Table of Content

Abstract

In the international trade, the import or export is a very important part between countries. The objective of this project report is to predict monthly Australian imports from Japan using time series analysis of the data from Jul. 65 to Oct. 93. We build SARIMA model with lowest AICc and BIC based on the stationary data (The training dataset is from Jul. 65 to Dec. 92). In order to find out the most proper model, we use model selection methods and model diagnostics, such as normality checking, independence checking and constant variance checking. Then we use the most proper model to predict the data from Jan. 93 to Oct. 93 and then compare with the real data, which shows that our result is good enough.

PART 1. Introduction

Import is a significant part in the international trade, which is related to economic scenario among countries. In this report, monthly periodic data is collected to study Australian imports from Japan from Jul. 65 to Oct. 93. We use R software to build time series model (SARIMA model) to predict Australian imports from Japan patterns and behavior.

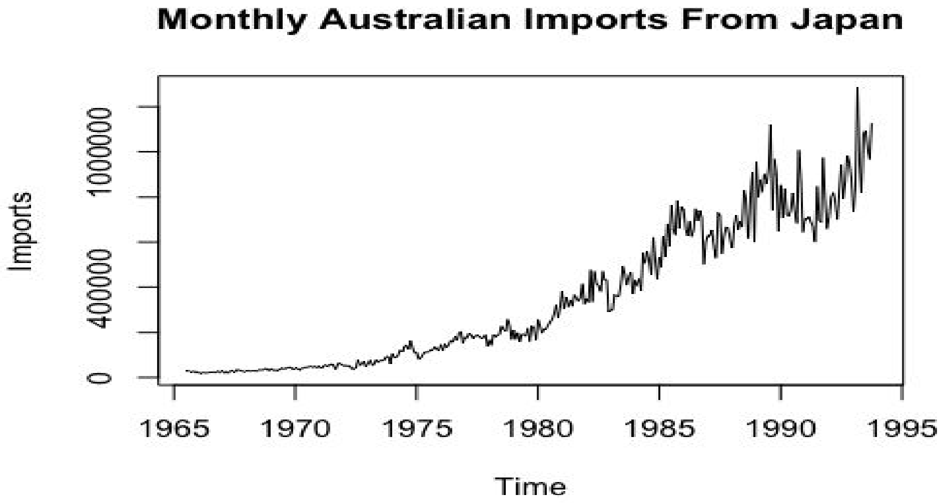
By performing exploratory analysis, we find out the variance is non-stationary and thus BOX-COX transformation should be applied to the original data; then we difference at lag 12 and lag 1 to remove seasonality and trend. By looking at the ACF and PACF plots of stationary series, we adopt the AIC and BIC model selection criteria. Lastly, by comparing their numbers of parameters and diagnostics plots, we decide the final model to be $SARIMA(2,1,0)*(2,1,3)_{12}$ Then we use the final model to predict ten month imports data and compare with the real data, finding that the result is similar and concluding that our model is effective.

PART 2. Data Exploratory Analysis

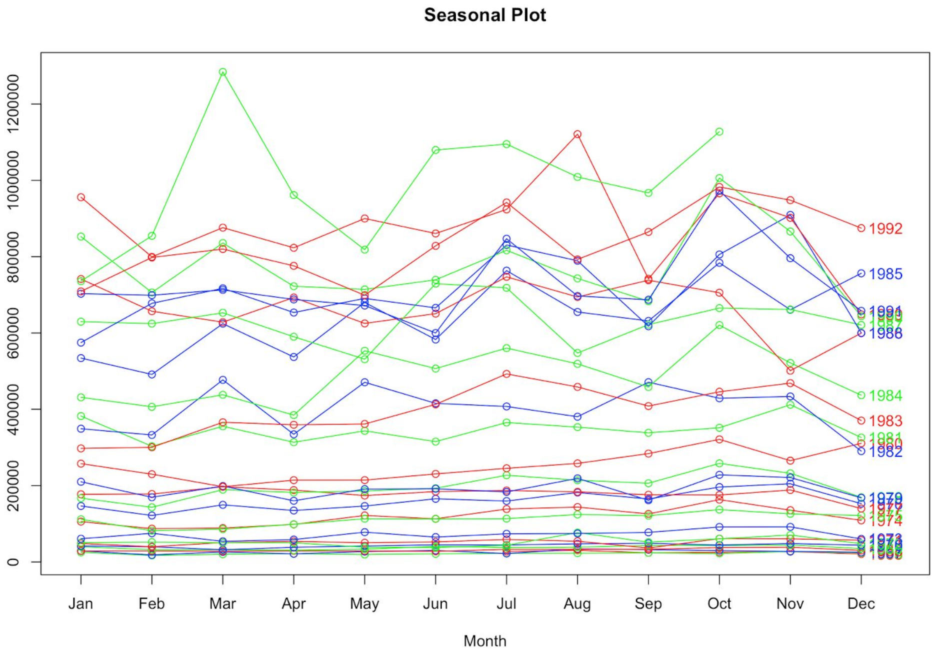
2.1 Preliminary Data Exploration

This time series data has 342 sets of observation points for monthly Australian import from Japan. For this Times Series, two variables relevant to the data can be taken into account, namely date on a monthly basis and Monthly Australian Imports From Japan in thousands of dollars.

As the preliminary data exploration, we save the Monthly imports data in time series form and plot it with the 342 observation results as mentioned below:



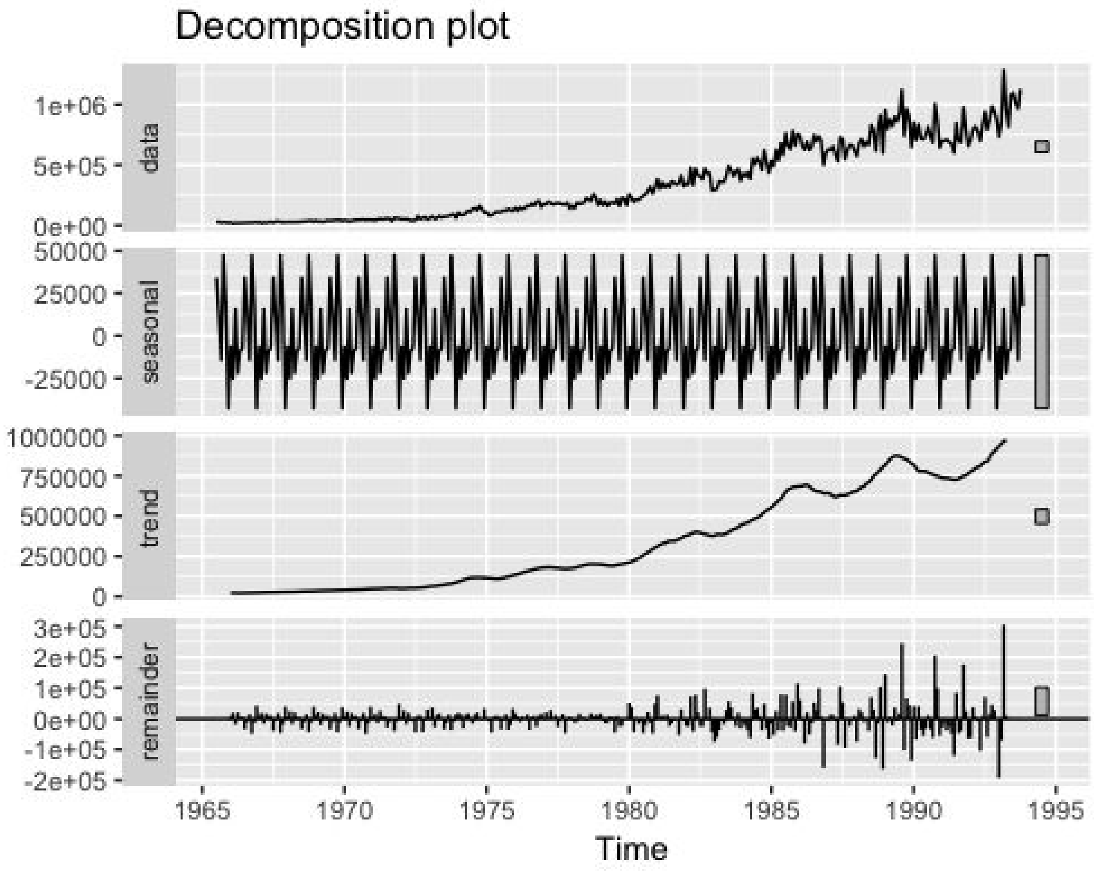
It’s reflected in the plot that the overall condition refers to an upward trend. From the above plot, we are not sure if seasonality exists. Under the purpose of clearly identifying the seasonal features of this data, we indicate the condition through a seasonal plot drawn as below:



The Seasonal plot shows that the import from Japan is increasing every year. There are two peak months, July and Oct. The shape of each year import are similar. Thus we conclude that this time series has seasonal effect.

2.2 Decomposition Model

Under the purpose of proving the existence of the trend and its seasonality, the data is decomposed into Y_t used to achieve a classical decomposition model, namely $Y_t = m_t + s_t + S_t$, where mt refers to the trend component while s_t indicates the seasonal component and S_t is defined as a stationary process. A decomposition plot can be drawn as following:



According to the plot, it can be observed that seasonal feature comes into existence with three positive spikes and three negative pikes on annual basis. Moreover, there is an upward trend takes place. Consequently, as the data non-stationary, further data transforming and differentiation need to be conducted so as to make the data stationary. Once after the data becomes stationary, it's time to establish and select the model.

PART 3. Data Transformation

In order to make our data stay stationary and prepare for model forecasting, we need to transform and difference our data. First, we need to stabilize the variance of our data by using Box-Cox transformation. And then, we remove the trend and seasonality by differencing the data.

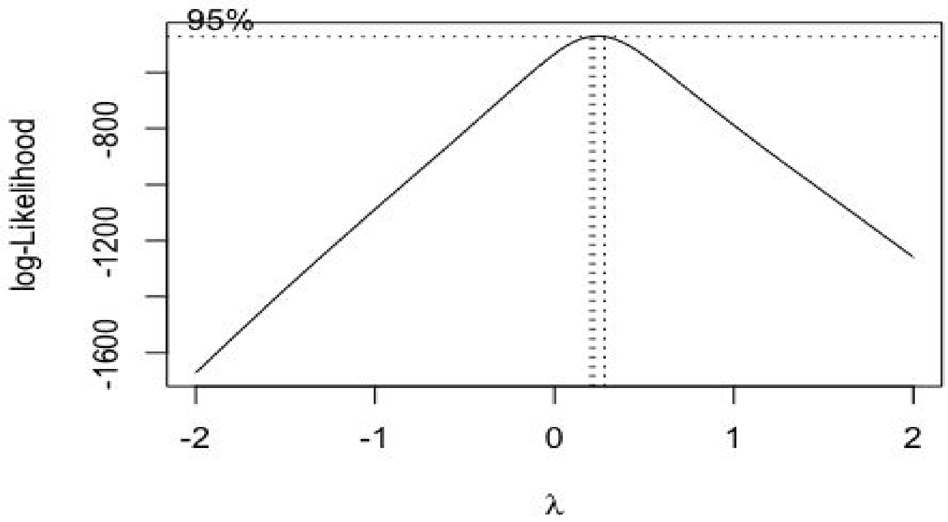
3.1 Stabilize Variance of the Import data

The Box-Cox transformation can help us to get rid of non-constant variance of the data. By using the Box-Cox transformation, we can obtain the “lamda” for transformation, which is 0.2222.

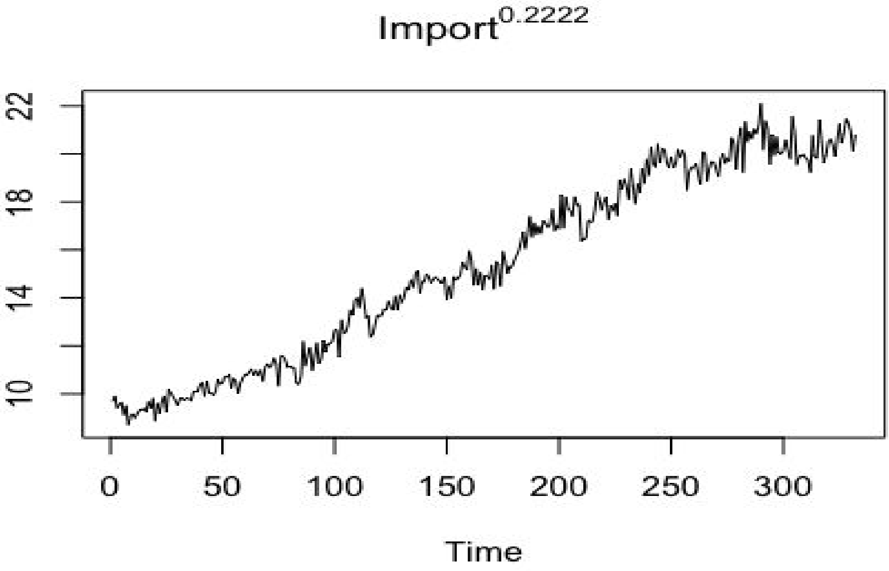
$$V_t = Y_t^{0.2222}$$

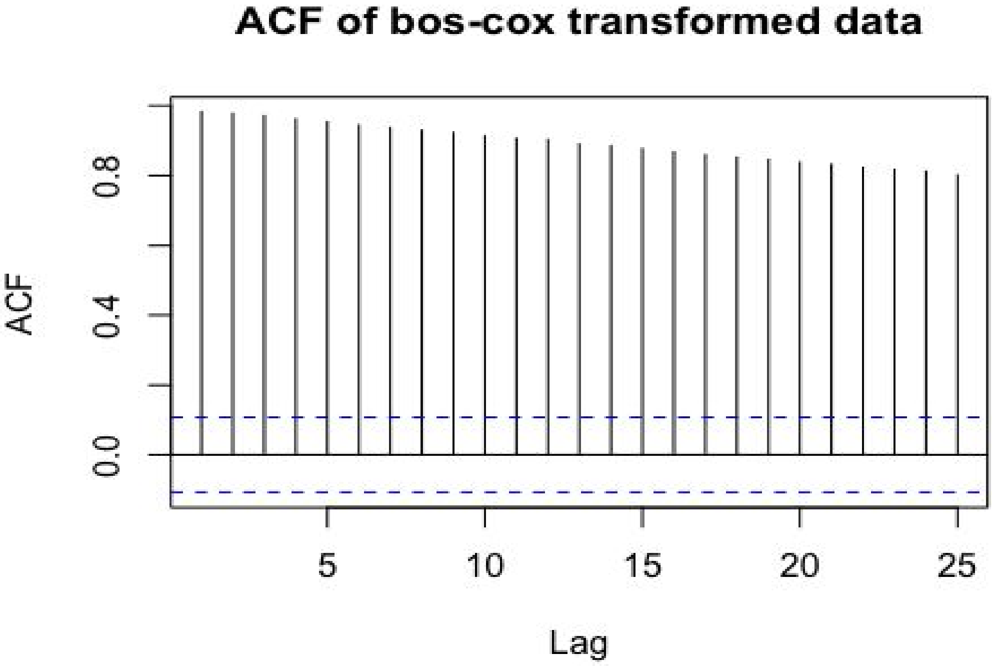
```
lamda <- bxTransform$x[which.max(bxTransform$y) ]
lamda # lamda=0.2222222

## [1] 0.2222222
```



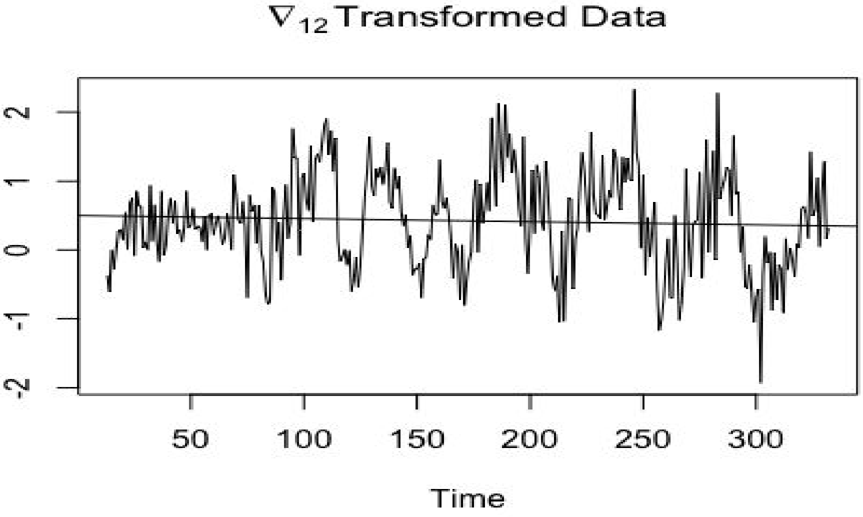
After transforming our data, we make a plot and an ACF plot of our data. By observing the plots, we can see the non-constant variance problem is fixed but we have a rising trend and seasonality within our data, meaning that we need to differencing our data to remove the seasonality and trend.



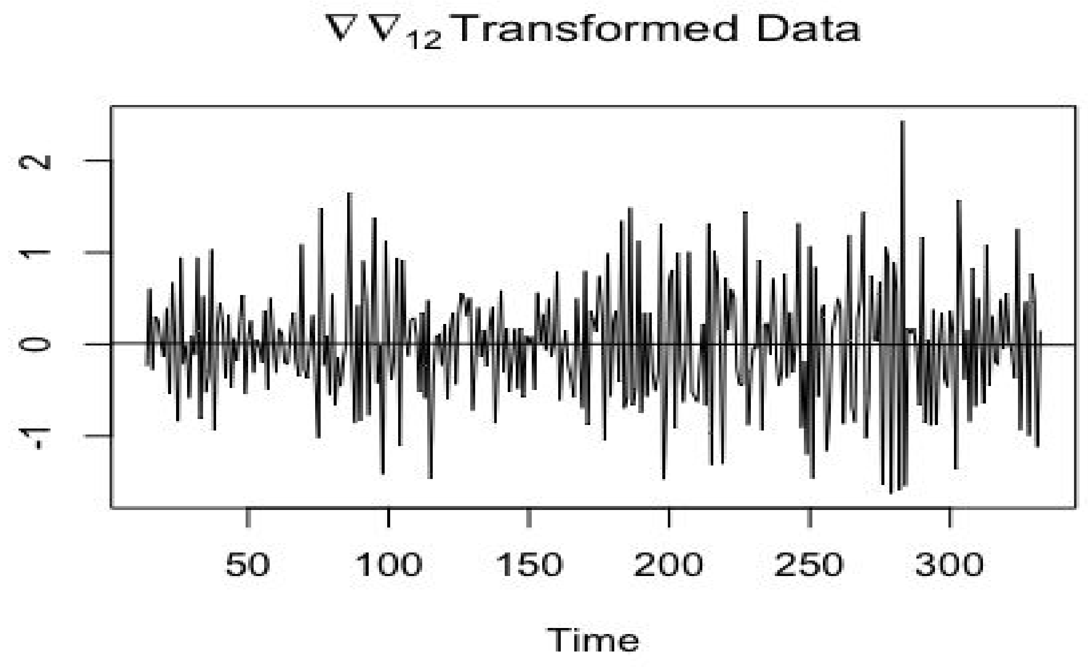


3.2 Remove Seasonality and Trend

In order to remove the seasonality, we assign the period of 12 towards the seasonal component, which is observed from decomposition part. After we do that, we find the the trend line is slightly downward with the variance of the data is 0.4845616.



Then, we need to remove the trend based on the deseasonalized data by differencing the data once more at lag 1 and plot the data again. In this part, we find the the trend line is horizontal with the variance of the data is 0.4377893. At this point, if we difference the data again at lag 1, the variance of the data will be 1.279259, which is larger than 0.4377893, meaning that it is over-differencing. Therefore, we only need to difference the data once.



In order to find out if the model is stationary or not, we use the Augmented Dickey-Fuller Test. The Null hypothesis of the test is that is non-stationary and the alternative hypothesis is is stationary. From the code below, we can see the p-value of the test is 0.01, which means we can reject the null hypothesis and conclude that is stationary within 95% confidence level.

```
# D-F test
adf.test(diff12diff1)

## Warning in adf.test(diff12diff1): p-value smaller than printed p-value

## 
##  Augmented Dickey-Fuller Test
## 
## data:  diff12diff1
## Dickey-Fuller = -7.3355, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

PART 4. Model Identification and Estimation

Since we have the seasonal data, which contains both seasonal and non-seasonal parts, so we would fit the SARIMA model as below:

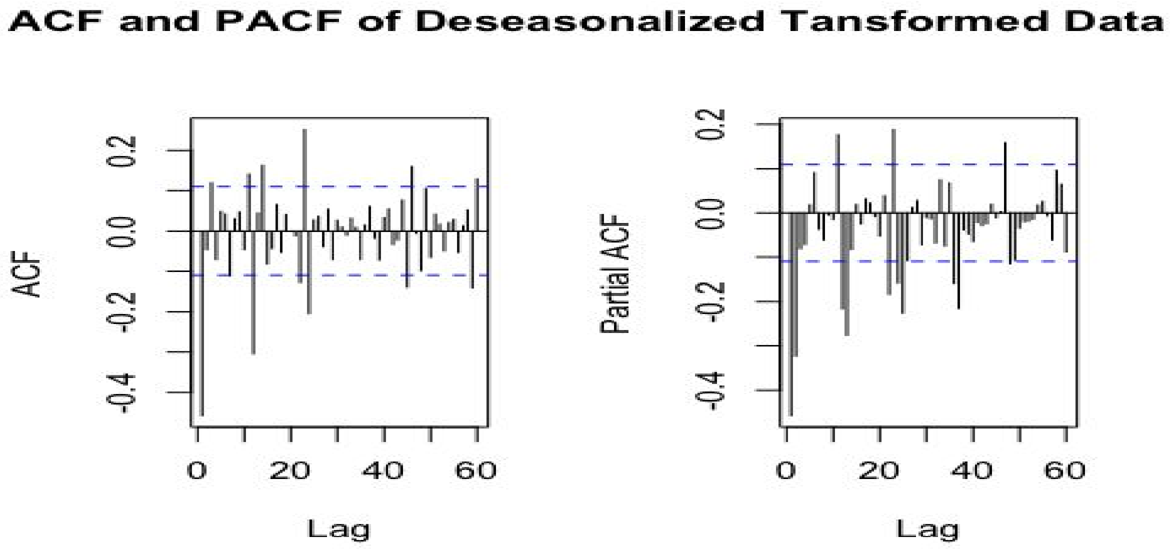
$$SARIMA(p, d, q) * (P, D, Q)_S$$

where p=the order of non-seasonal AR process, d=non-seasonal differencing, q= the order of non- seasonal MA process, P= the order of seasonal AR process, D=seasonal differencing, Q = the order of seasonal AR process and s=the period of the time lag.

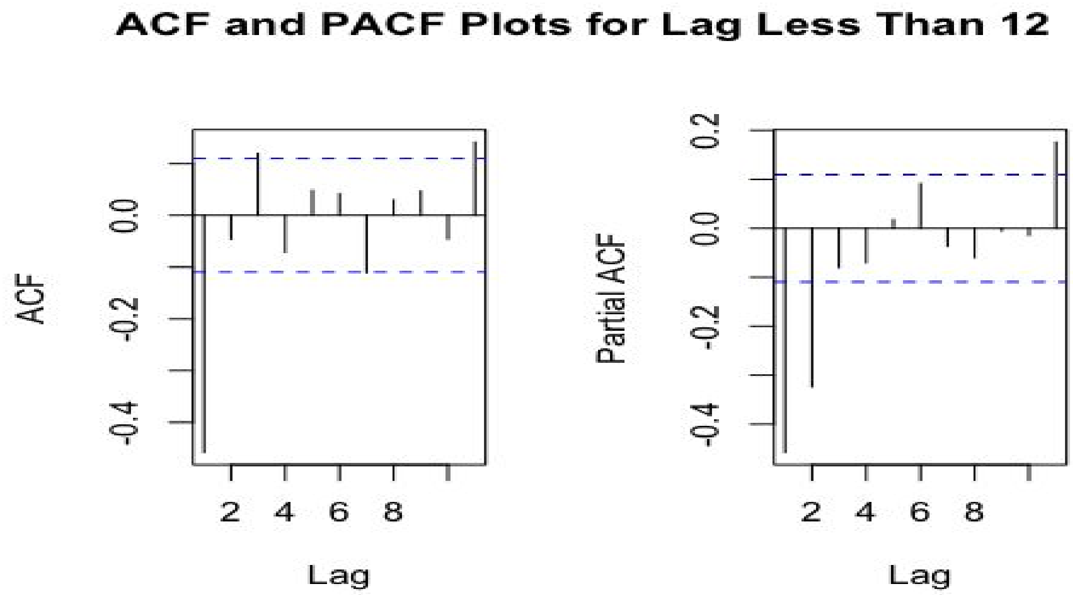
The data of international trade is a monthly data, thus we set S to be 12. Since we difference the data at log 12 in order to remove seasonality and at log 1 in order to remove trend in the previous steps, then we have d=1 and D=1.

Our next step is to find some possible p, q and P, Q values through the ACF and PACF plots in the preliminary model identification. Then, we want to find the best fitting value using information criteria (AICc and BIC).

4.1 Preliminary Model Identification



As for the seasonal term P, Q, we plot the ACF and PACF of stationary time series Xt first. From the ACF and PACF plots, ACF plot cut off after lag 36, so we have Q=3; and PACF plot cut off after lag 24, so we have P=2.



Then, we find p, q by looking at ACF and PACF plots of the zoomed plot. Therefore, for the non-seasonal terms p and q, ACF cuts off after lag=1 or tails off, thus, q=1 or q=0. As well as, PACF cut off after lag=2 or tails off, thus p=2 or p=0.

Therefore, we consider 6 models as our candidate models with combination of p = {0, 1, 2} and q = {0, 1}:

Model1: SARIMA(0,1,0)×(2,1,3)12 Model2: SARIMA(0,1,1)×(2,1,3)12

Model3: SARIMA(1,1,0)×(2,1,3)12 Model4: SARIMA(1,1,1)×(2,1,3)12

Model5: SARIMA(2,1,0)×(2,1,3)12 Model6: SARIMA(2,1,1)×(2,1,3)12

4.2 Model Selection

We use AICc and BIC to select the best model or models. We choose the minimum score for the best model for both AICc and BIC, which is -0. 6706968 and -1. 582391. Then, we choose two possible models:

$SARIMA(2,1,1) * (2,1,3)_{12}$ and $SARIMA(2,1,0) * (2,1,3)_{12}$.

```
AICc

##          q=0        q=1
## p=0 -0.2851886 -0.6625595
## p=1 -0.6356582 -0.6348451
## p=2 -0.6706968 -0.6663736
```

BIC

```
##           q=0       q=1
## p=0 -1.234685 -1.600857
## p=1 -1.573956 -1.561983
## p=2 -1.597835 -1.582391
```

4.3 Model Estimation

For the two models we selected above, we fit and estimate the coefficients based on MLE method. The results for both two models are showed in the below:

(1) For model-1: $SARIMA(2, 1, 1) * (2, 1, 3)_{12}$

```
## Call:
## arima(x = tryimport, order = c(2, 1, 1), seasonal = list(order = c(2, 1, 3),
##     period = 12), method = "ML")
##
## Coefficients:
##           ar1      ar2      ma1    sar1     sar2     sma1    sma2     sma3
##       -0.3670  -0.2028  -0.3150  0.6640  -0.4788  -1.4652  0.8109  -0.2213
## s.e.   0.1564   0.0989   0.1547  0.2224   0.3916   0.2376  0.4563   0.3865
##
## sigma^2 estimated as 0.1787:  log likelihood = -187.39,  aic = 390.79
```

(2) For model-2: $SARIMA(2, 1, 0) * (2, 1, 3)_{12}$

```
## Call:
## arima(x = tryimport, order = c(2, 1, 0), seasonal = list(order = c(2, 1, 3),
##     period = 12), method = "ML")
##
## Coefficients:
##           ar1      ar2    sar1     sar2     sma1    sma2     sma3
##      -0.6489  -0.3377  0.7358  -0.4314  -1.5381  0.8326  -0.1743
## s.e.   0.0544   0.0538  0.2271   0.2630   0.2410  0.4119   0.2482
##
## sigma^2 estimated as 0.1809:  log likelihood = -189.04,  aic = 392.08
```

PART 5. Diagnostics

Now, we have identified two models and estimated their parameters respectively. Hence, we now need to check normality, independence and constant variance of errors to make sure our assumption is valid since these three diagnostic checks are very important before we do the forecasting.

5.1 Normality Checking

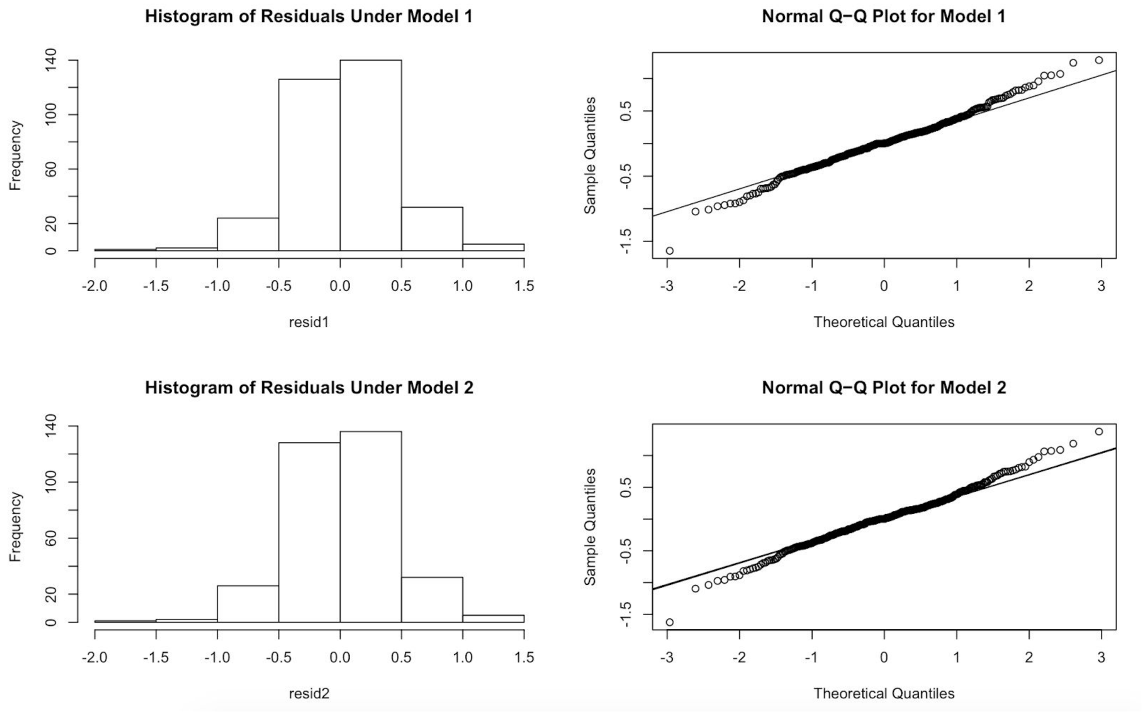
To check the normality, we need use the histogram and Q-Q plot. After we see the graph below, we can see that the histogram of residuals are nearly symmetrical. We can see that it is normal distribution.

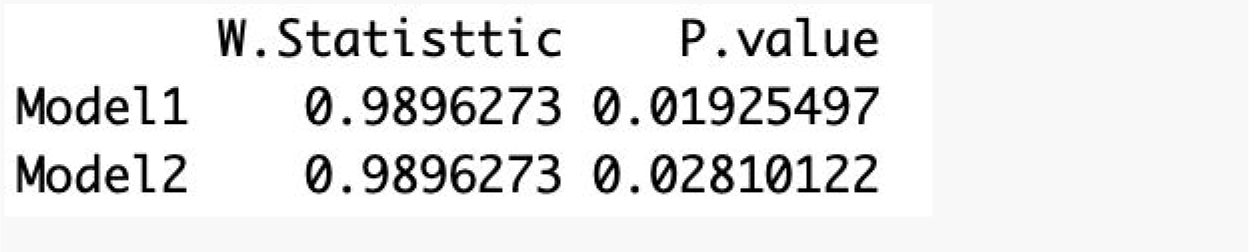
And from the qq-plot, we can see that nearly all points are lying on a straight line, that means the error is normal.

We also use the Shapiro Wilk Test since the Shapiro Wilk Test for normality is one for three general normality tests designed to detect all departures from normality.

The test rejects the hypothesis of normality when the p-value is greater than or equal to 0.01 (Sometimes residuals have other hidden component not only White Noise). Passing the normality test allows you to state with 99% confidence the data does fit the normal distribution at 99% confidence level. Passing the normality test only allows you to state no significant departure from normality was found.

We could find the model 2 is better than model 1 in Normality Checking.





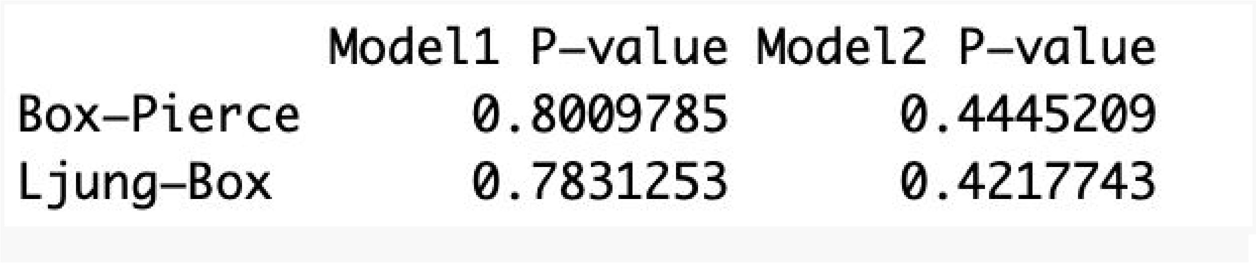
```
       W.Statisttic    P.value
Model1    0.9896273 0.01925497
Model2    0.9896273 0.02810122
```

5.2 Independence Checking

Then we need to check the independence. Errors are supposed to be uncorrelated for a time series model, so we perform two tests: the Ljung-Box Test and the Box-Pierce Test. The result shows below, say that the p-value > 0.05 in both test for both model, hence we do not reject the assumption of un-correlation between the residuals. That is we have the independent residuals.

H0: Residuals are independent

Ha: Residuals are correlated α = 0.05

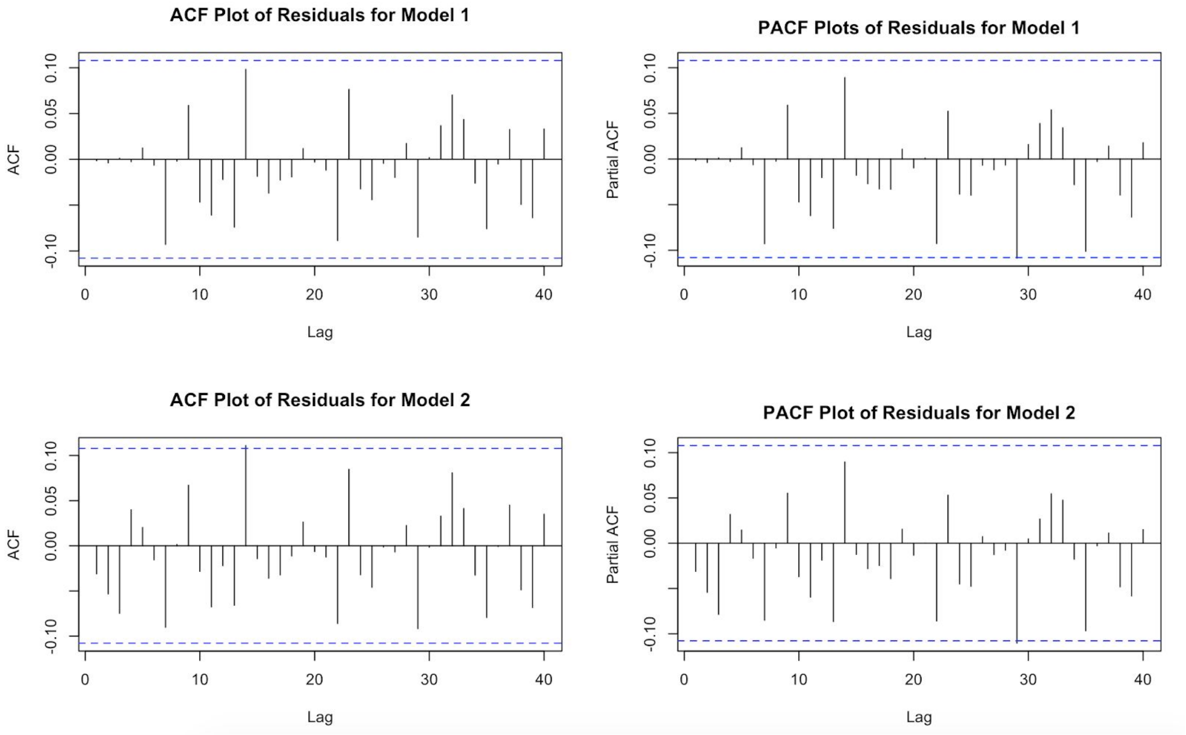


```
            Model1 P-value Model2 P-value
Box-Pierce       0.8009785      0.4445209
Ljung-Box        0.7831253      0.4217743
```

5.3 Constant Variance Checking

For the final model, we need to make sure our residuals have the constant variance. We can check the violation of constant variance of errors by the analysis of the ACF and PACF plots of the squared residuals in our model. The ACF and PACF should lie within 95% of White Noise limits.

Thus, we choose the model 2 as our final model.



ACF Plot of Residuals for Model 1

PACF Plots of Residuals for Model 1

ACF Plot of Residuals for Model 2

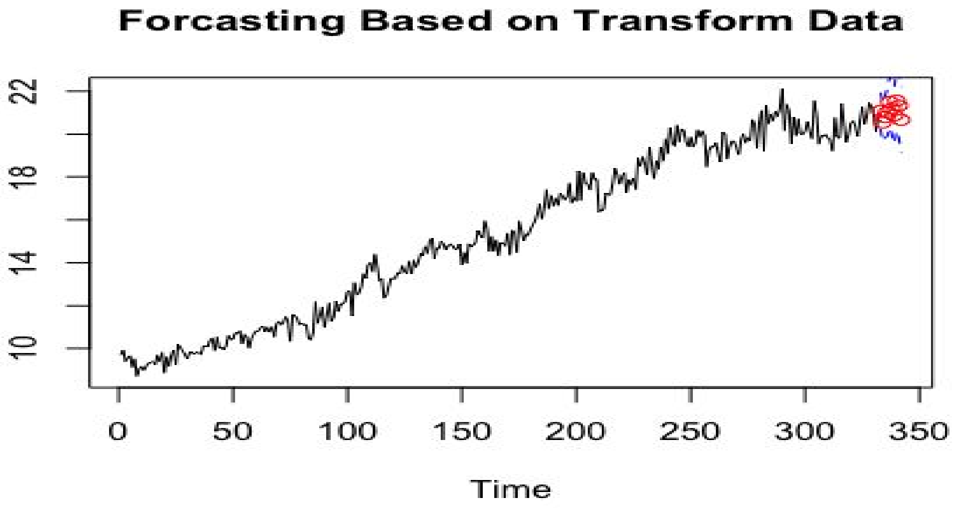
PACF Plot of Residuals for Model 2

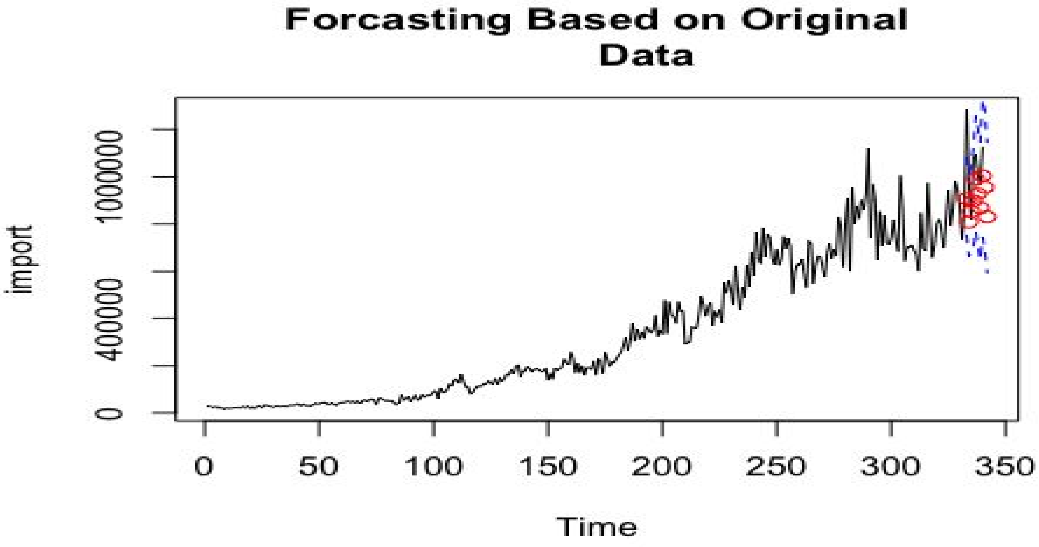
PART 6. Forecasting

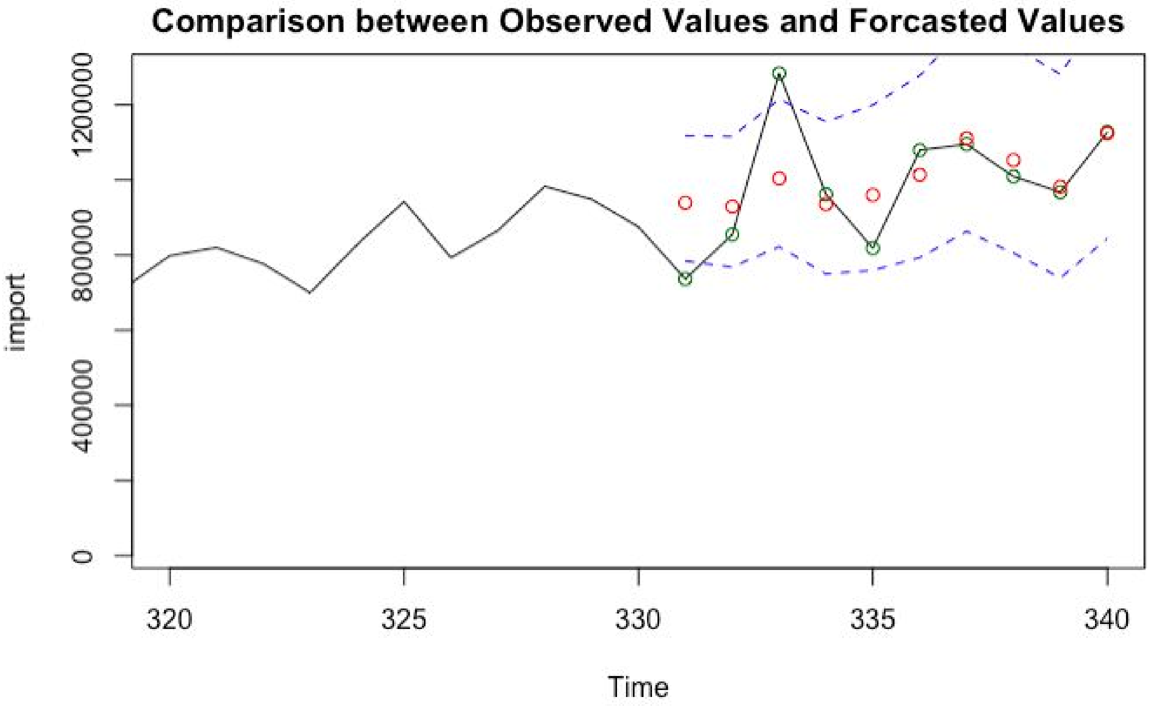
Based on the Model selection and Diagnostics result, we choose our final model : $SARIMA(2,1,0)*(2,1,3)_{12}$.

We now forecast 10 values ahead—import data from Jan. 93 until Oct. 93 at monthly intervals.

The First graph refers to the transformed data while the Second graph shows the original data. The 10 red dots represent the 10 forecasted values,where the blue lines represent the boundaries of confidence interval. Taking a serious analysis on these data, the difference between true observed values (green dots) and forecasted value (red dots) can be seen in Third Figure.







From our graph below, we can see that most values are lying within the bound for both model, and for some exceeded values, we can determine they are outlier in our dataset.

Our final model precisely captures the overall trend and seasonality of the data. As shown by the result of comparison , our model is strongly predictive.

PART 7. Conclusion

The goal of this study is to predict the future behavior of the imports from Jul. 65 to Oct. 93 by analyzing the previous monthly imports data and building a reasonable model for it . Along the process, the model-building steps include: data transformation for stable variability and normality, distribution check, model selection and diagnostics, i.e. normality checking, independence checking and constant-variance checking. Based on above analysis, we choose the final model :

$\text{SARIMA}\,(2,1,0) * (2,1,3)_{12}$.

Then, we use the model to predict the data from Jan. 93 to Oct. 93 and compare the result with the real data, which shows that our result is satisfactory. From the plot, it is shown that the predicted values are not only within a 95% confidence interval , but also very close to the actual values. Although there exists some outliers in the true observed values, it does not affect the main trend the data goes on and the forecasted trend goes along with the trend as well. Hence, we are able to predict the desired data based on our model.

Reference:

monthly Australian imports from Japan:
https://datamarket.com/data/set/22qx/monthly-australian-imports-from-japan-thousands-of-dollars-jul-65-oct-93#!ds=22qx&display=line

Appendix:

~~~
```{r}
library ( astsa )
library( tseries )
library (MASS)
library( forecast )
library (TSA)
library ( ggplot2 )
import0 <- read.csv("C:\\Users\\zoreee\\Desktop\\monthly-australian-imports-from-.csv")
import0 <-  monthly.australian.imports.from.
import<-ts(import0[,2], start=c(1965,7), frequency=12)
# plot
par(mfrow=c(1,1))
plot(import,xlab="Time",ylab="import(Thousands of dollars)", main="Monthly Australian Imports From Japan")
# seasonal Plot
seasonplot(import,12,col=rainbow(3),year.labels = TRUE, main = "Seasonal Plot")
#decomposition plot
decom<-decompose(import1)
autoplot(decom, main="Decomposition plot")
~~~

~~~
#stablize variance
#using box−cox
bxTransform <- boxcox(import ~ as.numeric(1:length(import)))
lamda <- bxTransform$x[which.max(bxTransform$y) ]
lamda # lamda=0.2222222
Transimport <- import^lamda #model transformation
#make sample set: drop last 10 data for later comparison of forecasting
tryimport <- ts (Transimport[1:(length(import) - 10)])
plot(tryimport, xlab="Time" , ylab="", main=expression(Import^0.2222))
acf(tryimport ,main="ACF of bos-cox transformed data")
#De−seasonalize
diff12 <- diff(tryimport ,lag=12)
plot(diff12 ,xlab="Time", ylab="", main=expression(nabla[12]~"Transformed Data"))
abline(lm(diff12~as.numeric(1:length(diff12))))
var(diff12)

#De−trend
diff12diff1<-diff(diff12,lag=1)
plot(diff12diff1 ,xlab="Time", ylab="", main=expression(nabla~nabla[12]~"Transformed Data"))
abline(lm(diff12diff1~as.numeric(1:length(diff12diff1))))
var(diff12diff1)
diff12diff2 <- diff(diff12diff1,lag=1)
var(diff12diff2)
# D-F test
adf.test(diff12diff1)
```
```{r}
~~~

~~~
#Model identifications
#identify P, Q
op <- par ( mfrow=c ( 1 , 2 ) )
acf(diff12diff1 ,lag.max=60,main="")
pacf ( diff12diff1 , lag.max=60, main="")
title (main = "ACF and PACF of Deseasonalized Tansformed Data", outer = TRUE,
    line = -1)
par(op)
#check value of acf , pacf , at lag=12, 24, 36 ,48...
#ACF cuts off after lag=24, then Q=2
#PACF cuts off after lag=36, then P=3
```
```{r}
#identify p,q
op <- par ( mfrow=c ( 1 , 2 ) )
acf(diff12diff1 ,lag.max=11,main="")
pacf ( diff12diff1 , lag.max=11,main="")
title (main="ACF and PACF Plots for Lag Less Than 12" ,outer=TRUE, line=-1)
par(op)
#at lag 1 ,2 ,3 ,... ,11
#acf cuts off after lag=1 or tails off , q=1 or q=0
#pacf cut off after lag=2 or tails off , p=2 or p=0
#test all combination of p in 0 to 2 and q in 0 to 1
```
```{r}
#Model selection by AICc
AICc<-numeric()
~~~

```
for (p in 0:2) {
  for (q in 0:1){
    AICc<-c(AICc,sarima(tryimport,p,1,q,2,1,3,12,details=FALSE)$AICc)
  }
}
AICc<-matrix(AICc,nrow=3,byrow=TRUE)
rownames(AICc)<-c("p=0", "p=1", "p=2")
colnames(AICc)<-c("q=0", "q=1")
AICc
AICc <- data.frame(AICc)
aicc <- setNames(AICc,c("q=0", "q=1"))
aicc
#smallest
#p=2,q=1 smallest
#p=2,q=0 2nd smallest
#AICc very close

#Model selection by BIC
BIC<-numeric()
for (p in 0:2) {
  for (q in 0:1){
    BIC<-c(BIC,sarima(tryimport,p,1,q,2,1,3,12,details=FALSE)$BIC)
  }
}
BIC<-matrix(BIC,nrow=3,byrow=TRUE)
rownames(BIC)<-c("p=0", "p=1", "p=2")
colnames(BIC)<-c("q=0", "q=1")
```

~~~
BIC
BIC <- data.frame(BIC)
bic <- setNames(BIC,c("q=0", "q=1"))
bic
#smallest
#p=2,q=1 smallest
#p=2,q=0 2nd smallest
#AICc very close
#Based on AICc and BIC, select two models
#Model 1, SARIMA(2 ,1 ,1 ,2 ,1 ,3)12
#Model 2, SARIMA(2 ,1 ,0 ,2 ,1 ,3)12
```
```{r}
#Fit and Estimation based on MLE method
# MODEL 1: p=2 and q =1 < SARIMA(2 ,1 ,1 ,2 ,1 ,3)12 >
fit1 <- arima(tryimport, order=c(2,1,1), seasonal=list(order=c(2,1,3), period=12) ,method="ML")
fit1

# Model 2 : p=2 and q=0 < SARIMA(2 ,1 ,0 ,2 ,1 ,3)12 >
fit2 <- arima(tryimport, order=c(2,1,0), seasonal=list(order=c(2,1,3), period=12) ,method="ML")
fit2

##Normality
resid1<-residuals(fit1) #resid for M1
resid2<-residuals(fit2) #resid for M2
op<-par ( mfrow=c ( 2 , 2 ) )
hist(resid1 ,main="Histogram of Residuals Under Model 1") #not good enough
~~~

~~~
qqnorm( resid1 , main="Normal Q−Q Plot for Model 1")

qqline( resid1 )

hist(resid2 ,main="Histogram of Residuals Under Model 2") #good

qqnorm( resid2 , main="Normal Q−Q Plot for Model 2")

qqline( resid2 )

qqline( resid1 )

par(op)

#Shapiro Test for Model 1 and 2

Shap<-matrix(c(shapiro.test(resid1)$statistic,
shapiro.test(resid1)$p.value,shapiro.test(resid1)$statistic ,shapiro.test(resid2)$p.value),

           nrow=2,byrow=T)

#greater than 0.05 , then good

rownames(Shap)<-c("Model1" ,"Model2")

colnames(Shap)<-c("W Statisttic","P−value")

Shap<-data.frame ( Shap )

Shap


```

```{r}

#Independence/Correlation diagnostics

b1<-Box.test(resid1, lag = 12, type = "Box-Pierce", fitdf = 2)$p.value

# Cor

b2<-Box.test(resid1 , lag = 12, type = "Ljung-Box", fitdf = 2)$p.value #Cor

b1 #>0.05 good

b2 #>0.05 good

b3<-Box.test(resid2, lag = 12, type = "Box-Pierce", fitdf = 2)$p.value

# Cor

b4<-Box.test(resid2 , lag = 12, type = "Ljung-Box", fitdf = 2)$p.value #Cor
~~~

~~~
b3 #>0.05

b4 #>0.05

boxT<-matrix(c(b1,b2,b3,b4) ,nrow=2,byrow=FALSE)

rownames(boxT)<-c("Box−Pierce" ,"Ljung−Box")

colnames(boxT)<-c("Model1 P−value" , "Model2 P−value")

boxT


#Test for constant variance of residuals

par(mfrow=c(2 ,2) ) # acf

acf ( resid1 , main = "ACF Plot of Residuals for Model 1" , lag.max=40) # pacf

pacf ( resid1 , main="" , lag.max=40)

title(main="PACF Plots of Residuals for Model 1",outer=FALSE,line=1) # acf

acf ( resid2 , main = "ACF Plot of Residuals for Model 2" , lag.max=40) # pacf

pacf ( resid2 , main="" , lag.max=40)

title (main="PACF Plot of Residuals for Model 2" ,outer=FALSE, line=1)

par(op)


```

```{r}

##Model 2 since AIC and BIC smaller

#forcasting based on Final model

pred.tr <- predict(fit2 ,n.ahead = 10)

U.tr= pred.tr$pred + 2*pred.tr$se # upper bound for the C. I . for transformed data

L.tr= pred.tr$pred - 2*pred.tr$se # lower bound for the C. I . for transformed data

ts.plot(tryimport , xlim=c(1,length(tryimport)+10),main="Forcasting Based on Transform
Data",ylab="")

lines (U.tr , col="blue" , lty="dashed")

lines (L.tr , col="blue" , lty="dashed")
~~~

~~~
points((length(tryimport)+1):(length(tryimport)+10), pred.tr$pred, col="red")

pred.orig <- pred.tr$pred^(1/lamda) # back−transform to get predictions of original time series

U= U.tr^(1/lamda) # bounds of the confidence intervals

L= L.tr^(1/lamda) # Plot forecasts with original data

import2<-ts(import0[ , 2 ] )

ts.plot(import2 , xlim=c(1,length(import2)) ,main="Forcasting Based on Original

    Data",ylab="import")

lines(U, col="blue", lty="dashed")

lines(L, col="blue", lty="dashed")

points((length(tryimport)+1):(length(tryimport)+10), pred.orig ,col="red")

#zoom effect

ts.plot(import2 , xlim=c(length(import2)-20,length(import2)) ,main="Comparison between
Observed Values and Forcasted Values",ylab="import")

points((length(tryimport)+1):(length(tryimport)+10),import2[331:340], col="dark green")

points((length(tryimport)+1):(length(tryimport)+10),pred.orig , col="red")

lines((length(tryimport)+1):(length(tryimport)+10),U, lty=2, col="blue")

lines((length(tryimport)+1):(length(tryimport)+10),L, lty=2, col="blue")

#close to observed value . within confidence interval , good forcasting


```


```
~~~