Quarterly Australian Gross non-farm product Time Series Data Analysis

Tinhang(David) Hong

2022-11-15

Contents

Abstract:

Basic introduction:The purpose of the project is to predict the Australian Gross non-farm product prices. The interest in the project is that predicion in finance market can usually bring unseenable advantages, creating better outcomes for industries. the data set i picked was chosen from tsdl and it is the data set that is about quarterly non-farm production: pounds per cow. Jan 62 – Dec 75, also it has the appropriate size of data. I chose this dataset because it can dissected into 4 quarters to predict.

During my analysis, I came up a few differnet sarima models that fit the data set, but the final decision was sarima(p=4,d=1,q=0,P=0,D=1,Q=1,S=4). Not only because it has the lowest AICc, but also because it passes all the normality tests and its residuals stay inside the confidence interval

Introduction:

To restate my problem and goal, I am trying to predict or forecast the future values of Australian Gross non-farm product prices. The dataset appeals interesting to me because it is quarterly and it relates to my day-to-day life closely, and it is from tsdl[132]. I firstly noticed the varying of trend and that suggests some transformations might be necessary to stabalizie the dataset. Also, since it has a strong linear trend and it is seasonal, at least differencing at lag 4(quarterly dataset) and lag 1 is necessary. I then look at the acf and pacf to come up with reasonal sarima models to fit the datasets and check AICcs. Pick the models with relatively lower AICcs and do diagnostic checkings on their residuals. If they pass the tests, then we could do the final part, which is forecasting. Since we transformed our datasets using some techniques, it is necessary to forecast on untransformed dataset. The results were optimal for my final model since my prediction all fall between the upper bound and lower bound of forecasts.

Data Importation and First Step Analysis

```
## Time Series Data Library: 1 Macroeconomic time series with frequency 4
##
##                Frequency
## Subject         4
##   Macroeconomic 1
```

```
## [1] 144
```

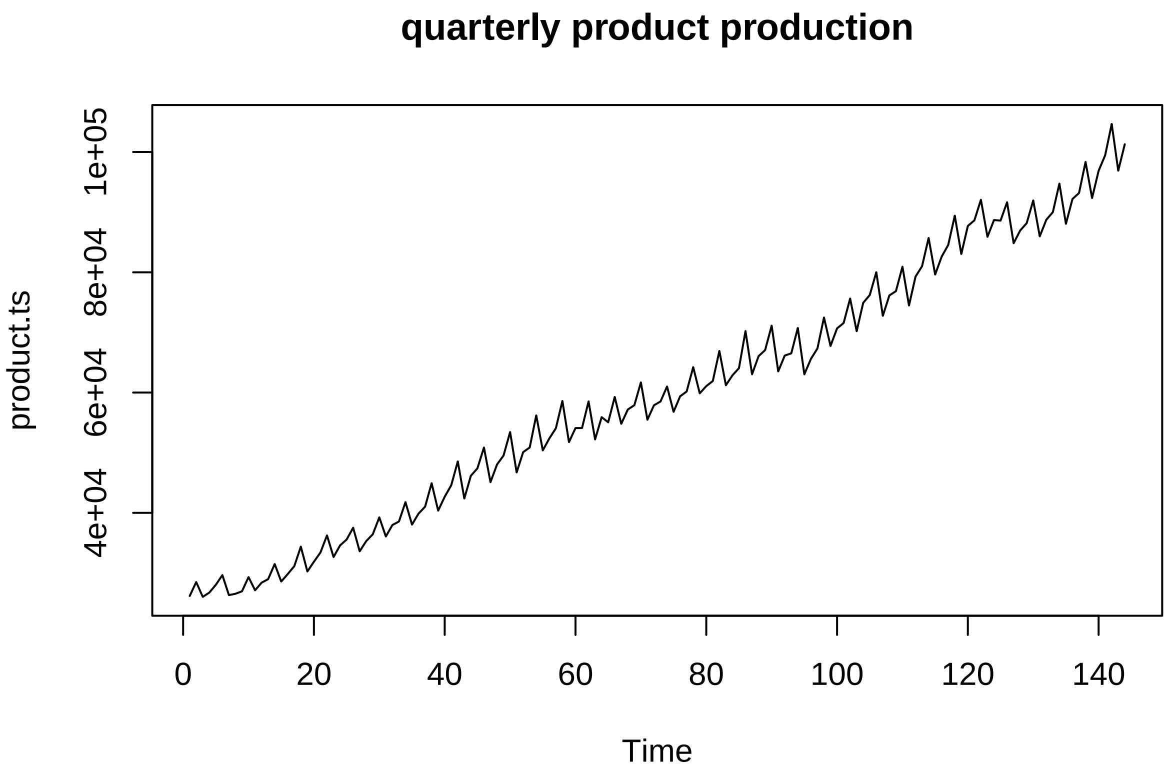
```
## [1] "Macroeconomic"
```

```
## [1] "Australian Bureau of Statistics"
```

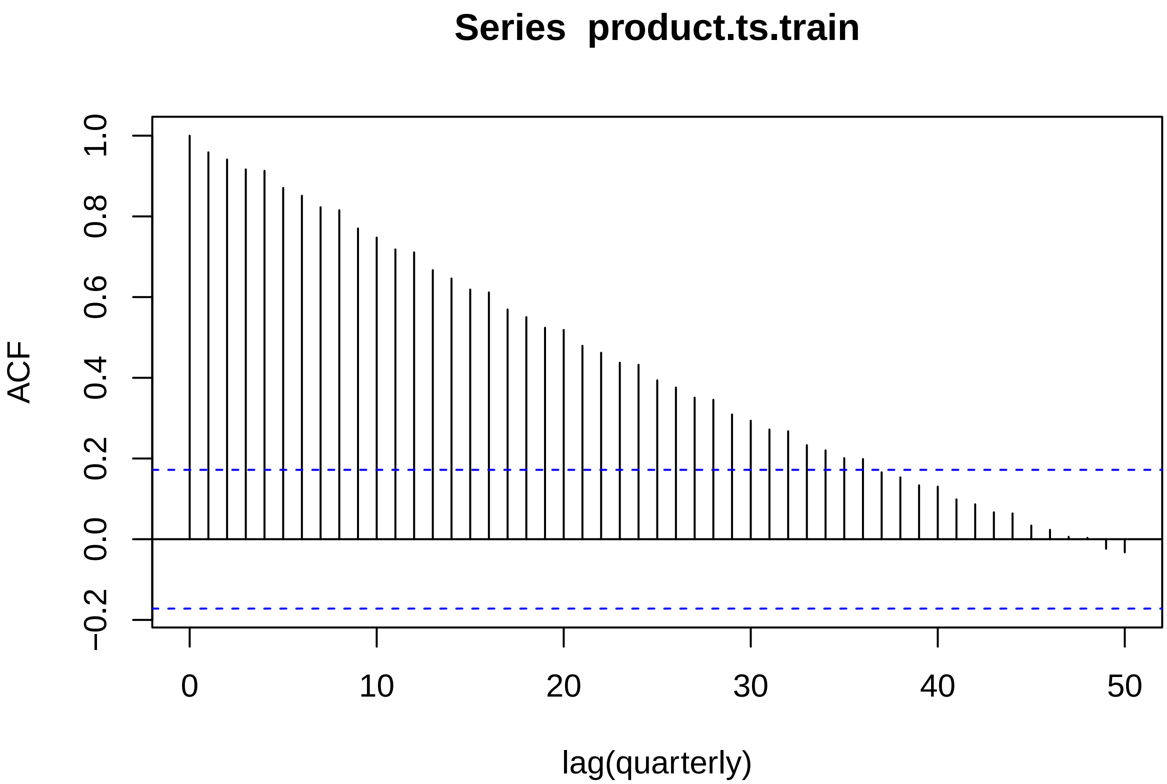
```
## [1] "Quarterly Australian Gross non-farm product: $m at average 1989/90 prices. Sep 5
```

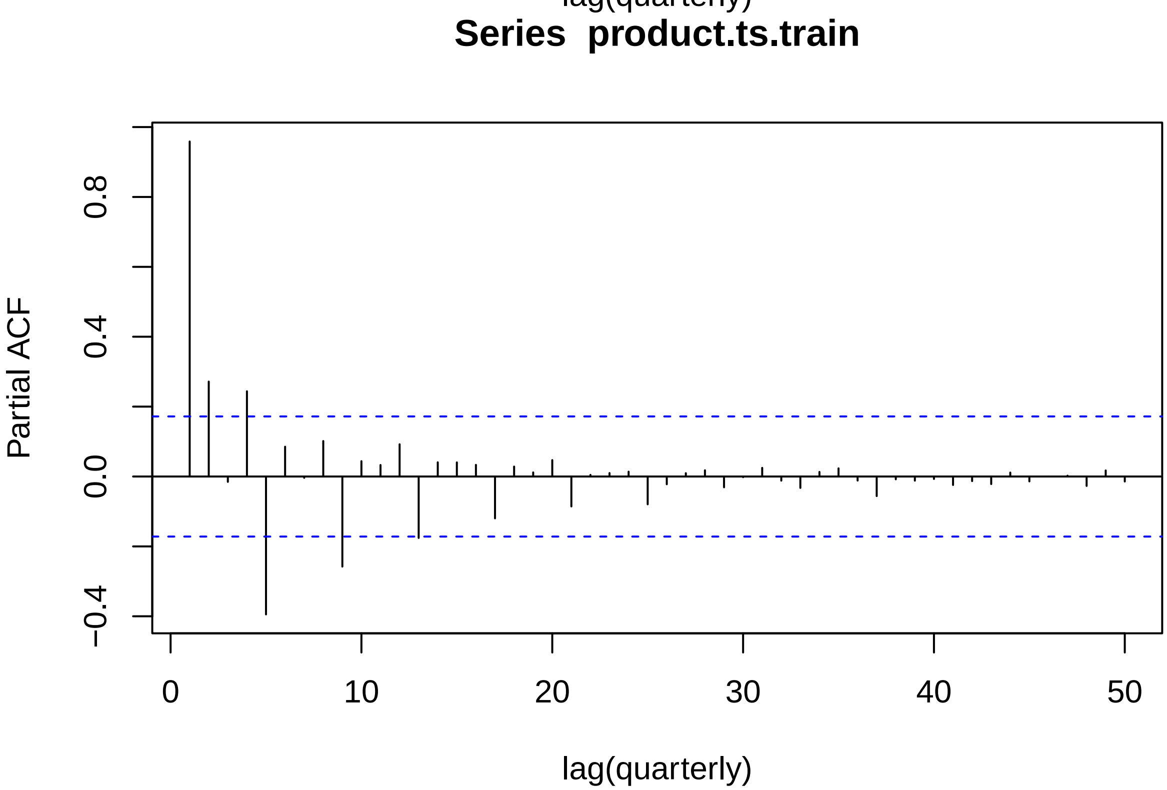
```
##        Qtr1   Qtr2   Qtr3   Qtr4
## 1959                        26169
## 1960  28492  26032  26731  28033
## 1961  29649  26317  26539  26950
## 1962  29318  27127  28386  28974
## 1963  31489  28574  29824  31133
## 1964  34381  30253  31872  33407
## 1965  36251  32649  34589  35554
## 1966  37522  33606  35305  36431
## 1967  39247  36065  37976  38553
## 1968  41779  38046  39872  41023
## 1969  44927  40364  42671  44596
## 1970  48549  42386  46163  47388
## 1971  50861  45105  48037  49532
## 1972  53440  46718  50098  50868
## 1973  56204  50382  52375  54063
## 1974  58599  51764  54092  54101
## 1975  58532  52211  55911  55060
## 1976  59270  54803  57180  57894
## 1977  61690  55483  57886  58510
## 1978  61005  56793  59376  60175
## 1979  64225  59871  61070  61918
## 1980  66917  61219  62862  64059
## 1981  70227  63042  66051  67061
## 1982  71120  63526  66156  66517
## 1983  70744  63039  65597  67339
## 1984  72476  67752  70674  71556
## 1985  75631  70204  74924  76210
## 1986  80012  72760  76153  76856
## 1987  80928  74478  79277  81011
## 1988  85705  79626  82606  84545
## 1989  89412  83040  87699  88633
## 1990  92060  85895  88690  88590
## 1991  91645  84832  86943  88171
## 1992  91942  85976  88732  90004
## 1993  94745  88057  92190  93177
## 1994  98349  92349  96878  99463
## 1995 104664  96897 101289
```

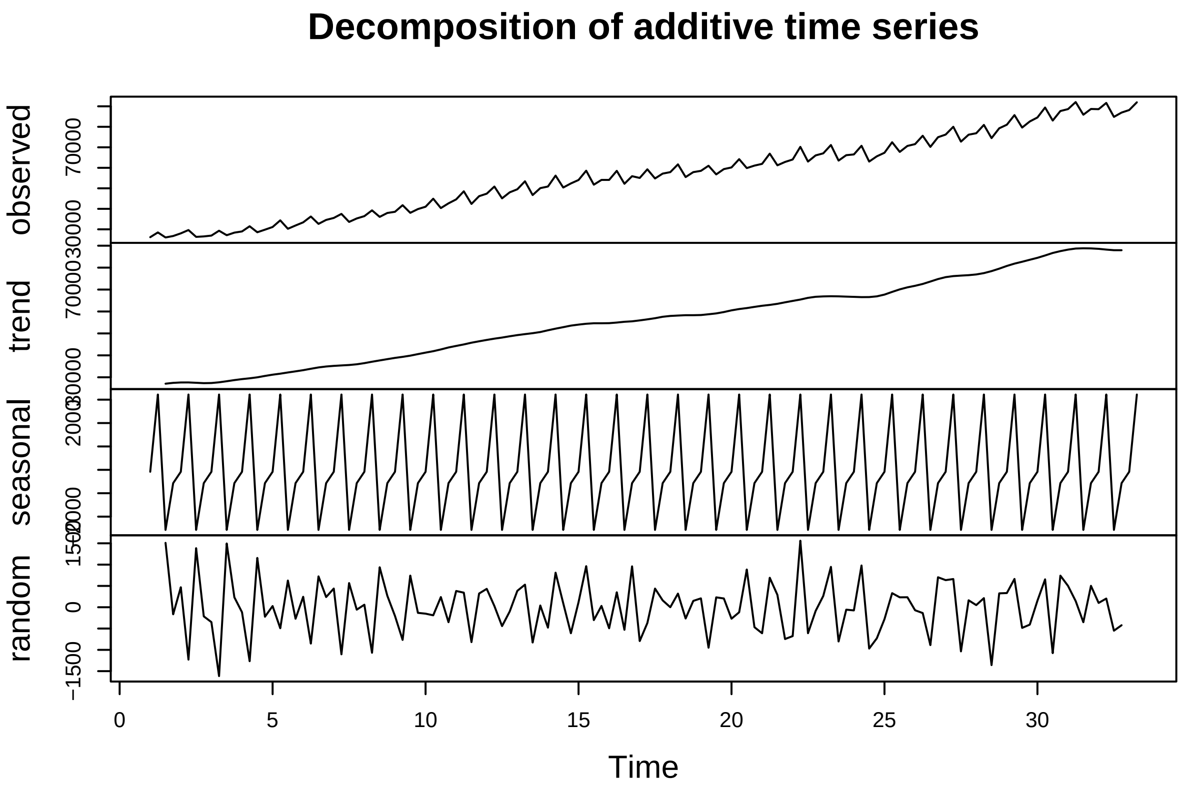
To get a glimpse of the dataset, we import the dataset as a variable and plot it as a time series graph



It seems to have a linear trend and seasonal component, but to be sure, let us plot the acf,pacf and the decomposition graph


Series product.ts.train
ACF
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
10
20
30
40
50
lag(quarterly)

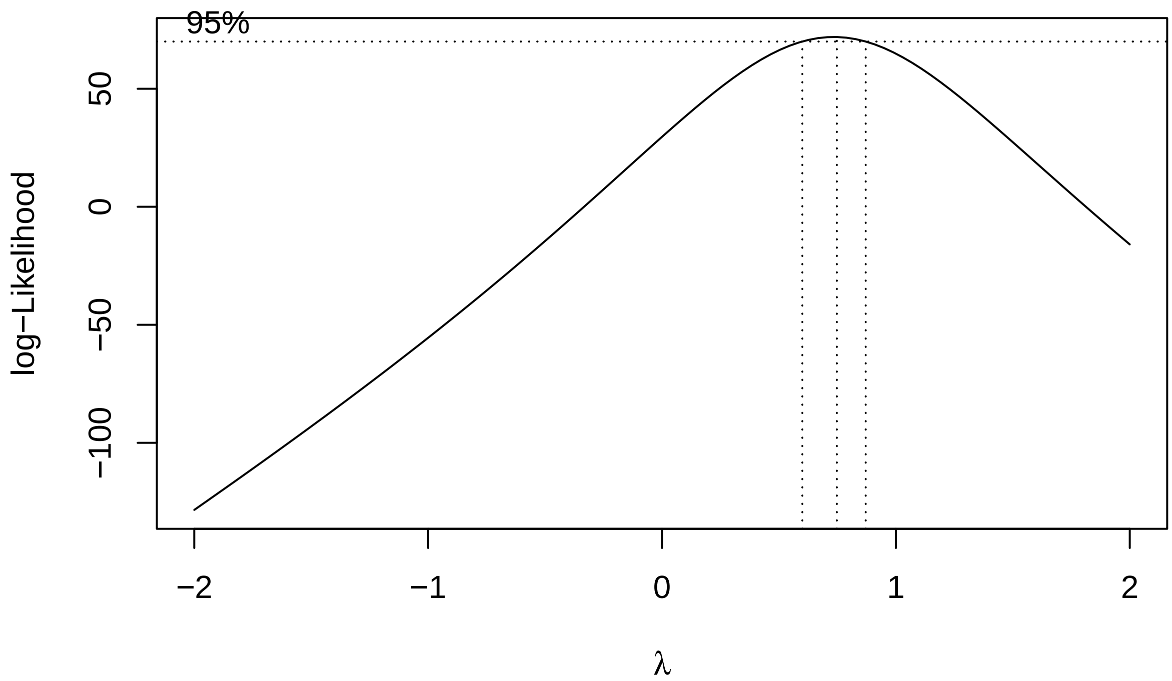

Series product.ts.train
Partial ACF
0.8
0.4
0.0
−0.4
0
10
20
30
40
50
lag(quarterly)



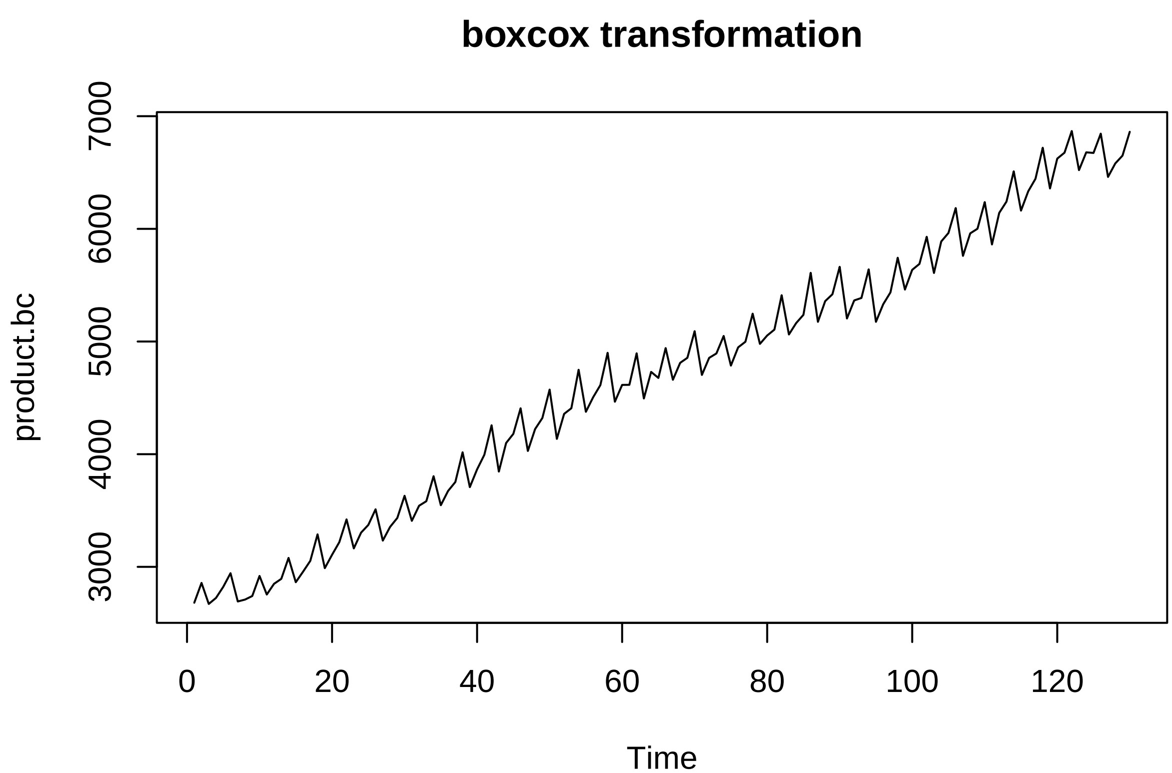
Data Transformation

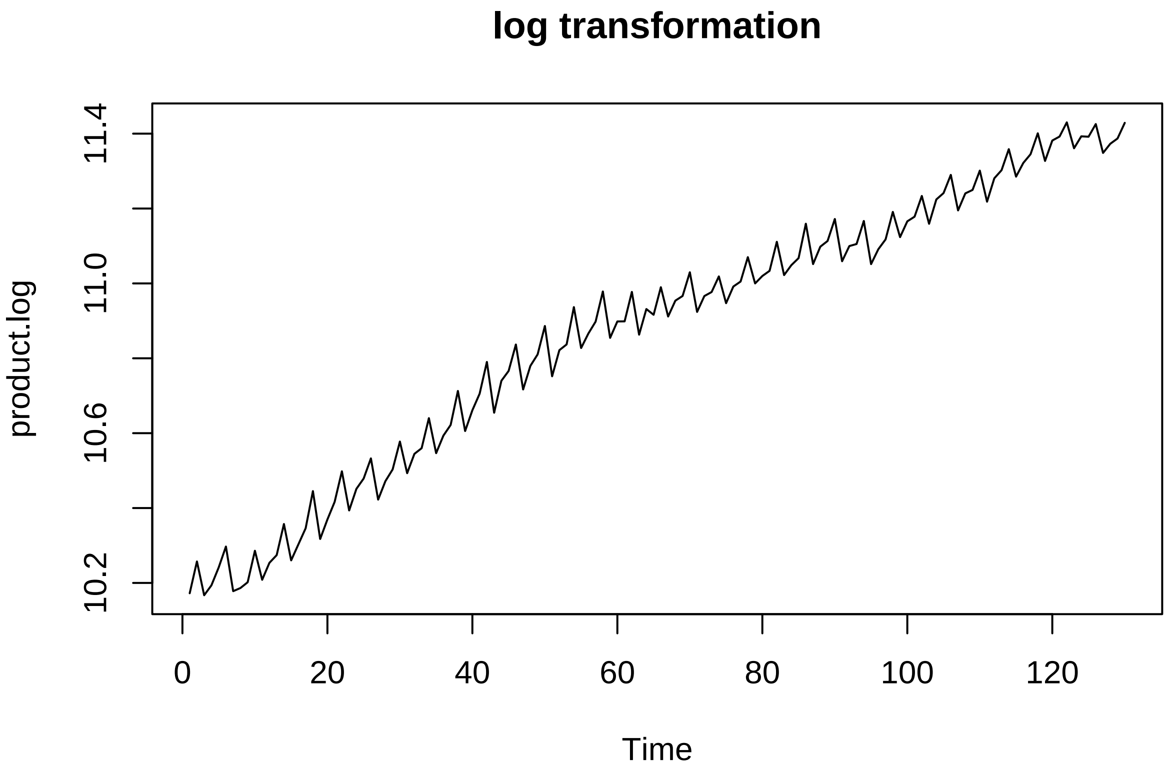
Boxcox Transformation

Now, we are sure that it has seasonal and linear increasing trend. What about variance stability? Is the variance stable? In order to find out the necessity of boxcox transformation, let us plot the λ graph and check if 1 is inside the 95 percent confidence interval.



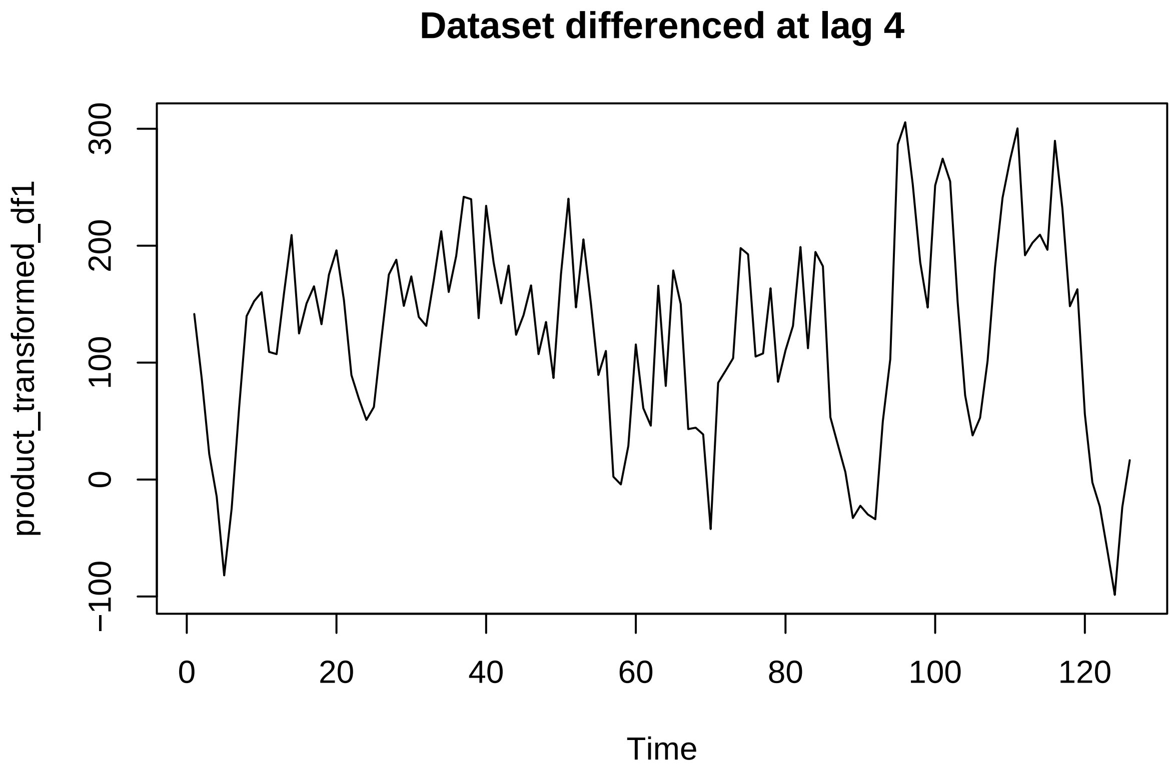
I also transfomred the dataset by using log transformation, but according to graph, it is more stable using boxcox transformation



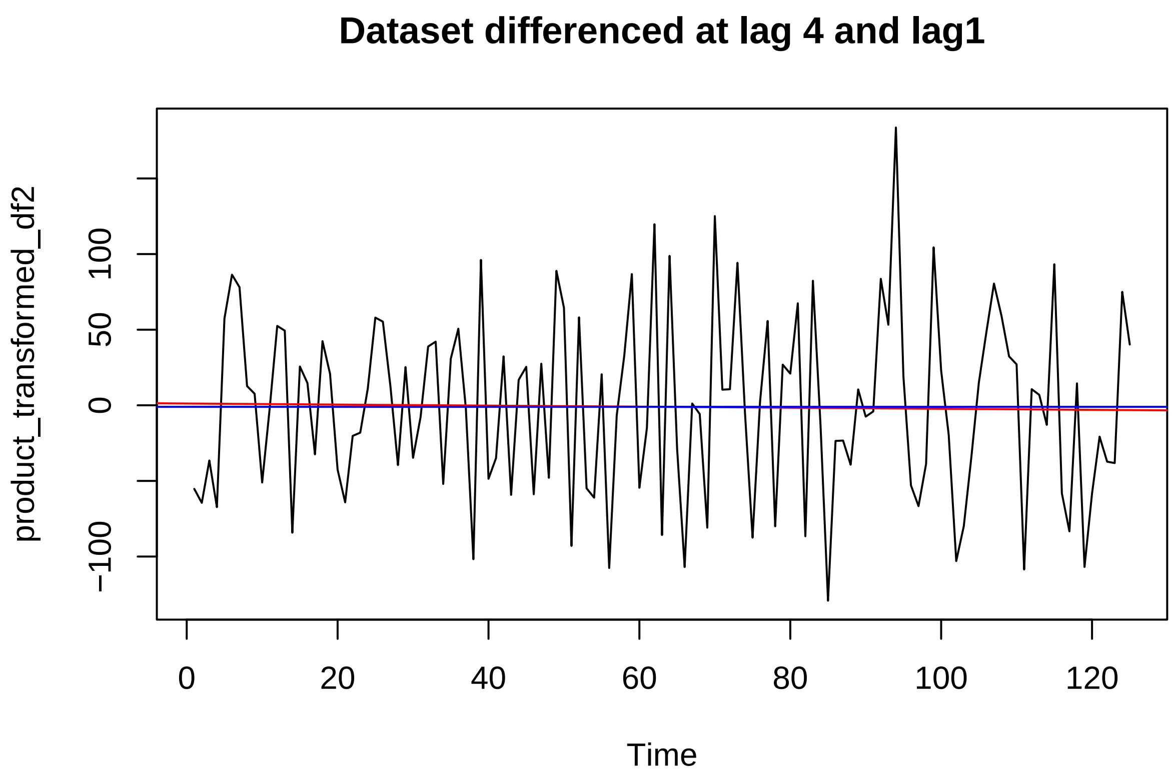


Differencing

Firstly, let us difference at lag4 and lag1 because that is the least difference we need(quarterly seasonal and linear trend). The lower variance of differencing at lag1 and lag 4 also suggests I am on the right track.

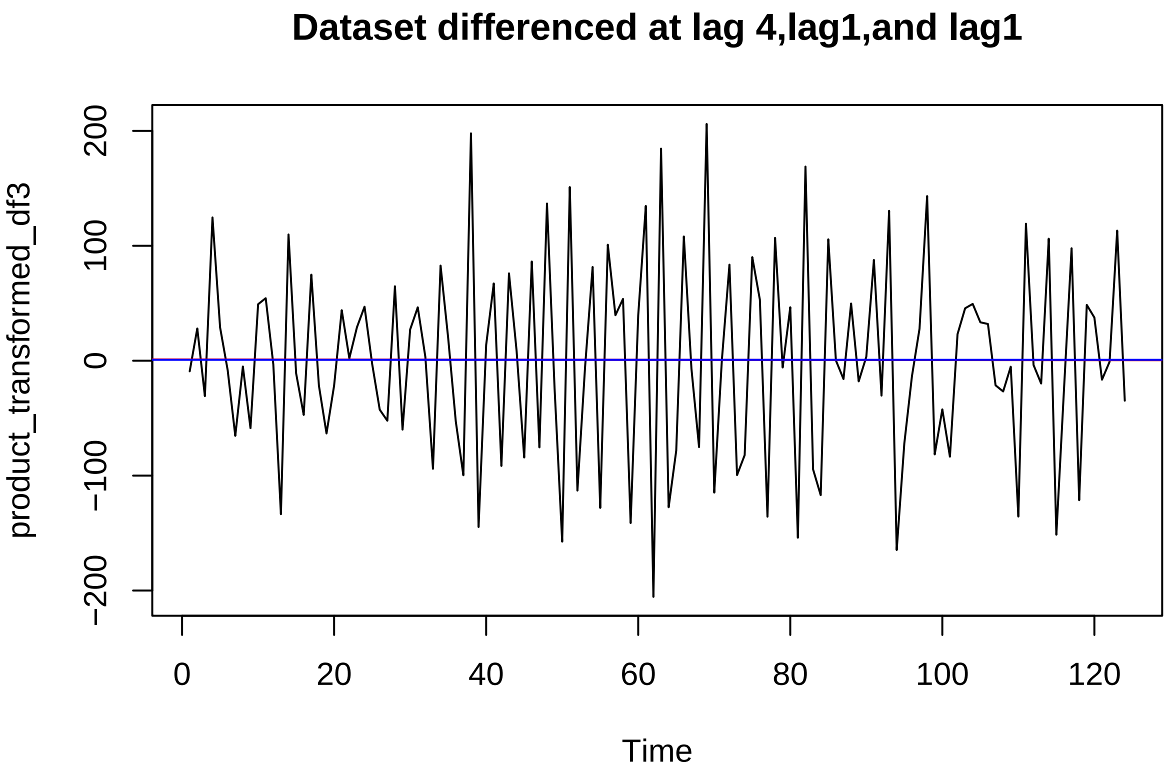



```
## [1] "the variance of orginal dataset is:  1491485.44510388"
## [1] "the variance of differencing at lag4 dataset is:  7605.94805299779"
```



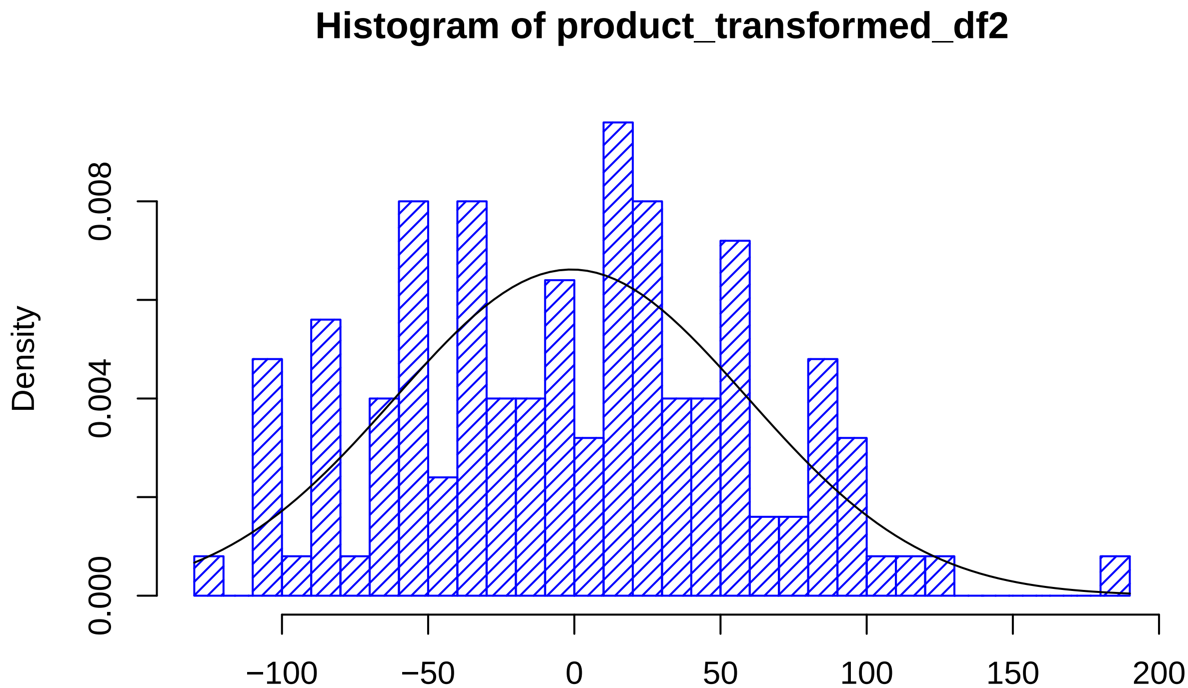

```
## [1] "the variance of differencing at lag4 and lag1 dataset is:  3635.10470112461"
```

To make sure we are not under differencing,let us difference one more lag




```
## [1] "the variance of differencing at lag4 ,lag1 and lag1 dataset is:  7482.5766638135
```

By differning one more lag, we can see the variance increases, and that suggests overdifferencing. Thus, differencing at lag4 and lag1 is the best outcome. Then, we check the differenced dataset are behaving like normal distribution to some extent.



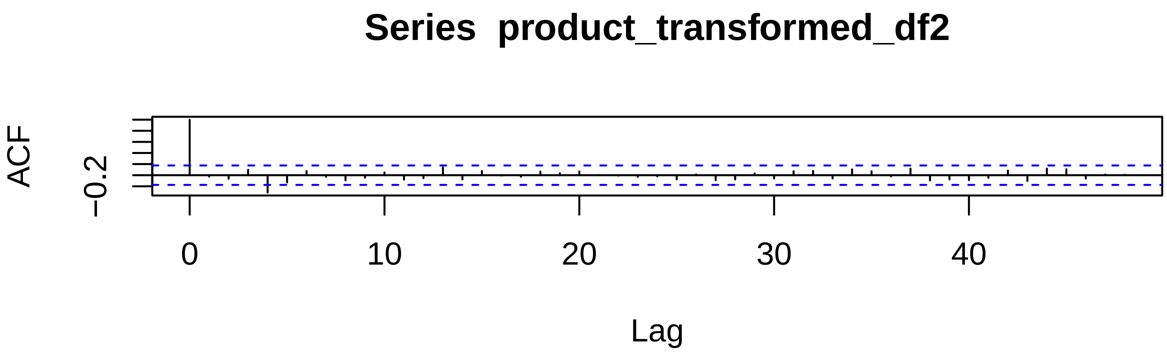
Model Construction

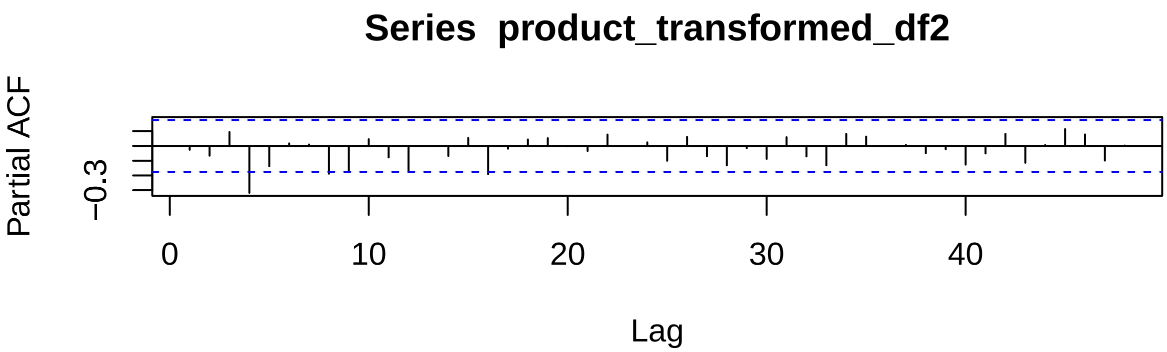
ACF and PACF analysis

First, let us graph the acf and pacf of the transformed(and differenced) dataset. Knowing this is the best choice of differencing, these acfs and pacfs are the suitable one to analyze. Since the acf cutsoff at lag4, that could suggest either Q=1 or q=4(since it is quarterly dataset). Although the pacf decays like moving average models would behave, it also has a strong "burst" at lag 4, that could suggest p=4 or P =1. Note that d=1,D=1 and S=4 because we differenced at lag1 and lag4(seasonal part). With these many different combinations, we are going to pick

- sarima(p=0,d=1,q=0,P=0,D=1,Q=1,S=4),
- sarima(p=0,d=1,q=0,P=1,D=1,Q=1,S=4),
- sarima(p=4,d=1,q=4,P=0,D=1,Q=0,S=4), and
- sarima(p=4,d=1,q=0,P=0,D=1,Q=1,S=4)

to analyze





AICcs,Invertibility and Stationarity

```
##       only Q=1  p=4,q=4  p=4,Q=1  P=1,Q=1
## [1,] 10.88018 10.88134 10.87996 10.86693
```

I combined the AICcs of four different models and integrated them into a matrix for better visual looking. It is found that sarima(p=0,d=1,q=0,P=1,D=1,Q=1,S=4), Model A and sarima(p=4,d=1,q=0,P=0,D=1,Q=1,S=4), Model B have the lowest AICcs. Thus, these are the model fits we are interested in.

The coefficients of Model A is given as follow:

```
## Coefficients
```

```
##       sar1       sma1
##  0.3030807 -0.7763828
```

Thus, the model is

$$(1-0.3030807B^4)Y_t = (1-0.5470189B^4)Z_t, \quad Z_t \sim WN\left(0, \sigma_Z^2\right), \text{ for } Y_t := (1-B)^1\left(1-B^4\right)^1 X_t$$

Note that the model is invertible and stationary because Θ and Φ are both smaller 1.Thus,the roots definitely lie outside of the unit interval

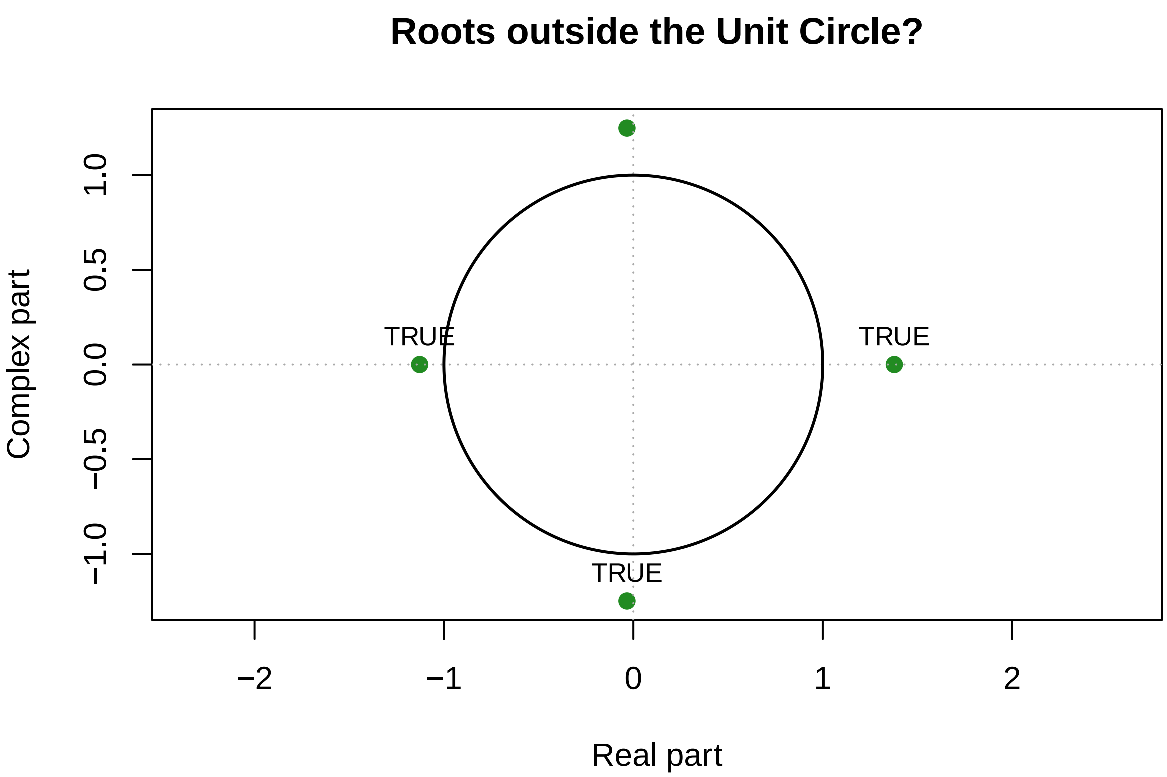
The coefficients of Model B is given as follow:

```
## Coefficients

##          ar1          ar2          ar3          ar4         sma1
## -0.203748723 -0.004944228 -0.075215667  0.412296181 -0.909658399

## [1]  0.8366285+0.9581430i -0.9278441+0.7988462i -0.9278441-0.7988462i
## [4]  0.8366285-0.9581430i

##        real   complex outside
## 1 -0.033650  1.248463    TRUE
## 2 -1.128357  0.000000    TRUE
## 3 -0.033650 -1.248463    TRUE
## 4  1.378089  0.000000    TRUE
## *Results are rounded to 6 digits.
```



Thus, the model is

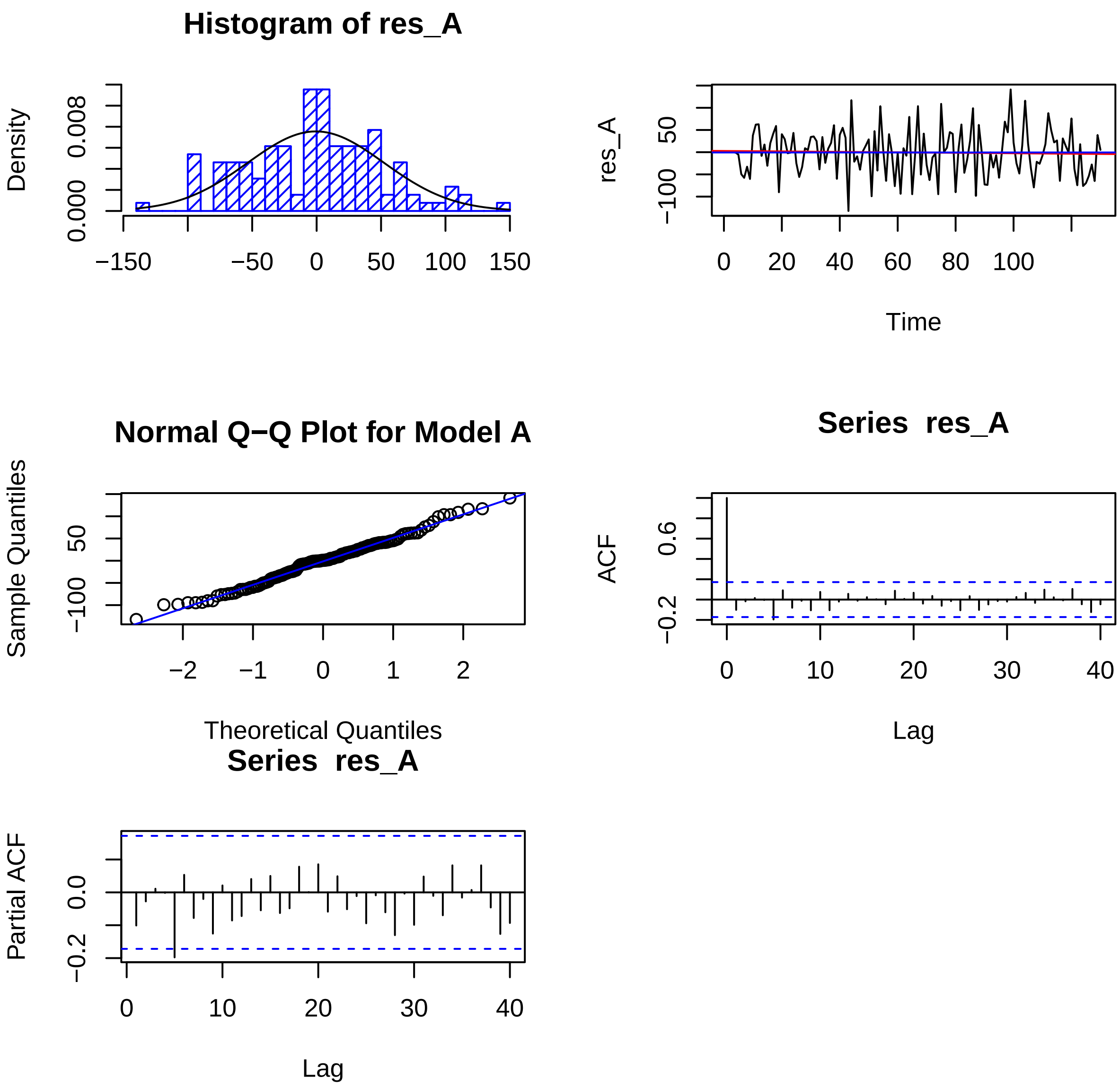
$$(1+0.20374B+0.004944B^2+0.07521B^3-0.41229B^4)Y_t = (1-0.90965B^4)Z_t, \text{ for } Y_t := (1-B)^1\left(1-B^4\right)^1 X_t$$

The model is invertible because Θ is smaller than one, thus the roots of MA part lie outside of the unit circle. The model is stationary because all the roots of ϕ polynomial lie outside of unit circle regardless of their imaginary parts.

Hence, both models are good for diagnostic checking

Diagnostic checking

Model A




```
##
##  Shapiro-Wilk normality test
```

```
##
## data:  res_A
## W = 0.99091, p-value = 0.5592
```

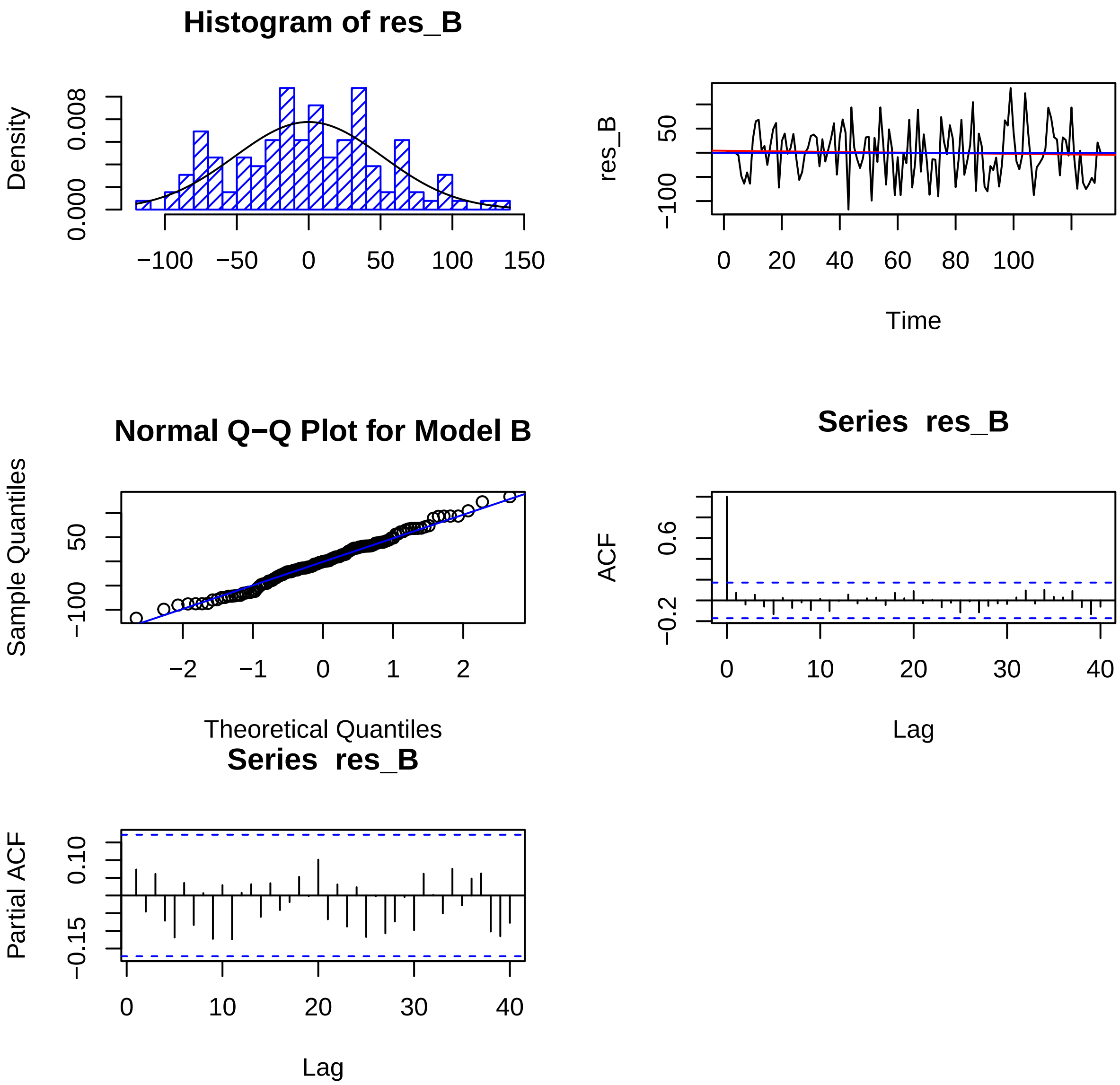
```
##
##  Box-Pierce test
##
## data:  res_A
## X-squared = 11.903, df = 9, p-value = 0.2189
```

```
##
##  Box-Ljung test
##
## data:  res_A
## X-squared = 12.707, df = 9, p-value = 0.1763
```

```
##
##  Box-Ljung test
##
## data:  (res_A)^2
## X-squared = 9.3614, df = 11, p-value = 0.5886
```

Note that for residuals' dependence tests, fitdf=2 and lag=11 because the size of data is 130. Model A passes the linear and nonlinear dependence tests with alpha greater than 0.05. It also passes the normality test(visual test through plot and shapiro-wilk normality test). However, most importantly, its residuals cross the confidence interval at lag 4 for both acf and pacf. Thus, furthuer improvement is needed for Model A if it is used practically.

Diagnostic checking for model B




```
##
##  Shapiro-Wilk normality test
##
## data:  res_B
## W = 0.99042, p-value = 0.5119

##
##  Box-Pierce test
##
## data:  res_B
## X-squared = 7.3695, df = 6, p-value = 0.288

##
```

```
##  Box-Ljung test
##
## data:  res_B
## X-squared = 7.8709, df = 6, p-value = 0.2477
```

```
##
##  Box-Ljung test
##
## data:  (res_B)^2
## X-squared = 13.766, df = 11, p-value = 0.2462
```

Note that for residuals' dependence tests, fitdf=5 and lag=11 because the size of data is 130. Model B passes the linear and nonlinear dependence tests with alpha greater than 0.05. It also passes the normality test(visual test through plot and shapiro-wilk normality test). The residuals' acf and pacf also stays inside the confidence interval

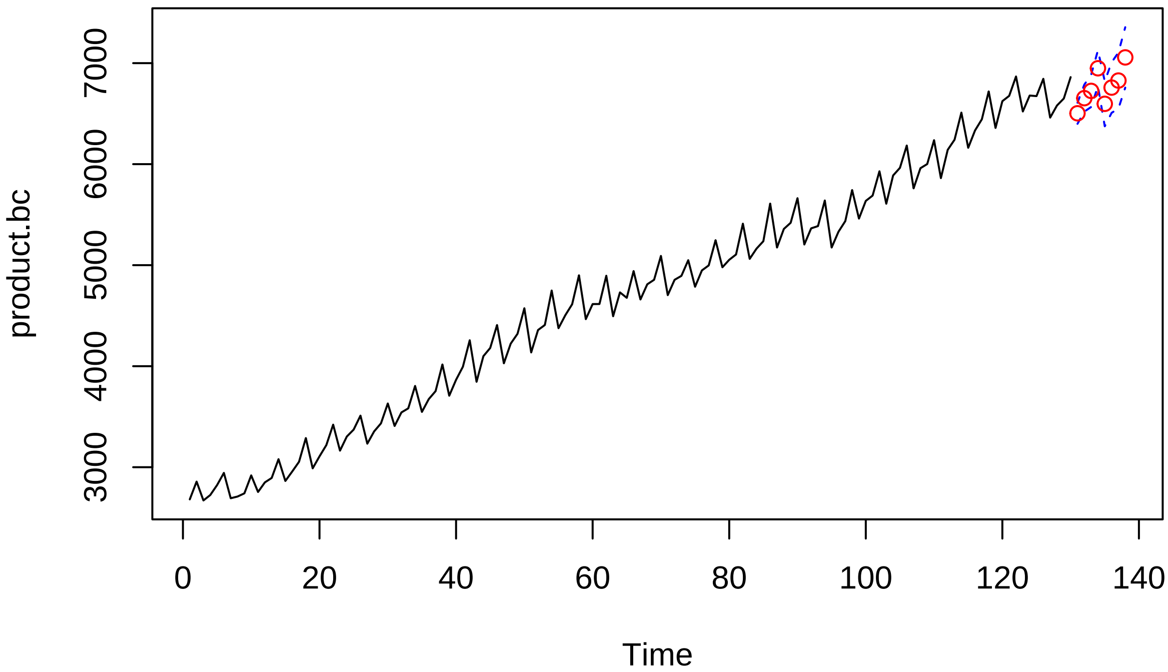
Conclusion: We could have made changes to Model A, but the best option is forecasting with Model B so we do not have to worry about the residuals.

Forecasts 8 steps ahead

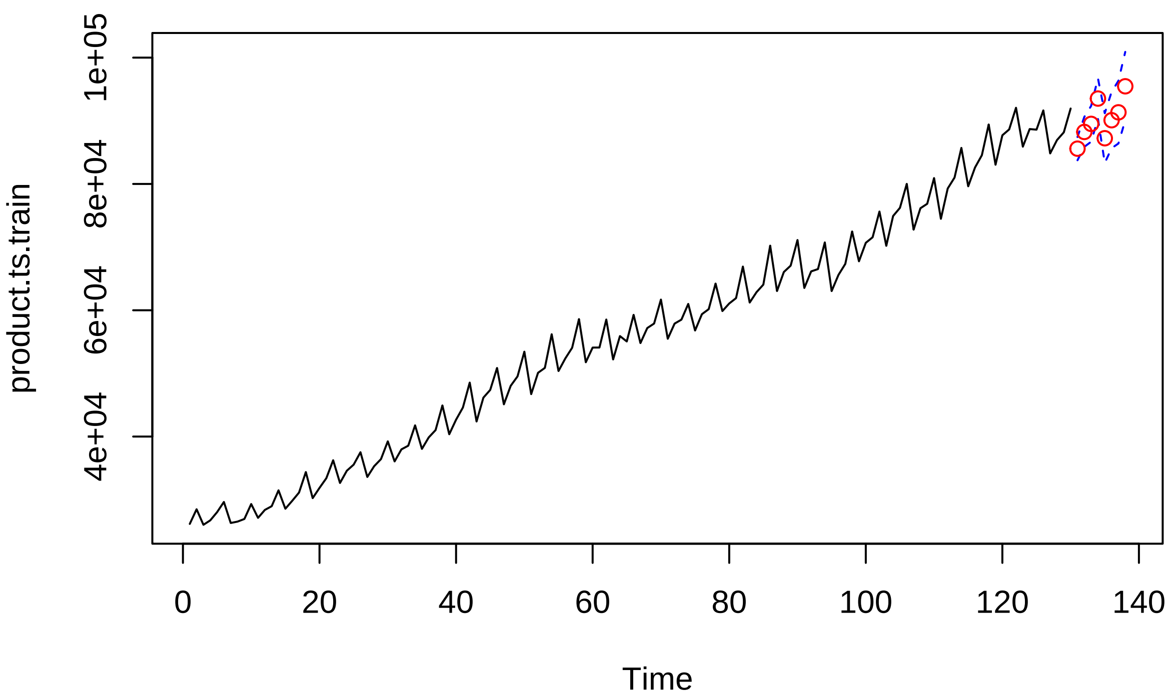
Now, after making sure our model is suitable for forecasting. We can put the model into practice. Note that we are forecasting 8 stepts ahead, and in this case, it will be 8 quarters ahead.

```
##     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 131       6503.548 6436.619 6570.478 6401.189 6605.908
## 132       6653.369 6567.810 6738.928 6522.518 6784.220
## 133       6725.186 6623.068 6827.304 6569.010 6881.362
## 134       6949.255 6835.426 7063.084 6775.168 7123.341
## 135       6597.242 6454.536 6739.948 6378.992 6815.492
## 136       6757.967 6597.669 6918.265 6512.813 7003.121
## 137       6827.600 6649.806 7005.395 6555.687 7099.513
## 138       7056.790 6865.254 7248.327 6763.861 7349.720
```

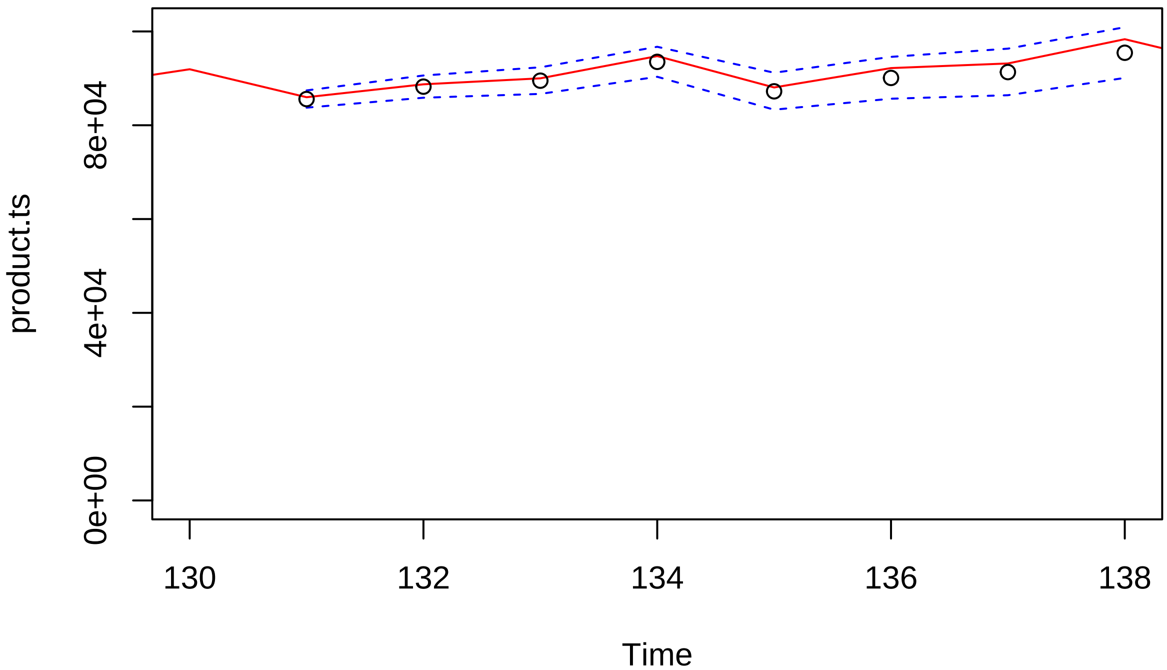
Let us firstly forecast on boxcox transformed data to see how it performs.Check to see if it falls in side $\widehat{Y}_n$ (h) $\pm$ error bound the prediction interval



It does well! Now, we could forecast on original data by taking the boxcox inverse of the transformed data's model(there is a function called inv_boxcox that could inverse the boxcox transformation).



To take a closer look at the prediction by zooming in



The model's prediction on original data also falls inside the prediction interval. Hence, our model performs well on the Quarterly Australian Gross non-farm product dataset.

Appendix

necessary packages

```
library(tsdl,quietly = T)
library(astsa,quietly = T)
library(MASS,quietly = T)
library(forecast,quietly = T)
library(ggplot2,quietly = T)
library(ggfortify,quietly = T)
library(UnitCircle)
library(ldsr)
```

Data Importation and First Step Analysis

```
tsdl[132]
length(tsdl[[132]])
attr(tsdl[[132]], "subject")
attr(tsdl[[132]], "source")
attr(tsdl[[132]], "description")
tsdl[[132]]
```

```r
product<-tsdl[[132]]
product.ts<-ts(product,frequency = 1)
ts.plot(product.ts,main="quarterly product production")
```

```r
product.ts.train<-product.ts[1:130] # divide the datasets into training set and testing
product.ts.test<-product.ts[130:140]
acf(product.ts.train,lag.max=50,xlab="lag(quarterly)")
pacf(product.ts.train,lag.max=50,xlab="lag(quarterly)")
```

Data Transformation

Boxcox Transformation and log transformation

```r
t<-time(product.ts.train)
bcTransform <- boxcox(product.ts.train~ as.numeric(1:length(product.ts.train)))
```

```r
lambda=bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
product.bc = (1/lambda)*(product.ts.train^lambda-1)
plot.ts(product.bc,main="boxcox transformation")
product.log <- log(product.ts.train)
plot.ts(product.log,main="log transformation")
```

Differencing

```r
product_transformed_df1<-diff(product.bc,4)
variance.orginal<-var(product.bc)
variance.lag4<-var(product_transformed_df1) #variance of differecing at lag 4
plot.ts(product_transformed_df1, main="Dataset differenced at lag 4")
print(paste("the variance of orginal dataset is: ",variance.orginal))
print(paste("the variance of differencing at lag4 dataset is: ",variance.lag4))
```

```r
product_transformed_df2<-diff(product_transformed_df1,1)
variance.lag4.lag1<-var(product_transformed_df2) #variance of differecing at lag 4 and
plot.ts(product_transformed_df2, main="Dataset differenced at lag 4 and lag1")
fit <- lm(product_transformed_df2 ~ as.numeric(1:length(product_transformed_df2)))
abline(fit, col="red")
abline(h=mean(product_transformed_df2), col="blue")
print(paste("the variance of differencing at lag4 and lag1 dataset is: ",variance.lag4.l
```

```r
product_transformed_df3<-diff(product_transformed_df2,1)
variance.lag4.lag1.lag1<-var(product_transformed_df3) #variance of differecing at lag 4
plot.ts(product_transformed_df3, main="Dataset differenced at lag 4,lag1,and lag1")
fit <- lm(product_transformed_df3 ~ as.numeric(1:length(product_transformed_df3)))
abline(fit, col="red")
```

```
abline(h=mean(product_transformed_df3), col="blue")
print(paste("the variance of differencing at lag4 ,lag1 and lag1 dataset is: ",variance.
```

```
hist(product_transformed_df2, density=20,breaks=30, col="blue", xlab="", prob=TRUE)
 m<-mean(product_transformed_df2)
std<- sqrt(var(product_transformed_df2))
curve(dnorm(x,m,std), add=TRUE )
```

Model Construction

ACF and PACF analysis

```
par(mfrow=c(2,1))
acf(product_transformed_df2,4*12)
pacf(product_transformed_df2,4*12)
```

AICcs,Invertibility and Stationarity

```
aicc<-matrix(0,nrow = 1,ncol=4)
colnames(aicc)<-c("only Q=1","p=4,q=4","p=4,Q=1","P=1,Q=1")
fit<- sarima(xdata=product.bc,details=F,p=0,d=1,q=0,P=0,D=1,Q=1,S=4)
aicc[1,1]<-fit$AICc
fit<- sarima(xdata=product.bc,details=F,p=4,d=1,q=4,P=0,D=1,Q=0,S=4)
aicc[1,2]<-fit$AICc
fit<- sarima(xdata=product.bc,details=F,p=4,d=1,q=0,P=0,D=1,Q=1,S=4)
aicc[1,3]<-fit$AICc
fit<- sarima(xdata=product.bc,details=F,p=0,d=1,q=0,P=1,D=1,Q=1,S=4)
aicc[1,4]<-fit$AICc
aicc
```

```
fit.1<-sarima(xdata=product.bc,details=F,p=0,d=1,q=0,P=1,D=1,Q=1,S=4)
cat("Coefficients");fit.1$fit$coef
```

```
fit.2<-sarima(xdata = product.bc,details=F,p=4,d=1,q=0,P=0,D=1,Q=1,S=4)
cat("Coefficients");fit.2$fit$coef
polyroot(c(1,0.2037487,0.004944228,0.075215667,0.412296181))
uc.check(pol_ = c(1,0.2037487,0.004944228,0.075215667,-0.412296181), plot_output = TRUE)
```

Diagnostic checking

Model A

```
par(mfrow=c(2,2))
res_A<-fit.1$fit$residuals
```

```
hist(res_A,density=20,breaks=20, col="blue", xlab="", prob=TRUE)
m <- mean(res_A)
std <- sqrt(var(res_A))
curve( dnorm(x,m,std), add=TRUE )
plot.ts(res_A)
fitt <- lm(res_A ~ as.numeric(1:length(res_A))); abline(fitt, 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")
acf(res_A, lag.max=40)
pacf(res_A, lag.max=40)
```

```
shapiro.test(res_A)
Box.test(res_A, lag = 11, type = c("Box-Pierce"), fitdf = 2)
Box.test(res_A, lag = 11, type = c("Ljung-Box"), fitdf = 2)
Box.test((res_A)^2, lag = 11, type = c("Ljung-Box"), fitdf = 0)
```

Diagnostic checking for model B

```
par(mfrow=c(2,2))
res_B<-fit.2$fit$residuals
 hist(res_B,density=20,breaks=20, col="blue", xlab="", prob=TRUE)
 m <- mean(res_B)
std <- sqrt(var(res_B))
curve( dnorm(x,m,std), add=TRUE )
plot.ts(res_B)
fitt <- lm(res_B ~ as.numeric(1:length(res_B))); abline(fitt, 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")
acf(res_B, lag.max=40)
pacf(res_B, lag.max=40)
```

```
shapiro.test(res_B)
Box.test(res_B, lag = 11, type = c("Box-Pierce"), fitdf = 5) #lag is the square root of
Box.test(res_B, lag = 11, type = c("Ljung-Box"), fitdf = 5)
Box.test((res_B)^2, lag = 11, type = c("Ljung-Box"), fitdf = 0)
```

Forecasts

Transformed datasets

```
fit.B <- arima(product.bc,order=c(4,1,0), seasonal = list(order = c(0,1,1), period = 4),
fixed = NULL, method="ML")
```

```r
forecast(fit.B)
#prints forecasts with prediction bounds in a table
```

```r
pred.tr <- predict(fit.B, n.ahead = 8)
U.tr= pred.tr$pred + 2*pred.tr$se #the upperbound of forecasts
L.tr= pred.tr$pred - 2*pred.tr$se #the lowerbound of forecasts
ts.plot(product.bc, xlim=c(1,length(product.bc)+8), ylim = c(min(product.bc),max(U.tr)))
lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")
points((length(product.bc)+1):(length(product.bc)+8), pred.tr$pred, col="red")
```

Original datasets

```r
pred.orig <- inv_boxcox(pred.tr$pred,lambda=lambda)# use inverse boxcox function to cha
U= inv_boxcox(U.tr,lambda=lambda)
L= inv_boxcox(L.tr,lambda=lambda)
ts.plot(product.ts.train, xlim=c(1,length(product.ts.train)+8), ylim = c(min(product.ts.
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(product.ts.train)+1):(length(product.ts.train)+8), pred.orig, col="red")
```

```r
ts.plot(product.ts, xlim = c(130,length(product.ts.train)+8), ylim = c(0,max(U)), col="r
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(product.ts.train)+1):(length(product.ts.train)+8), pred.orig, col="black"
```