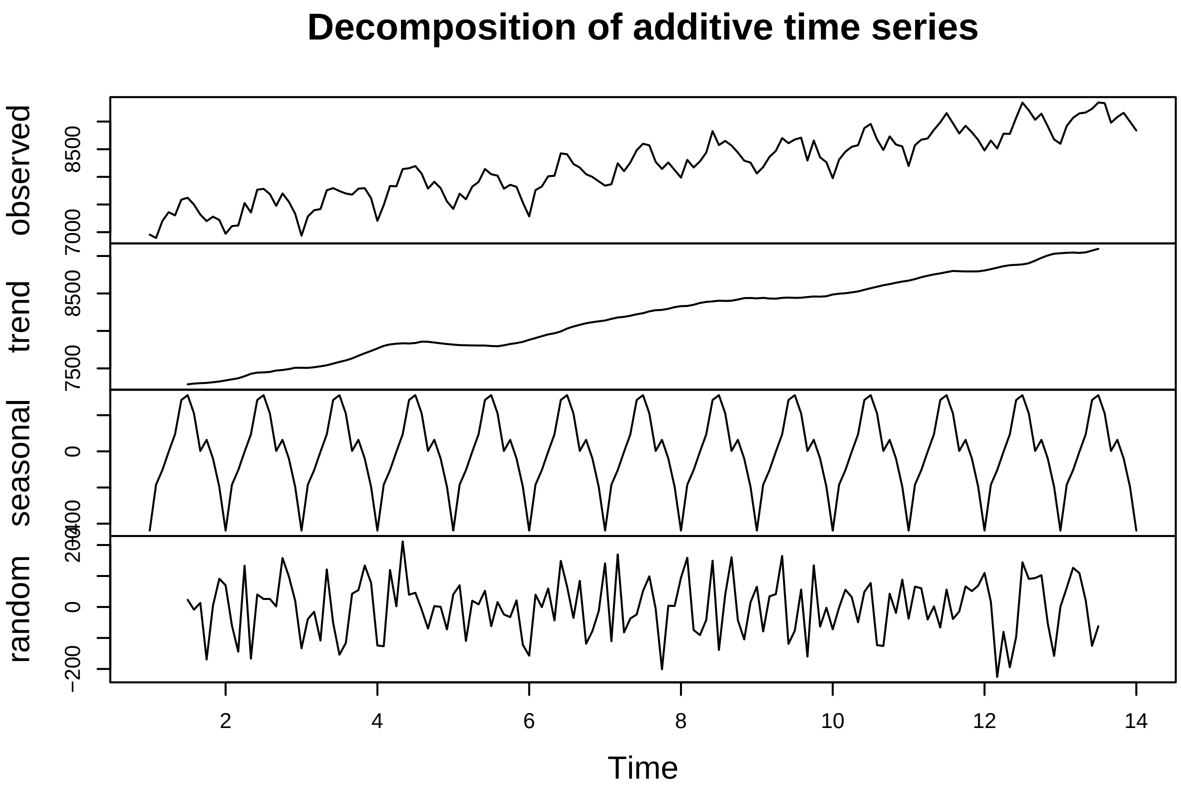
Because 1 is in my confidence interval, I also considered not transforming the data. However, the transformed data makes it difficult for me to identify the acf/pacf for appropriate models so I used non-transformed data. The comparison are shown later.

Compare the transformed data and original data, plot the transformed histogram:



We can see that they both approximately fit normal. So, my final decision is to us the original training data.

Decompose Data:



We can see from the decomposed data that there is a seasonal part and a trend in this data.

Remove trend and seasonality:

```
ds_train <- diff(train, 12)
dst_train <- diff(ds_train, 1)
var(train)
```

```
## [1] 362999.1
```

```
var(ds_train)
```

```
## [1] 26928
```

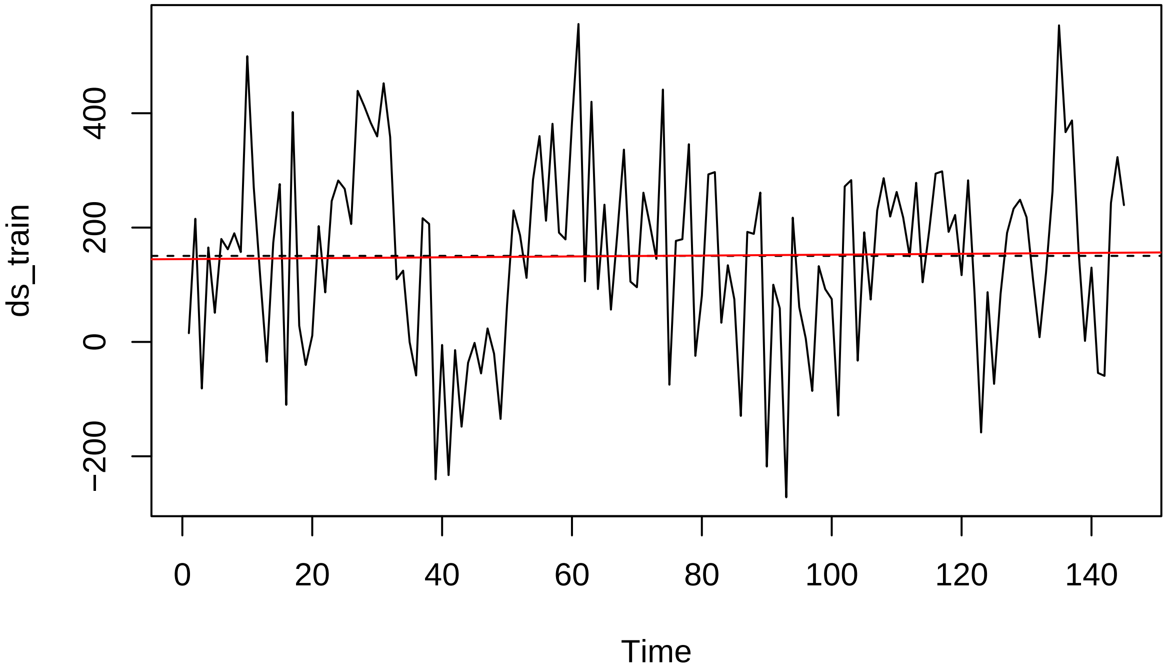
```
var(dst_train)
```

```
## [1] 37886.17
```

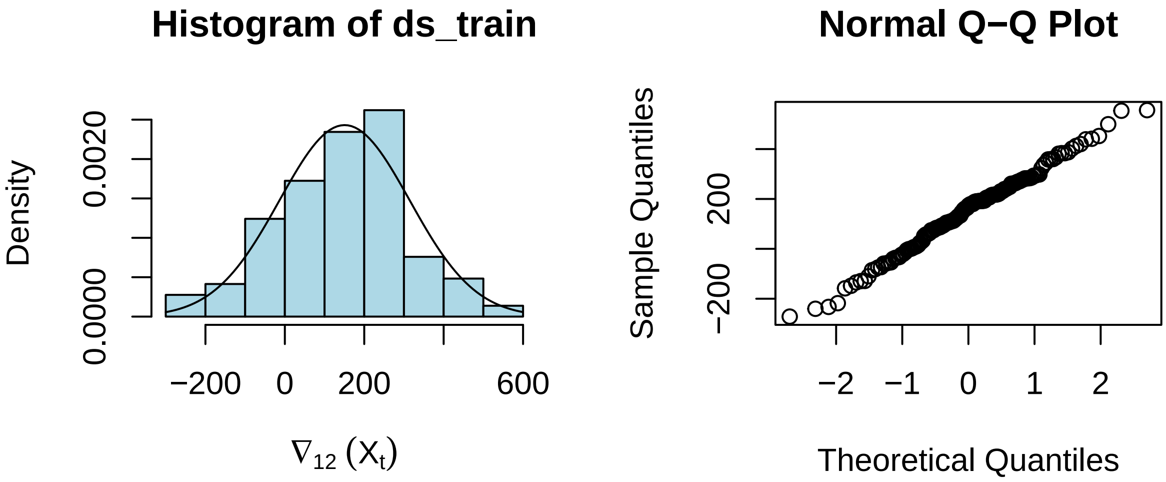
The three are r generated variances. The first is transformed data variance. The second is differenced at $\nabla_{12}X_t$.

The third is $\nabla_{12}\nabla_1 X_t$. We can see that de-seasonalized data have the lowest variance.

Observed that after removing trend this model have variance increased, so just remove seasonality.

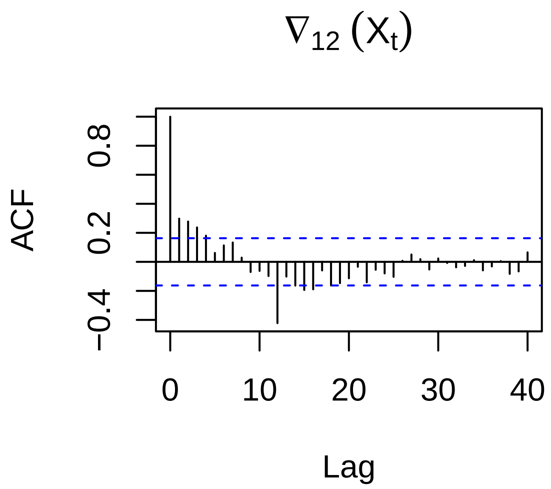


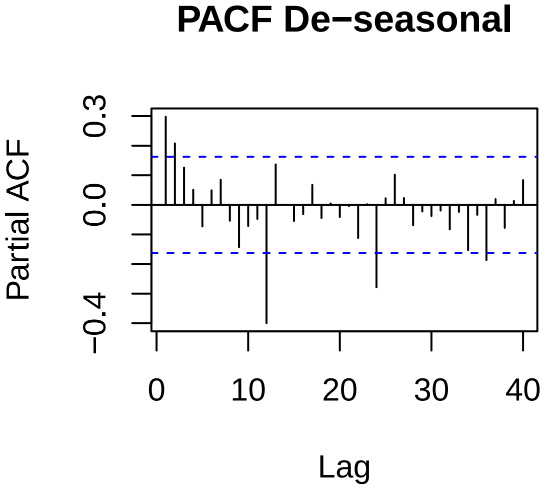
Plot the d_seasonalized data:



It might seems good Guassian but choose the d_seasonal one because it has lower variance

Plot acf and pacf for the data original data:





acf maybe lag 1, lag 2, lag 3, lag 4, lag 12 means s=12,might q = 2 or 3 or 4 need to check different q because we are considering MA part also. pacf maybe lag 1, lag 2, lag 3, lag 11, lag 12, lag 13 means p = 2 pacf mostly affected by seasonal MA part so we can try out different AICCs s =12
from both plot we can be sure that Q = 1

Model Checking comparing AICc:

I tried pure MA but residual acf and pacf shows ar part for non-seasonal.

Then I also considered the AR(1) but the results having unit roots and the residual acf/pacf plots suggest that there is a MA part which brings us here for considering AR(2) and adding MA part.

```
##
## Call:
## arima(x = train, order = c(2, 0, 0), seasonal = list(order = c(0, 1, 1), period = 12),
##     method = "ML")
##
## Coefficients:
##          ar1     ar2     sma1
##       0.5395  0.4601  -0.9678
## s.e.  0.0753  0.0753   0.1197
##
## sigma^2 estimated as 14929:  log likelihood = -916.08,  aic = 1840.16
```

```
## [1] 1840.316
```

```
##
## Call:
## arima(x = train, order = c(2, 0, 1), seasonal = list(order = c(0, 1, 1), period = 12),
##     fixed = c(NA, NA, NA, NA), method = "ML")
##
## Coefficients:
##          ar1      ar2      ma1     sma1
##       1.0640  -0.0641  -0.7906  -0.9714
## s.e.  0.1209   0.1209   0.0856   0.0801
##
## sigma^2 estimated as 12319:  log likelihood = -902.82,  aic = 1815.64
```

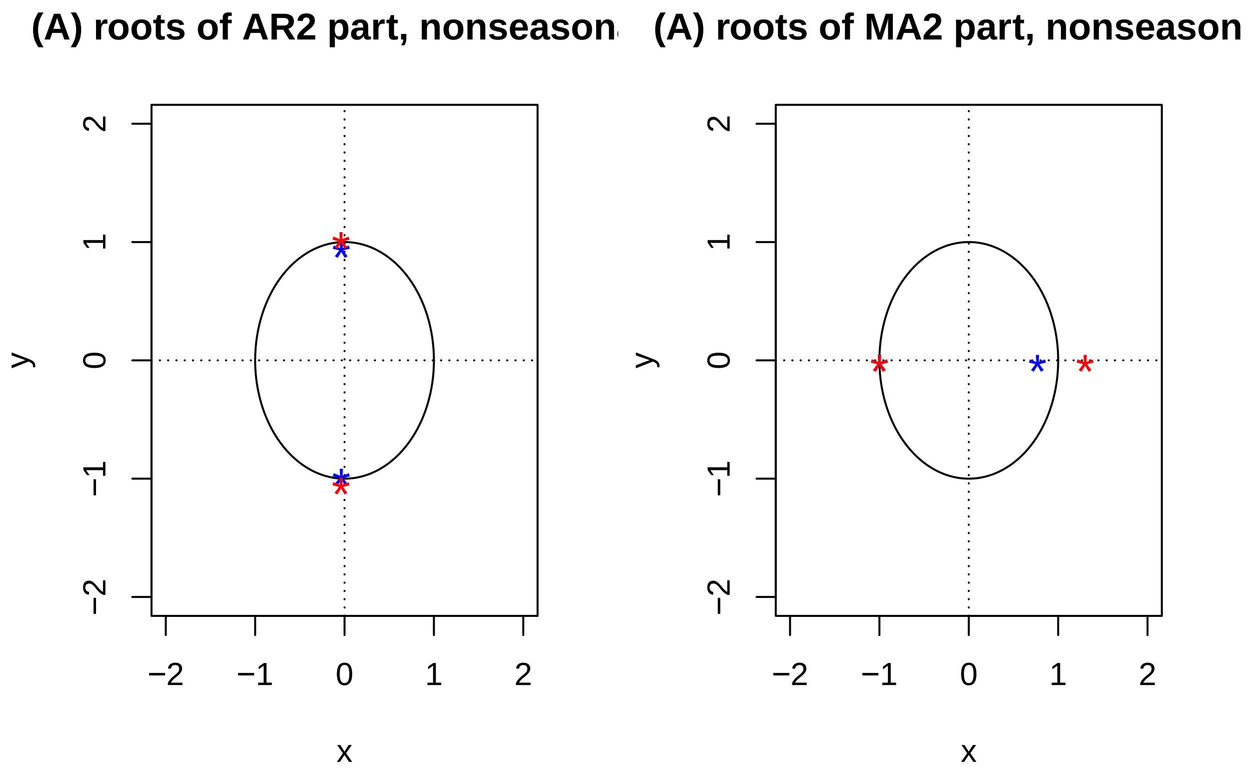
```
## [1] 1815.902
```

confidance interval suggests that the AR part is not working well because it suggests the ϕ_2 to have 0 in confidence interval

```
##
## Call:
## arima(x = train, order = c(2, 0, 2), seasonal = list(order = c(0, 1, 1), period = 12),
##     fixed = c(NA, NA, NA, NA, NA), method = "ML")
##
## Coefficients:
##          ar1     ar2     ma1      ma2     sma1
##       0.0729  0.9271  0.2319  -0.7681  -0.9776
## s.e.  0.0391  0.0391  0.0980   0.0852   0.0826
##
## sigma^2 estimated as 11766:  log likelihood = -901.9,  aic = 1815.8
```

```
## [1] 1816.197
```

It seems fine for confidence interval but having unit roots:

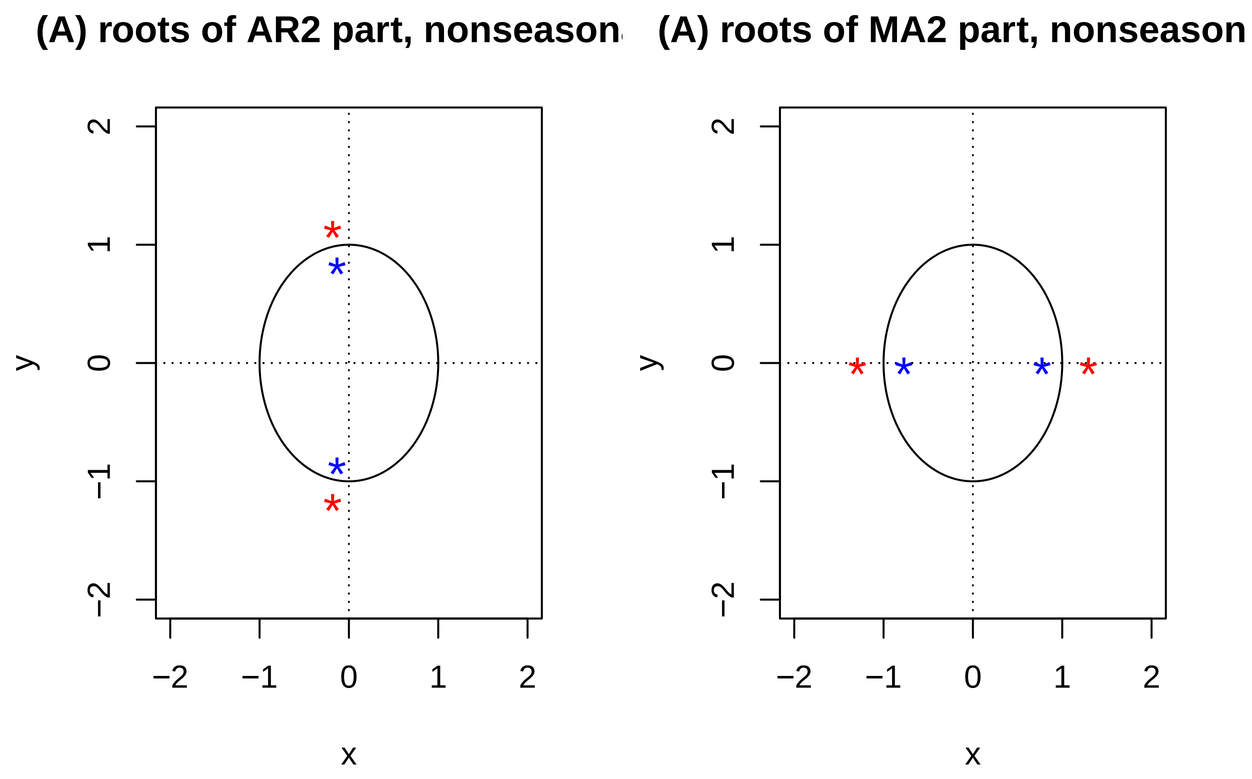


so next I tried to set the θ_1 to be zero and I got the following results

```
##
## Call:
## arima(x = train, order = c(2, 0, 2), seasonal = list(order = c(0, 1, 1), period = 12),
##     fixed = c(NA, NA, 0, NA, NA), method = "ML")
##
## Coefficients:
##          ar1     ar2  ma1      ma2     sma1
##       0.2677  0.7323    0  -0.5979  -0.9749
## s.e.  0.0733  0.0733    0   0.1005   0.0738
##
## sigma^2 estimated as 12270:  log likelihood = -902.77,  aic = 1815.54
```

```
## [1] 1815.934
```

Checking for unit roots:



AR and MA outside unit root passed

so far this model seems fine and we can use it for testing later residual plots.

I also tried to set MA part to 3

```
##
## Call:
## arima(x = train, order = c(2, 0, 3), seasonal = list(order = c(0, 1, 1), period = 12),
##     fixed = c(NA, NA, NA, NA, NA, NA), method = "ML")
##
## Coefficients:
##          ar1     ar2     ma1      ma2      ma3     sma1
##       0.0699  0.9301  0.2440  -0.7737  -0.0178  -0.9776
## s.e.  0.0416  0.0416  0.1124   0.0868   0.0932   0.0819
##
## sigma^2 estimated as 11767:  log likelihood = -901.88,  aic = 1817.76
```

```
## [1] 1818.324
```

It is vary close forθ_3 to be zero so I stopped here.

Therefore SARIMA(2,0,2)(0,1,1)_12 have the lowest AICcs and do not have unit roots after comparing it to other.

Diagnostic Checking:

Fitting the model and checking the residual plots.

Calculate the mean:

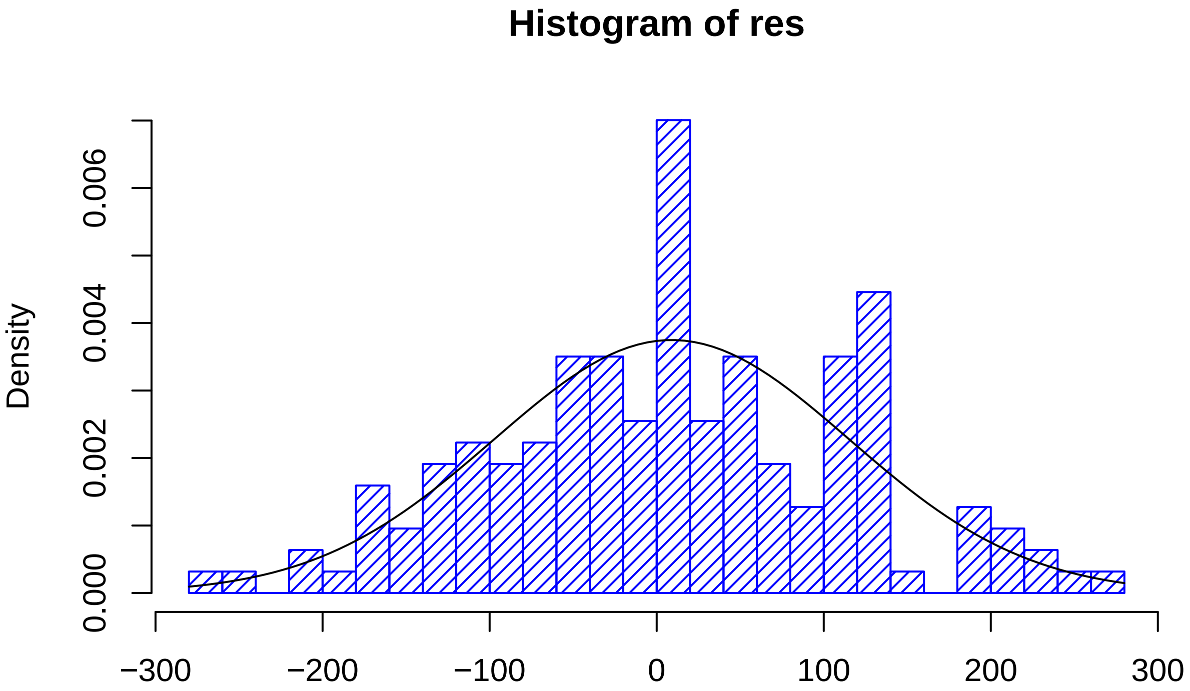
```
## [1] 8.981813
```

Variance:

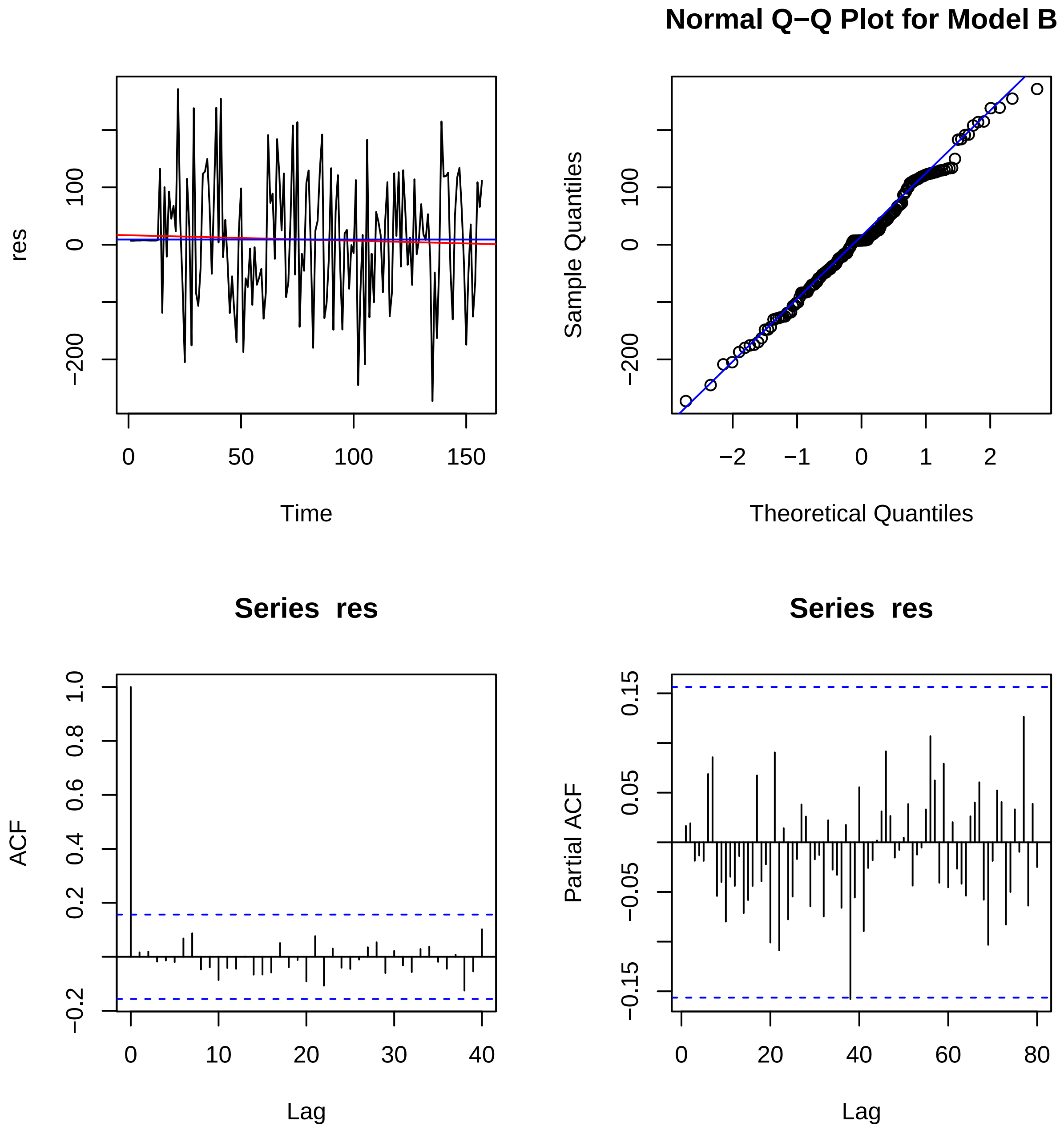
```
## [1] 11328.08
```

Even though I have larger mean and variance than the homework's datasets, my residual have minimized its mean and variance on its original data scale.

Residual behaves like normal and mean is small relatively speaking:



Plot the residual and acf/pacf of the residuals:



The plots shows no trend no visible change of variance, no seasonality. Histogram and Q-Q plot look OK All acf and pacf of residuals are within confidence intervals and can be counted as zeros because it passed the tests as shown later.

Run tests:

```
##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.99318, p-value = 0.6683
```

```
##
##  Box-Pierce test
##
## data:  res
## X-squared = 4.4896, df = 9, p-value = 0.8763
```

```
##
##  Box-Ljung test
##
## data:  res
## X-squared = 4.7976, df = 9, p-value = 0.8516
```

```
##
##  Box-Ljung test
##
## data:  res^2
## X-squared = 14.986, df = 13, p-value = 0.3082
```

```
##
## Call:
## ar(x = res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  11328
```

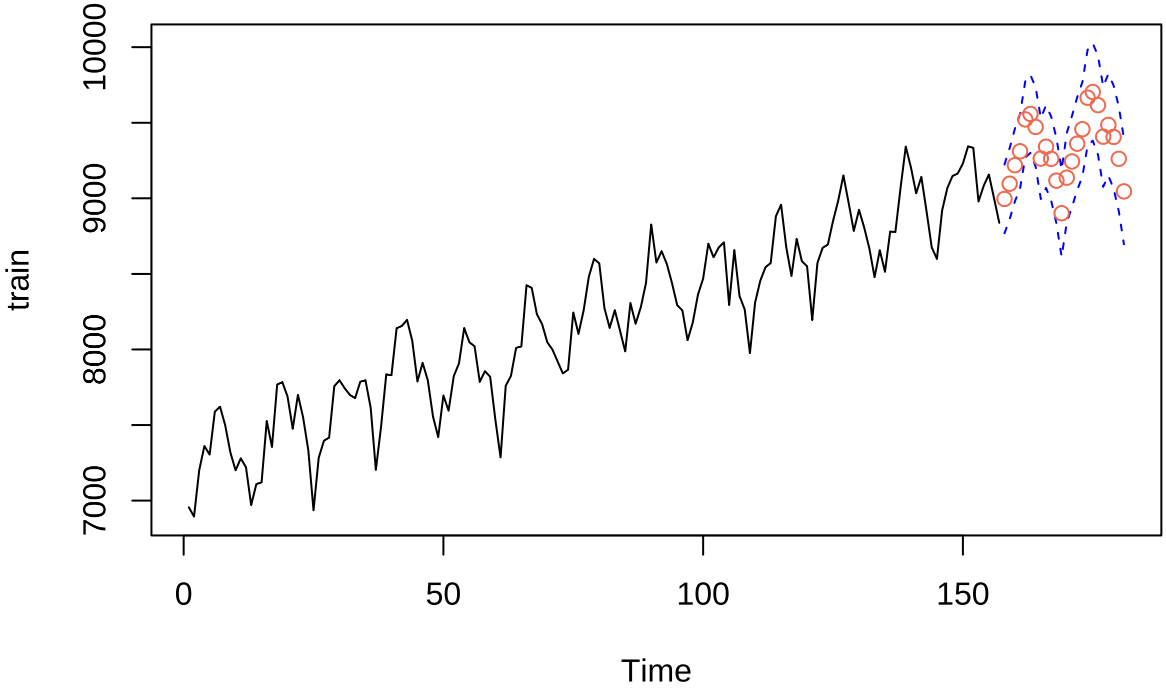
All tests p-value is larger than 0.05 passed. This is great! We can move on to prediction.

4. Prediction and Forecasting:

We have picked the model

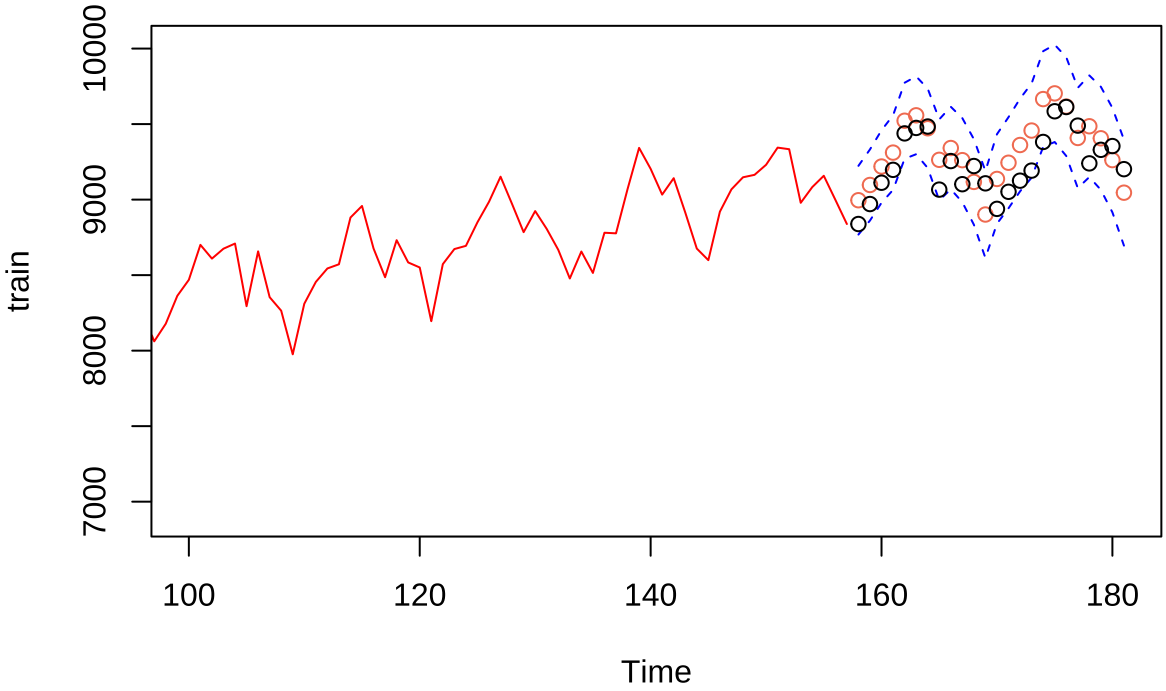
$$\nabla_{12}(1 - 0.2677_{(0.0733)}B - 0.7323B^2_{(0.0733)})X_t = (1 - 0.5979B^2_{(0.1005)})(1 - 0.9749B^{12}_{(0.0738)})Z_t$$

Draw the prediction interval for 2 year in the future:



We can see that here the prediction interval and the predicted values have approximately simulated the trend of 2 year monthly data in the future.

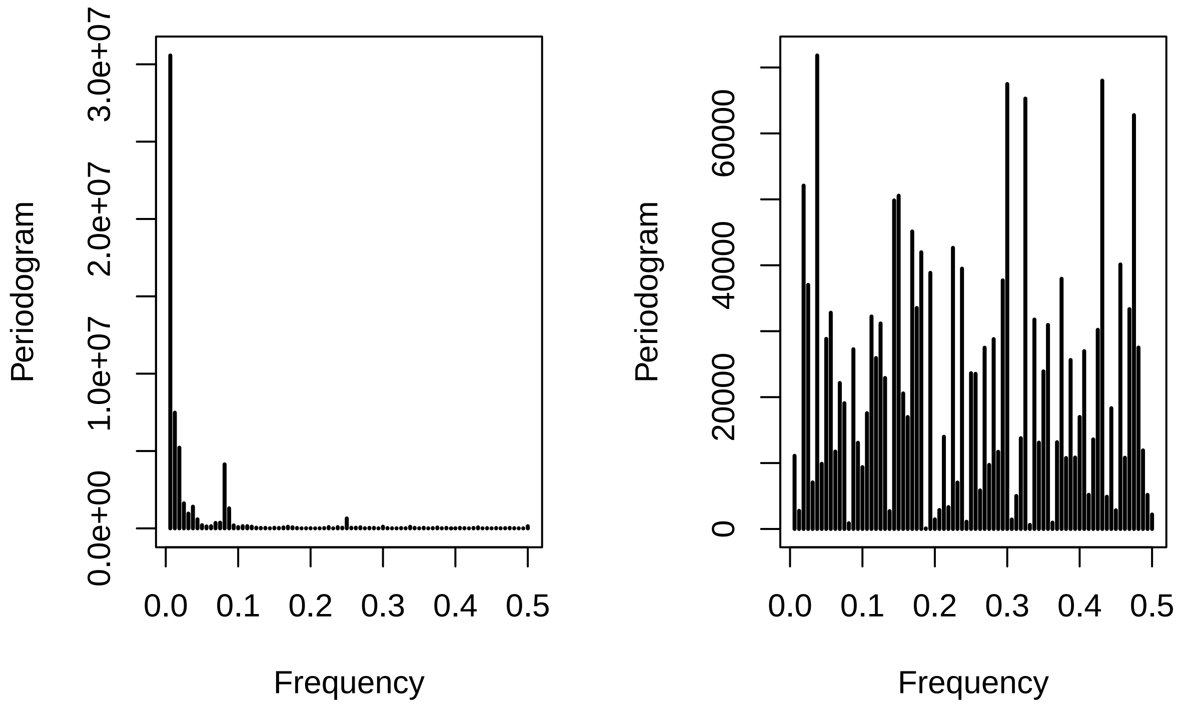
Then I zoom in on 100 and later and added the testing set, our model seems to preform well:



This is great! We can see that for two years prediction that all observations are in side confidence interval and follows the approximate pattern. (Time as in month since 1992.1 and the black dots are the actual values)

5. Spectral Analysis:

plot the periodogram for both Data and Residuals:



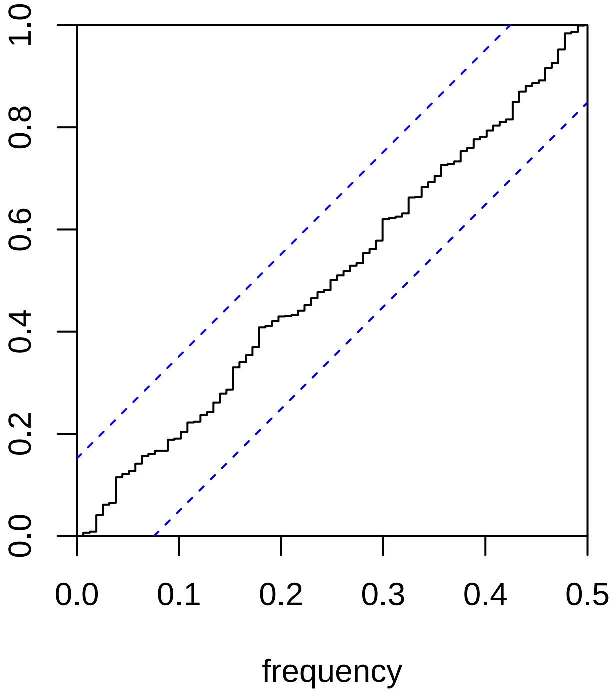
We can see a period here from the graph of the training data on the left, there exhibit no obvious seasonal and periodical trend on the residual plots.

Preform Fisher and Kolmogorov-Smirnov Test:

```r
library("GeneCycle")
fisher.g.test(res)
```

```
## [1] 0.9675539
```

From the fisher's p-value with the respect to a 0.05 level, we fail to reject the hypothesis that the residuals does not resemble a white noise process.



Also, from the Kolmogorov-Smirnov test we can see that the lines are all withing confidence interval that can be counted as passing this test.

6. Conclusion:

The goal of the project is to predict US gas supply with data presented and time series analysis. I managed to achieve good predicting with solid time series analysis that models the US monthly gas supply in two years. The model I used is SARIMA(2,0,2)(0,1,1)_12 and parameterized as the following:

$$\nabla_{12}(1 - 0.2677_{(0.0733)}B - 0.7323B^2_{(0.0733)})X_t = (1 - 0.5979B^2_{(0.1005)})(1 - 0.9749B^{12}_{(0.0738)})Z_t$$

this model work fine by the validation of diagnostic checking and spectral analysis. I examined the acf and pacf for further validating the residuals that it resembles white noise process. Thanks to professor Feldman's great course contents and instruction, I finally achieved using time series analysis to build model.

7. Reference:

People Helped: Prof. Raya Fledman and Lecture slides & notes

Data collection : https://www.eia.gov/naturalgas/data.php

Software: R Studio and R Markdown.

Package and Functions Reference: https://cran.r-project.org/web/packages/forecast/forecast.pdf

https://cran.r-project.org/web/packages/TSA/TSA.pdf

https:
//github.com/nickpoison/astsa/blob/master/fun_with_astsa/fun_with_astsa.md#arimaˆsimulation

https://cran.r-project.org/web/packages/astsa/astsa.pdf

8. Appendix:

```
#input data
#### Python code for data retrieval:
#python starts:
import pandas as pd
a = pd.read_csv('Gas_Supply.csv')
def cleandf(df):
    start = df.iat[0,0][0:2]
    all = []
    cell = []
    cell.append(df.iat[0,0])
    cell.append(df.iat[0,1])
    all.append(cell)
    for i in range(len(df)-1):
        temp = df.iat[i+1,0][0:2]
        celln = []
        if(start != temp):
            start = temp
            celln.append(df.iat[i+1,0])
            celln.append(df.iat[i+1,1])
            all.append(celln)
    dfn = pd.DataFrame(all, columns = ['Date', 'Usage'])
    return dfn
dfnew = cleandf(a)
dfnew.to_csv('US_Gas_Supply')
#python code ends.


#R code:

library(tsdl)
gas.csv <- read.csv("US_Gas_Supply.csv")
rusage <- gas.csv$Usage
usage = rev(rusage)
gas = ts(usage, start= c(1992,1), frequency = 12)
plot.ts(gas)
```

```
#test train split and draw test
train <- gas[c(1:157)]
test <- gas[c(157:180)]
plot.ts(train, main = "US Oil Supply Monthly Data")
fit <- lm(train ~ as.numeric(1:length(train)))
abline(fit, col = "red")
abline(h = mean(train), col = "blue")

#histogram and qqplots for normality checking
par(mfrow = c(1,2))
hist(train, col = "lightgreen", main = "histogram of training data", freq=F)
curve(dnorm(x,mean(train), sqrt(var(train))), col = "red", add = TRUE)
qqnorm(train)

#acf/pacf checking
par(mfrow=c(1,2))
acf(train, lag.max = 50, main = "acf of the training data")
pacf(train, lag.max = 50, main = "acf of the training data")

#box-cox transform and compare it to original
library(MASS)
t <- 1:length(train)
bcTransform <- boxcox(train ~ t)
lambda <- bcTransform$x[which.max(bcTransform$y)]
train.bc = (1/lambda)*((train)^lambda - 1)
par(mfrow=c(1,2))
plot.ts(train.bc,main = "Box-Cox transformed data")
plot.ts(train,main = "Original data")
par(mfrow=c(1,2))
hist(train.bc, col="light blue", xlab="",
     main="Histogram of the Box-Cox Transformed Data",freq = F)
m1 <- mean(train.bc)
std1 <- sqrt(var(train.bc))
curve(dnorm(x, m1, std1), col="red", add=TRUE)
hist(train, col="light blue", xlab="", main="Histogram of the original Data",freq = F)
m2 <- mean(train)
std2 <- sqrt(var(train))
curve(dnorm(x, m2, std2), col="red", add=TRUE)

#decompose data
y <- ts(as.ts(train), frequency = 12)
decomp <- decompose(y)
plot(decomp)
#difference data
ds_train <- diff(train, 12)
dst_train <- diff(ds_train, 1)
var(train.bc)
var(ds_train)
var(dst_train)

#plot data
plot.ts(ds_train)
abline(h=mean(ds_train), lty=2)
```

```
fitdiff1 <- lm(ds_train ~ as.numeric(1:length(ds_train)))
abline(fitdiff1, col="red")
par(mfrow=c(1,2))
hist(ds_train, col="light blue", xlab=expression(nabla[12]~(X[t])), prob=TRUE)
m <- mean(ds_train)
std <- sqrt(var(ds_train))
curve(dnorm(x,m,std), add=TRUE)
qqnorm(ds_train)

#plotted acf and pacf for the data
par(mfrow=c(1,2))
acf(ds_train, lag.max=40, main=expression(nabla[12]~(X[t])))
pacf(ds_train, lag.max=40, main="PACF De-seasonal")

#plot the roots for unit root checking
plot.roots <- function(ar.roots=NULL,
                       ma.roots=NULL, size=2, angles=FALSE, special=NULL,
                       sqecial=NULL,my.pch=1,first.col="blue",
                       second.col="red",main=NULL)
{xylims <- c(-size,size)
      omegas <- seq(0,2*pi,pi/500)
      temp <- exp(complex(real=rep(0,length(omegas)),imag=omegas))
      plot(Re(temp),Im(temp),typ="l",xlab="x",ylab="y",xlim=xylims,
           ylim=xylims,main=main)
      abline(v=0,lty="dotted")
      abline(h=0,lty="dotted")
      if(!is.null(ar.roots))
        {
          points(Re(1/ar.roots),Im(1/ar.roots),col=first.col,pch=my.pch)
          points(Re(ar.roots),Im(ar.roots),col=second.col,pch=my.pch)
        }
      if(!is.null(ma.roots))
        {
          points(Re(1/ma.roots),Im(1/ma.roots),pch="*",cex=1.5,col=first.col)
          points(Re(ma.roots),Im(ma.roots),pch="*",cex=1.5,col=second.col)
        }
      if(angles)
        {
          if(!is.null(ar.roots))
            {
              abline(a=0,b=Im(ar.roots[1])/Re(ar.roots[1]),lty="dotted")
              abline(a=0,b=Im(ar.roots[2])/Re(ar.roots[2]),lty="dotted")
            }
          if(!is.null(ma.roots))
            {
              sapply(1:length(ma.roots),
                     function(j) abline(a=0,b=Im(ma.roots[j])/Re(ma.roots[j]),
                                        lty="dotted"))
            }
        }
      if(!is.null(special))
        {
          lines(Re(special),Im(special),lwd=2)
        }
      if(!is.null(sqecial))
        {
          lines(Re(sqecial),Im(sqecial),lwd=2)
        }
        }

#parameter estimation and AICc chenkings
arima(train, order=c(2,0,3), seasonal = list(order = c(0,1,1), period = 12),
      fixed = c( NA,
NA,NA, NA, NA, NA), method="ML")
AICc(arima(train, order=c(2,0,3), seasonal = list(order = c(0,1,1), period = 12),
           fixed = c( NA,
NA,NA, NA, NA,NA), method="ML"))

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

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

#Checking for unit roots:

#source("plot.roots.R")
library(qpcR)
plot.roots(NULL,polyroot(c(1, 0.2677,0.7323)), main="(A) roots of AR2 part, nonseasonal ")
plot.roots(NULL,polyroot(c(1,0,-0.5979)), main="(A) roots of MA2 part, nonseasonal ")

#Diagonistic Checking
library(astsa)
fit<- arima(train, order=c(2,0,2), seasonal = list(order = c(0,1,1), period = 12),  fixed = c( NA,
NA,0, NA, NA), method="ML")
res <- fit$residuals
mean(res)
var(res)
hist(res,density=20,breaks=20, col="blue", xlab="", prob=TRUE)
m1 <- mean(res)
std1 <- sqrt(var(res))
curve( dnorm(x,m1,std1), add=TRUE )
plot.ts(res)
fitt <- lm(res ~ as.numeric(1:length(res))); abline(fitt, col="red")
abline(h=mean(res), col="blue")
qqnorm(res,main= "Normal Q-Q Plot for Model B")
```

```
qqline(res,col="blue")
acf(res, lag.max=40)
pacf(res, lag.max=80)

#run tests
shapiro.test(res)
Box.test(res, lag = 13, type = c("Box-Pierce"), fitdf = 4)
Box.test(res, lag = 13, type = c("Ljung-Box"), fitdf = 4)
Box.test(res^2, lag = 13, type = c("Ljung-Box"), fitdf = 0)
#acf(res^2, lag.max=40)
ar(res, aic = TRUE, order.max = NULL, method = c("yule-walker"))

#predictions:
library(forecast)
fit.A <- arima(train, order=c(2,0,2),
               seasonal = list(order = c(0,1,1), period = 12),  fixed = c( NA,
NA,0, NA, NA), method="ML")

#plot it
pred.tr <- predict(fit.A, n.ahead = 24)
pred.orig <- pred.tr$pred
U= pred.tr$pred + 2*pred.tr$se #upper bound of prediction interval
L= pred.tr$pred - 2*pred.tr$se #lower bound
ts.plot(train, xlim=c(1,length(train)+24), ylim = c(min(train),max(U)))
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(train)+1):(length(train)+24), pred.orig, col="red")

ts.plot(train, xlim = c(100,length(train)+24),
        ylim = c(min(train),max(U)), col="red")
lines(U,col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(train)+1):(length(train)+24), pred.orig, col="green")
points((length(train)+1):(length(train)+24), test, col="black")

#Spectral Analysis
#install.packages("TSA")
#require(TSA)
#library("TSA")
TSA::periodogram(train, plot = T) #abline(h=0);
TSA::periodogram(res, plot = T)
#axis(1,at=c(0.01, 0.02, 0.03,0.083, 0.1))
library("GeneCycle")
fisher.g.test(res)
cpgram(res,main="")
```