**Time Series Forecasting**

1. Objective

Our objective is to forecast Australian monthly gas production for 12 time period in future using Time Series data Gas.

1. Exploratory Analysis of Time Series data
   1. Introduction

We have a Time Series data which contains Australian monthly gas production data starts form Jan 1956 and ends at Aug 1995.

* 1. Structure and Summary of data

Data is already in Time-Series format

str(data)-

Time-Series [1:476] from 1956 to 1996: 1709 1646 1794 1878 2173 ...

summary(data)-

Min. 1st Qu. Median Mean 3rd Qu. Max.

1646 2675 16788 21415 38629 66600

* 1. Plot of the data

Let’s look at the plot of data to have overall idea of underling patterns and components of data.

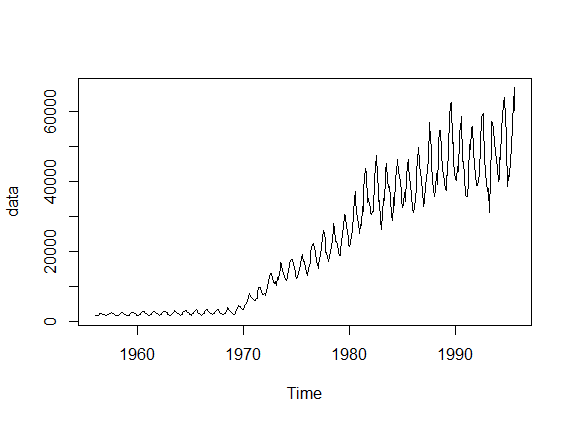


Figure 1- Plot of Time Series data

* 1. Observations from the Plot of Time Series data
* We can see that before 1970 there is only seasonality and inter year fluctuations seems constant across years also no trend component is seen.
* After 1970 series is changed now, we can see there is both trend and seasonality in data and also as trend increases the amplitude of seasonal activity also increases which indicate it’s a multiplicative time series.
* It seems there is no outliers in the data.
  1. Periodicity of data

The data is a monthly series because the values or data points are recorded after each month. So, periodicity or frequency of dataset is 12.

* 1. Trimming the data

We can see in Figure 1 that before 1970 data is showing too much different behaviour and its too much in the past so for our analysis, we can use data from 1970 onwards.

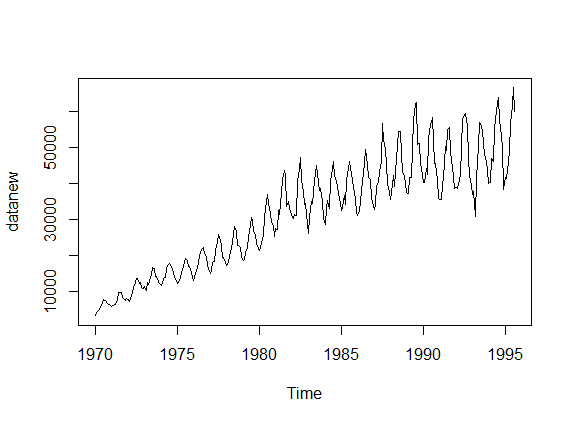


Figure 2- Plot of trimmed Time Series data

* Trend – YES
* Seasonality – YES
* Cyclicity – NO
* Outliers – NO
  1. Transformation of Multiplicative Time series in Additive Time series

Some models need Additive Time series because they do not work on Multiplicative Time series for example – Forecasting by decomposition.

Using log transformation- datanew1<-log(datanew)

Yt=Tt\*St\*It 🡺 log(Yt)=log(Tt)+log(St)+log(It)

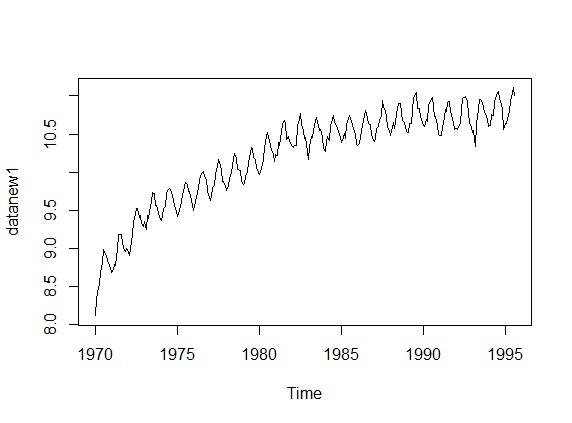


Figure 3- Plot of Transformed Time series data

* 1. Decomposition of Time series data

Let’s decompose and inspect the various components of time series data.

* + 1. Decomposition of original Time Series data

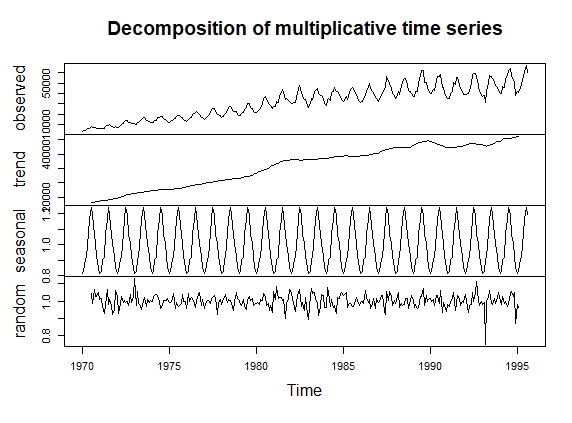


Figure 4- Decomposition of Original Time series data

* + 1. Decomposition of Transformed Time Series data

Here we will use transformed data because its additive and easy to understand. We can use exponential of transformed data to convert again into original data.

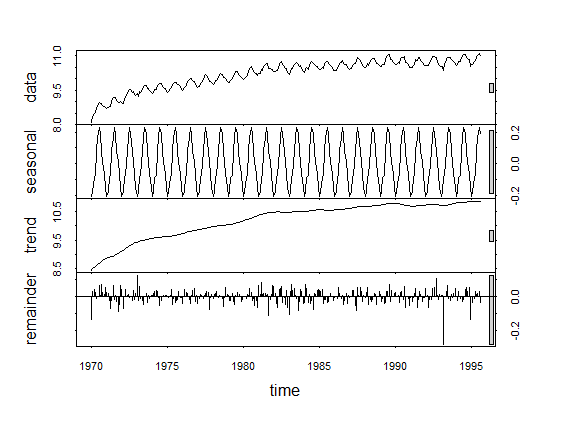


Figure 5- Decomposition of transformed Time series data

* Right hand side rectangles suggest that trend is more important component to explain the time series data then seasonal component.
* There one data point which we can see in reminder which is not explained by trend and seasonality, because when data falls at that point it drops more than usual.
  1. Seasonality of Time Series

Let’s inspect the seasonality component of the Time series

* Month Plot suggests that amount of gas production in each month show sharp increase thought the years.
* In Figure 6&7- of Seasonality of Time Series we can see that Australian gas production is lowest in January and then its gradually increases and reach peak in July then it starts decreasing gradually till December.
* Original values of seasonality are

[1] 0.8147508 0.8374066 0.9143366 0.9299156 1.0882161 1.1794476

[7] 1.2571992 1.2069528 1.0794051 1.0165282 0.9390107 0.8590660

* From Figure 8&9- we are able to visualize the seasonal component more clearly.

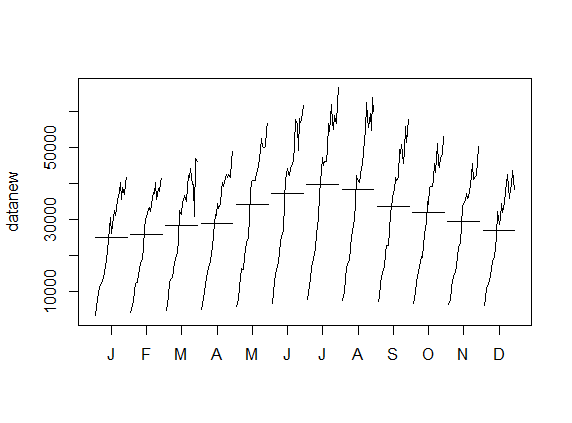


Figure 6- Month Plot

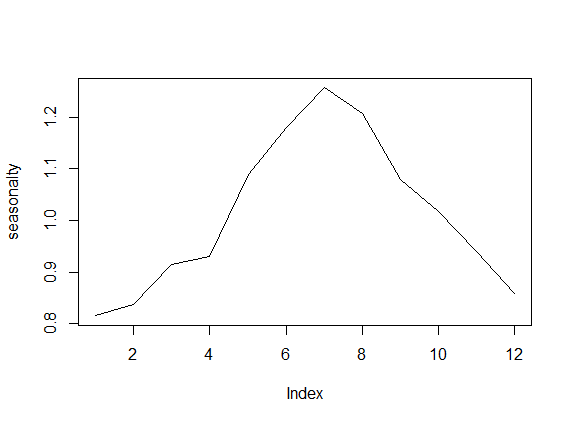


Figure 7- Seasonality of Time Series

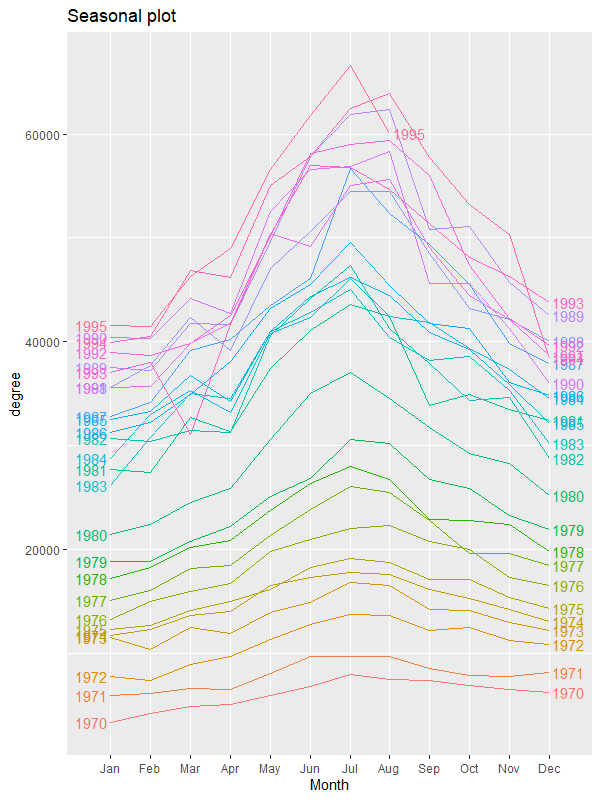


Figure 8- Seasonal plot year- wise

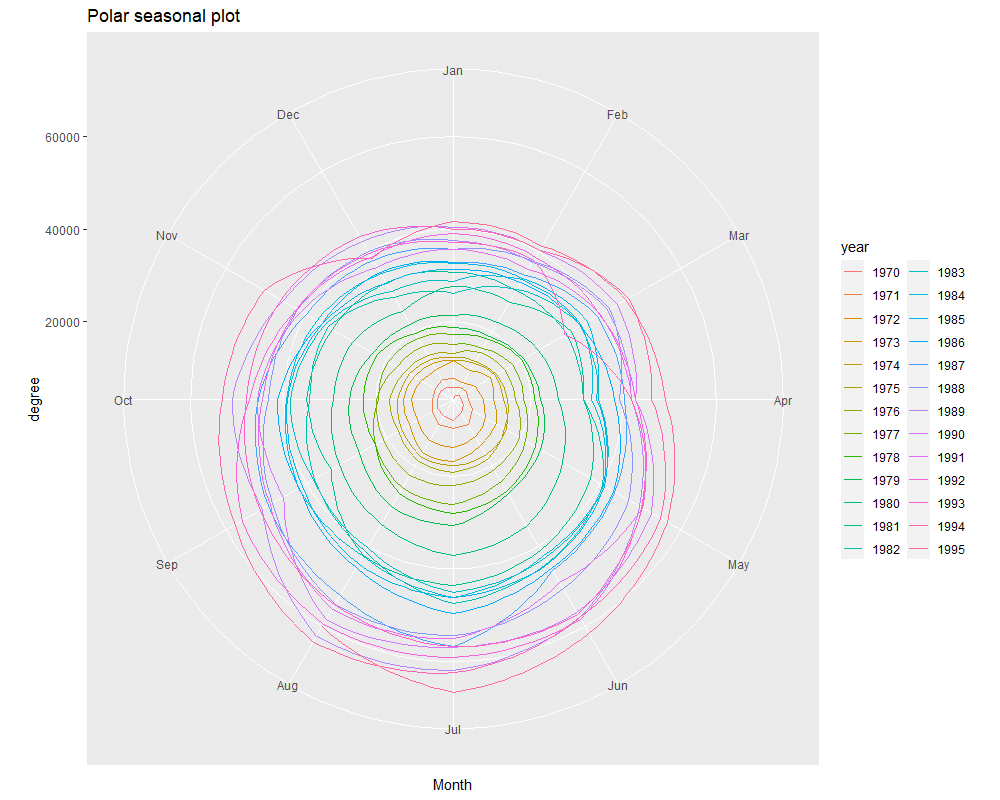


Figure 9- Seasonal plot month -wise

* 1. De- seasonalized Time Series

Let’s inspect Time Series data without the seasonality component by De- seasonalizing the Time Series. We will de- seasonalize the series by removing seasonality component form the decomposed series.

deseason<-decomp1$time.series[,2]+decomp1$time.series[,3]

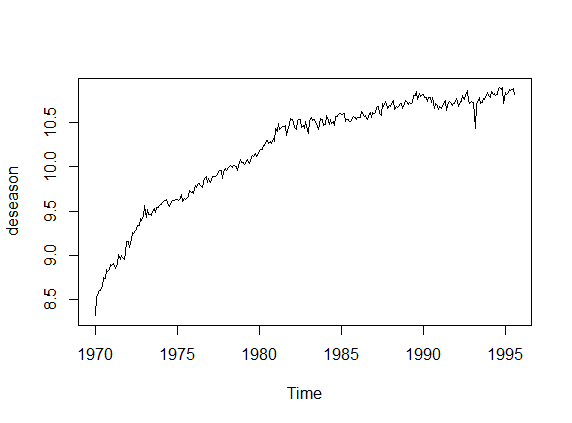


Figure 10- De- seasonalized Time Series

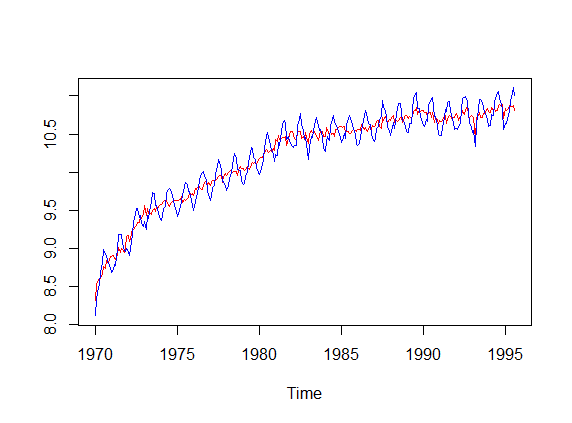


Figure 11- De- seasonalized vs Transformed series

By taking exponential of we can convert our series to original scale. So, let us take exponential of De- seasonalized data and compare it with original series.

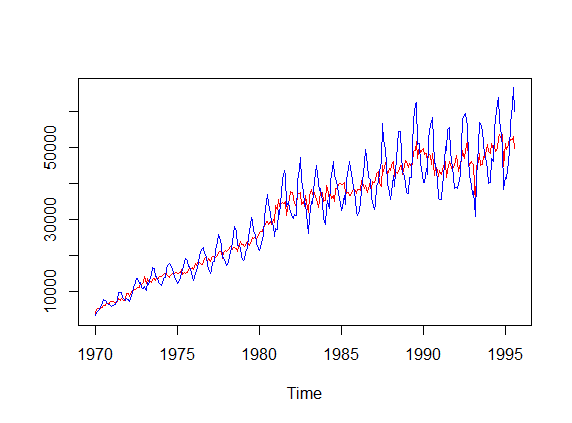


Figure 12- De- seasonalized vs Original series

* We can clearly see an increasing trend after removing seasonality component.
  1. Stationarization of the Series
     1. Visual Inspection
* From Figure 2 we can clearly say that series is not stationary because it contains both trend and seasonal components and variance of series is also changing.
* From Figure 3 we can see that by taking logarithm variance of series is stabilised.
* We can also see that Time Series data contains both Trend and Seasonal component. So, to stabilize Trend we will take ordinary first order difference and to stabilize Seasonality we will take seasonal first order difference.
* Now let’s preform Augmented Dickey-Fuller Test on Transformed Time Series data and check for Stationarity.
  + 1. Hypothesis Testing

Null hypothesis Ho: Time series is non-stationary

Alternative hypothesis Ha: Time series is stationary

Augmented Dickey-Fuller Test

data: datanew1

Dickey-Fuller = -2.8263, Lag order = 6, p-value = 0.228

alternative hypothesis: stationary

We can see that p-value > 0.05 and Null hypothesis is not rejected so we conclude that series is non- stationary.

* + 1. Differencing of Time Series data

Since we concluded in above section that series is non- stationary now, we want to convert the series into stationary.

First, we will take normal first order difference to stabilize Trend

Then we will take seasonal first order difference to stabilize seasonality.

* + - 1. Normal first order difference

datadiff<-diff(datanew1)

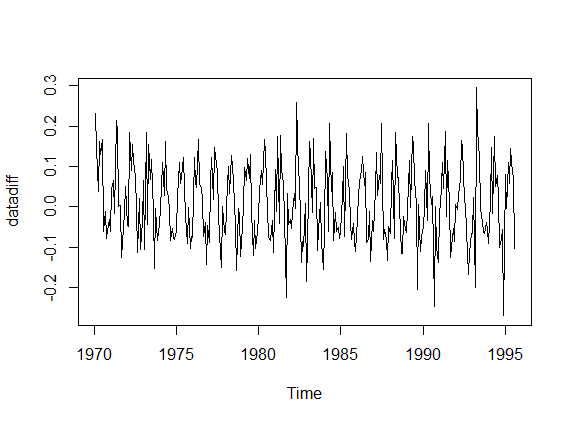


Figure 13- First order difference

We can see after First order difference trend component is stabilized, but this

resulting Time Series still carry seasonal component.

* + - 1. Seasonal first order difference

In previous section we did normal First order differencing and stabilized the trend component. Here in this section we will stabilize the seasonal component by taking first order seasonal difference.

datadiff<-diff(datadiff,12)

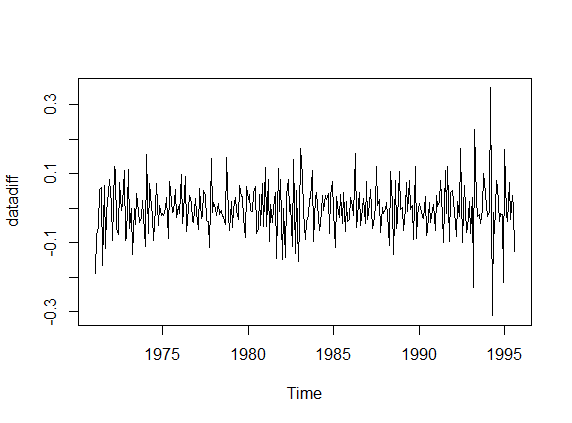


Figure 14- First order difference

* + 1. Hypothesis Testing

Null hypothesis Ho: Time series is non-stationary

Alternative hypothesis Ha: Time series is stationary

Augmented Dickey-Fuller Test

data: datadiff

Dickey-Fuller = -7.5186, Lag order = 6, p-value = 0.01

alternative hypothesis: stationary

We can see that p-value < 0.05 and Null hypothesis is rejected so we conclude that series is now stationary.

* 1. Training and Testing Data

For making and validating models, we will now form training and testing data.

train<-window(datanew1,start=c(1970,1),end=c(1993,12))

test<-window(datanew1,start=c(1994,1))

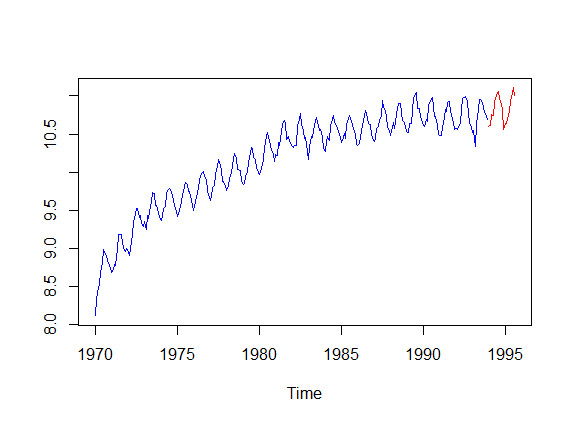
Visualization of training and testing data 

Figure 15- Train and Test data on logarithmic scale

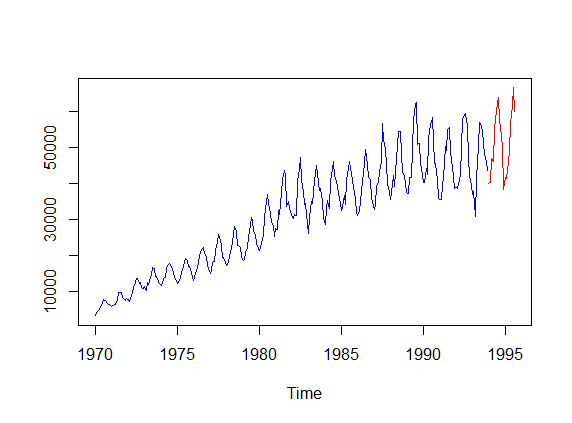


Figure 16- Train and Test data on original scale

* 1. Summary
* First of all, we imported our data which was already in Time-Series format.
* Then we trimmed the data and eliminated very past observations.
* Then we used logarithmic transformation on original data to stabilize the variance of series.
* Then we decomposed our data to analyse various components of Time Series data.
* Then we studied seasonal component.
* Then we de- seasonalized the data and observed the trend in the data.
* Then we stationaries the series by using first order difference and first order seasonal difference.
* And at last we make training and testing dataset.
* Now we will make several models for forecasting.

1. Model preparation and forecasting
   1. Model 1 – Manual ARIMA

* In this model we will use De- seasonalized time series data to prepare our ARIMA model and in last we will add seasonality manually in it.
* So, by making ARIMA model on de- seasonalized data, we will forecast our trend component and because seasonal component repeat itself after every season so it has some constant repeating values which we can add manually to forecasted trend component and finally get ours final forecast.
* In section 2.7 we transformed our data using logarithmic transformation which helped us to stabilize the variance (homogeneous variance).
* Then in section 2.10 we de- seasonalized the data which helped us to eliminate seasonal component for now.
* But remaining de- seasonalized data still contains trend component which means our series is till not stationary and for ARIMA model we need stationary series.
* Let’s perform Augmented Dickey-Fuller Test on de- seasonalized series and then make the series stationary and then prepare our model.
  + 1. Hypothesis Testing of de- seasonalized series

Null hypothesis Ho: Time series is non-stationary

Alternative hypothesis Ha: Time series is stationary

Augmented Dickey-Fuller Test

data: train2

Dickey-Fuller = -2.8461, Lag order = 6, p-value = 0.2196

alternative hypothesis: stationary

We can see that p-value > 0.05 and Null hypothesis is not rejected so we conclude that series is non- stationary this is because trend component was present in the series.

* + 1. Stationarization of the Series by using first order difference

Let’s use first order differencing on de- seasonalized series to stabilize the trend component.

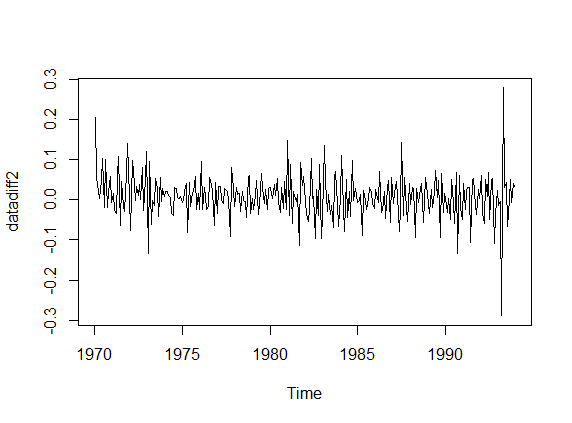
datadiff2<-diff(train2)

Figure 17- First order difference on de- seasonalized series

* + 1. Hypothesis Testing after first order differencing

Null hypothesis Ho: Time series is non-stationary

Alternative hypothesis Ha: Time series is stationary

Augmented Dickey-Fuller Test

data: datadiff

Dickey-Fuller = -7.1063, Lag order = 6, p-value = 0.01

alternative hypothesis: stationary

We can see that p-value < 0.05 and Null hypothesis is rejected so we conclude that series is now stationary.

* + 1. Auto-Correlation and Partial Auto-Correlation Function (ACF and PACF)

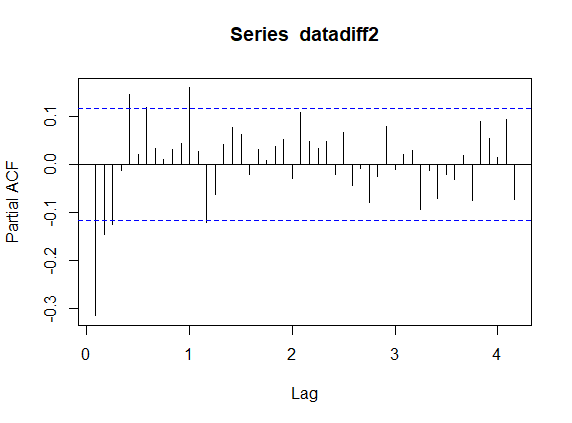
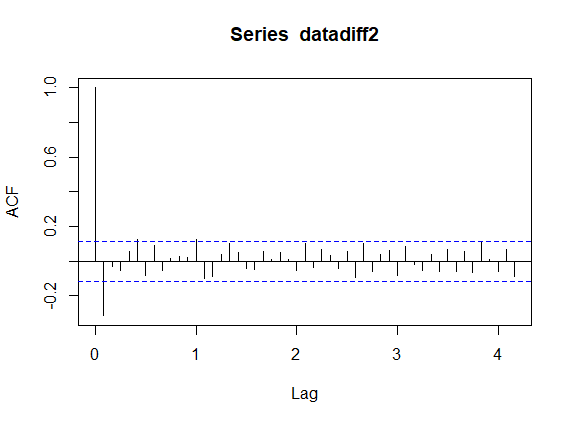


Figure 18- ACF and PACF plot

* + 1. Identification of AR(p), I(d) and MA(q) terms for ARIMA model
* We can see from figure 18- ACF plot that lag(1) show a significant spike and after that barely any lag is significant which indicates MA(q) = 1.
* Now from PACF plot we can see lag (1) to lag (3) are significant and the lag (4) goes inside blue lines and then lag (5) again shows a spike and after that lag (7) and lag (11) shows a significant spike. So, selecting AR(p) term is little confusing but because lag (11) is significant enough let’s take AR(p)= 11.
* Now because after first order differencing, we get our stationary series so I(d)= 1.
  + 1. Fitting ARIMA(11,1,1) model on training data

For this model we have again formed Training and Testing data from de- seasonalized series

train2<-window(deseason,start=c(1970,1),end=c(1993,12))

test2<-window(deseason,start=c(1994,1))

Now model fitting

modelx1<-arima(train2,c(11,1,1))

Summary of model

Call:

arima(x = train2, order = c(10, 1, 1))

Coefficients:

ar1 ar2 ar3 ar4 ar5 ar6 ar7 ar8 ar9 ar10 ma1

0.4900 0.1521 0.0731 0.1662 0.1273 -0.0827 0.0773 -0.0794 -0.0027 0.0723 -0.9356

s.e. 0.0701 0.0685 0.0681 0.0676 0.0678 0.0683 0.0682 0.0686 0.0686 0.0697 0.0371

sigma^2 estimated as 0.00239: log likelihood = 457.87, aic = -891.73

* + 1. Residual Analysis

Now because Residuals are expected to be white noise and ACF of residual is expected to be non- significant to ensure that we didn’t left out any useful information.

* + - 1. Box- Ljung test

Let’s perform Portmanteau Test to check whether the residuals are independent till lag 30

Ho: Residuals are independent

Ha: Residuals are not independent

Box-Ljung test

data: modelx1$residuals

X-squared = 24.16, df = 30, p-value = 0.7648

We can see that p-value >0.05 so null hypothesis is not rejected which implies Residuals are independent

* + - 1. Distribution of Residuals

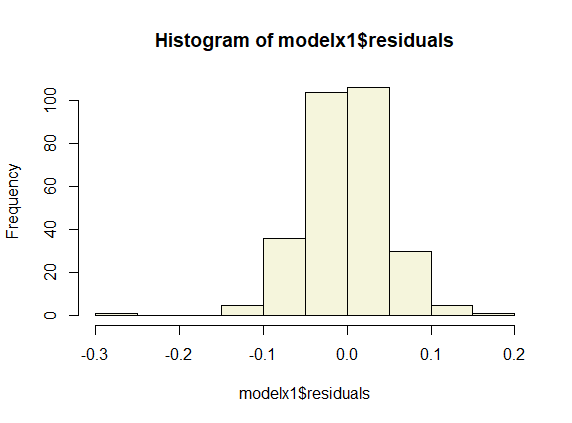


Figure 19- Distribution of Residuals

So, we can see other than one outlier distribution seems to be follow bells curve.

* + - 1. ACF of Residuals

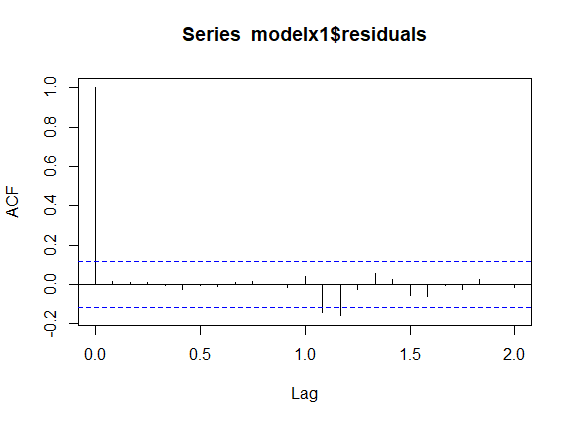


Figure 20- ACF of residuals

By looking at ACF plot it seems Residuals are white noise.

* So, Residuals are independent, they follow normal distribution and it’s a white noise which suggest we didn’t miss out any useful information and our model is good to go.
  + 1. Test forecasting

Let’s do forecasting for next 20 time point using the model prepared from training data and compare the values with original test data.

forecast(modelx1,h=20)

Now we can see from here that there is only trend component forecasted

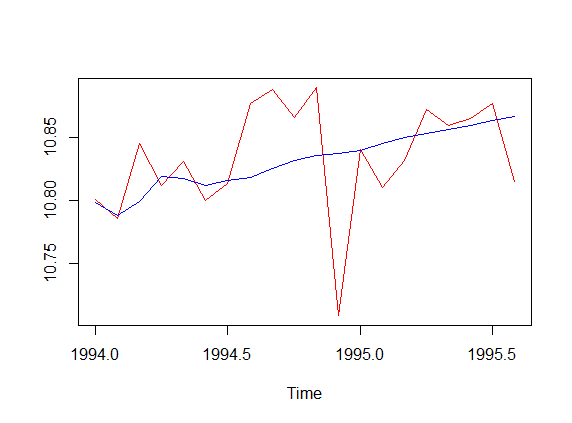


Figure 21- Test Forecasting vs Original test data

* + 1. Adding Seasonal component to the forecast

forecast(modelx1,h=20)$mean+decomp1$time.series[,1][c(1:20)]

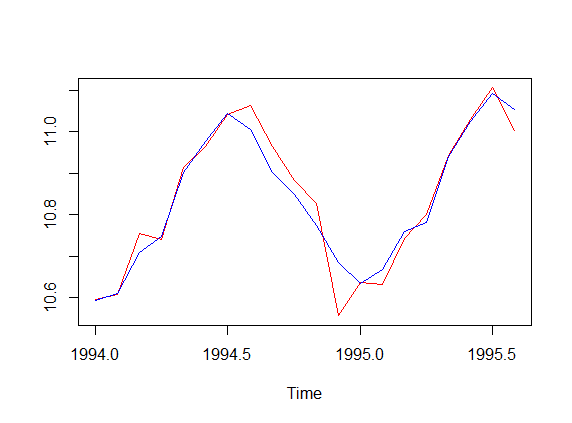


Figure 22- Test Forecasting with seasonality vs Original test data

* + 1. MAPE (Mean absolute percentage error)

MAPEx1<-mean(abs(vecx1[,1]-vecx1[,2])/vecx1[,1])

MAPEx1\*100

MAPE= 2.88%

2.88% seems to be decent enough so now we can build our final model using the full data.

* + 1. Final Model

Because our training model is performing good now, we can make our model using full data and then add seasonality manually

modelfinalx1<-arima(deseason,c(11,1,1))

finalforcastx1<-forecast(modelfinalx1,h=12)$mean+decomp1$time.series[,1][c(9:12,1:8)]

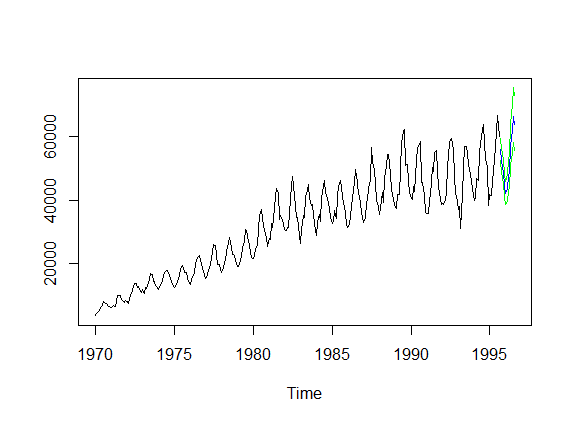


Figure 23- Final Forecast

* + 1. Original values of forecast

Jan Feb Mar Apr May Jun Jul

1995

1996 42074.92 43732.84 47707.01 48808.47 57275.81 62056.24 66307.85

Aug Sep Oct Nov Dec

1995 55957.58 52444.91 48779.88 44549.37

1996 63788.08

* 1. Model 2- SARIMA

In section 2.11 we stationarized the transformed time series by taking first order differencing and then first order seasonal differencing.

Now lets see ACF and PACF plots and determine (p,d,q)(P,D,Q)m terms

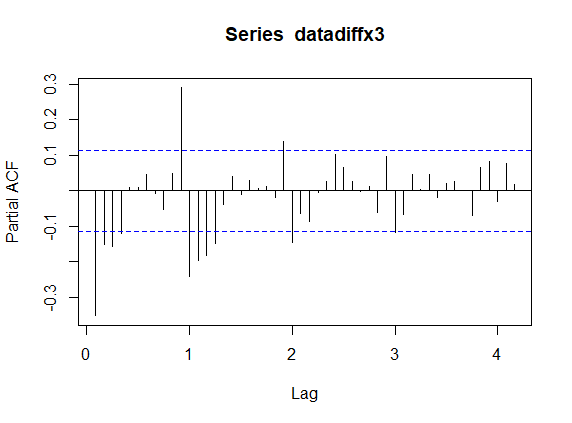
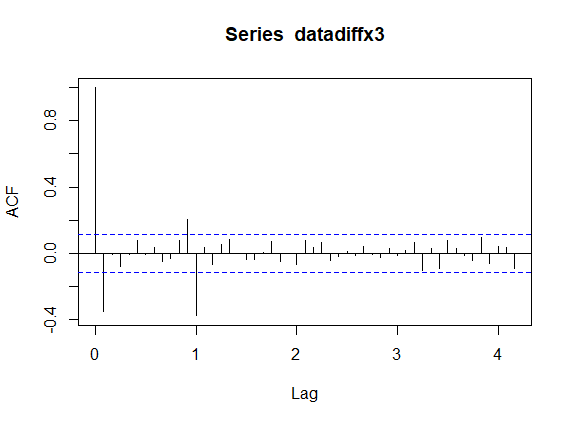


Figure 24- ACF and PACF

* + 1. Identification of AR(p), I(d) and MA(q) terms for SARIMA model
* We can see from figure 24- ACF plot that lag(1) show a significant spike which means MA(q) = 1 and then lag(12) shows significant spike which means seasonal MA(Q)=1 and after that barely any lag is significant.
* Now from PACF plot we can see lag (1) to lag (4) are significant which means AR(p)= 4 and then lag(12), lag(24), lag(36) are significant which means seasonal AR(P)=3.
* Now because after first order differencing then first order seasonal differencing, we get our stationary series so I(d)= 1 and seasonal I(D)=1.
* Periodicity of model is 12 so m=12
  + 1. Fitting SARIMA(4,11)(3,1,1)12 model on training data

armmodl<-Arima(train,order = c(4,1,1),seasonal = c(3,1,1),method = "CSS")

Summary of model

Series: train

ARIMA(4,1,1)(3,1,1)[12]

Coefficients:

ar1 ar2 ar3 ar4 ma1 sar1 sar2 sar3 sma1

-0.3019 -0.2207 -0.2441 -0.0691 -0.2034 0.1264 -0.0254 -0.0182 -0.8661

s.e. 0.2479 0.1250 0.0941 0.0815 0.2467 0.0659 0.0607 0.0569 0.0499

sigma^2 estimated as 0.002142: part log likelihood=437.84

* + 1. Residual Analysis

Now because Residuals are expected to be white noise and ACF of residual is expected to be non- significant to ensure that we didn’t left out any useful information.

* + - 1. Box- Ljung test

Let’s perform Portmanteau Test to check whether the residuals are independent till lag 30

Ho: Residuals are independent

Ha: Residuals are not independent

Box-Ljung test

data: modelx1$residuals

X-squared = 29.72, df = 30, p-value = 0.48

We can see that p-value >0.05 so null hypothesis is not rejected which implies Residuals are independent

* + - 1. Distribution of Residuals

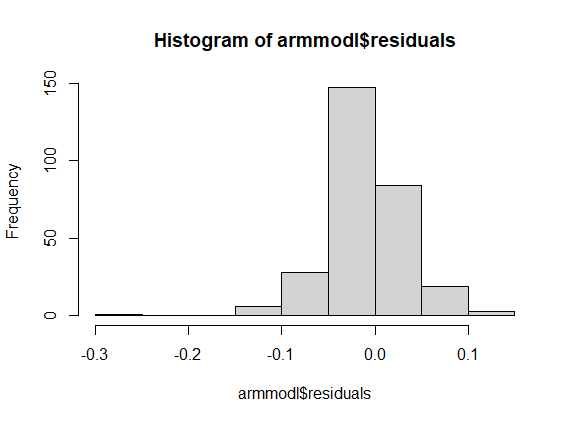


Figure 25- Distribution of Residuals

So, we can see other than one outlier distribution seems to be follow bells curve.

* + - 1. ACF of Residuals

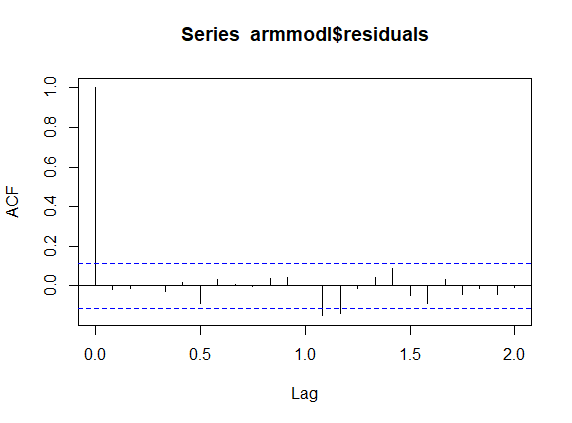


Figure 26- ACF of residuals

By looking at ACF plot it seems Residuals are white noise.

* So, Residuals are independent, they follow normal distribution and it’s a white noise which suggest we didn’t miss out any useful information and our model is good to go.
  + 1. Test forecasting

Let’s do forecasting for next 20 time point using the model prepared from training data and compare the values with original test data.

forecast(armmodl,h=20)

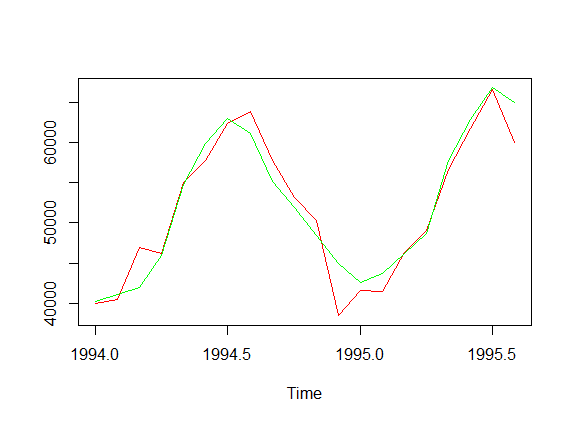


Figure 27- Test Forecasting vs original test data

* + 1. MAPE (Mean absolute percentage error)

MAPEx3<-mean(abs(vecx3[,1]-vecx3[,2])/vecx3[,1])

MAPEx3\*100

MAPE= 3.56%

3.56% seems to be decent enough so now we can build our final model using the full data.

* + 1. Final Model

Because our training model is performing good now, we can make our model using full data.

modelfinalx3<-Arima(datanew1,order = c(4,1,1),seasonal = c(3,1,1),method = "CSS")

finalforcastx3<-forecast(modelfinalx3,h=12)

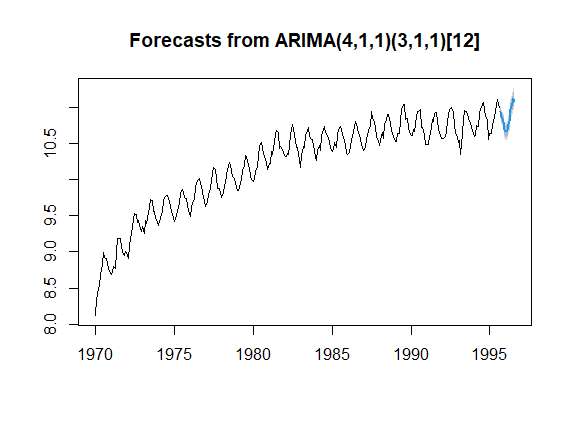


Figure 28- Final Forecast

* + 1. Original values of forecast

Jan Feb Mar Apr May Jun Jul

1995

1996 42711.04 43624.87 47181.24 49238.41 58172.76 63128.78 67378.90

Aug Sep Oct Nov Dec

1995 56274.90 52798.55 49564.15 44020.44

1996 65044.51

* 1. Model 3- Auto.arima

In this model we don’t have to do much work it automates the selection of AR(p) , I(d) , MA(q) terms and give us required model.

* + 1. Fitting auto.arima model on training data

modelx2<-auto.arima(train,seasonal = TRUE)

Summary of model

Series: train

ARIMA(1,1,2)(0,1,2)[12]

Coefficients:

ar1 ma1 ma2 sma1 sma2

-0.0241 -0.4115 -0.0512 -0.6974 -0.1415

s.e. 0.6022 0.5995 0.2775 0.0725 0.0693

sigma^2 estimated as 0.002855: log likelihood=411.21

AIC=-810.42 AICc=-810.11 BIC=-788.72

* + 1. Residual Analysis

Now because Residuals are expected to be white noise and ACF of residual is expected to be non- significant to ensure that we didn’t left out any useful information.

* + - 1. Box- Ljung test

Let’s perform Portmanteau Test to check whether the residuals are independent till lag 30

Ho: Residuals are independent

Ha: Residuals are not independent

Box-Ljung test

data: modelx1$residuals

X-squared = 33.715, df = 30, p-value = 0.2924

We can see that p-value >0.05 so null hypothesis is not rejected which implies Residuals are independent

* + - 1. Distribution of Residuals

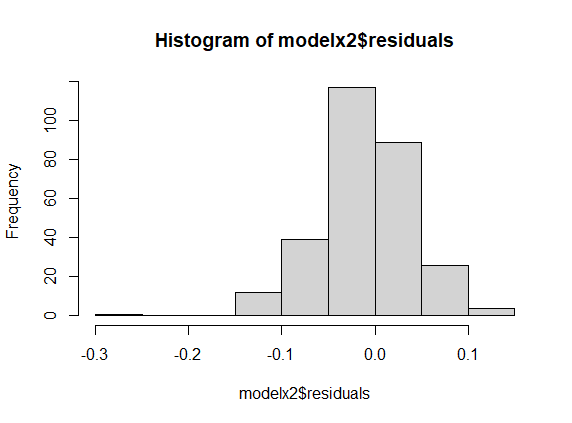


Figure 29- Distribution of Residuals

So, we can see other than one outlier distribution seems to be follow bells curve.

* + - 1. ACF of Residuals

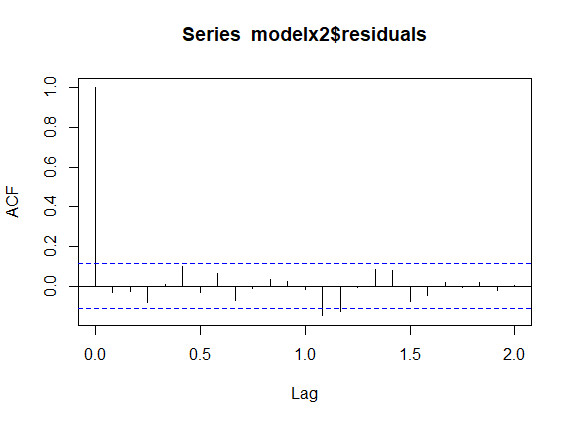


Figure 30- ACF of residuals

By looking at ACF plot it seems Residuals are white noise.

* So, Residuals are independent, they follow normal distribution and it’s a white noise which suggest we didn’t miss out any useful information and our model is good to go.
  + 1. Test forecasting

Let’s do forecasting for next 20 time point using the model prepared from training data and compare the values with original test data.

forecast(armmodl,h=20)

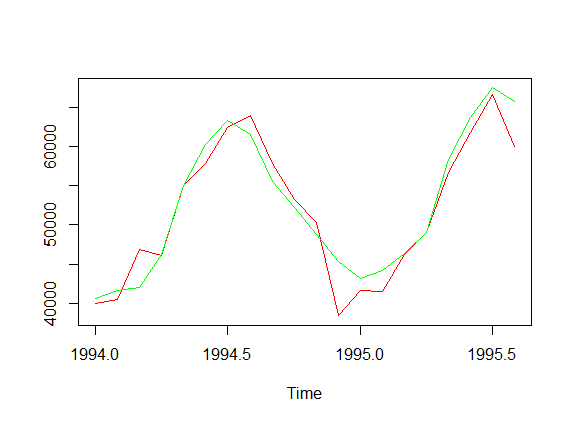


Figure 31- Test Forecasting vs original test data

* + 1. MAPE (Mean absolute percentage error)

MAPEx2<-mean(abs(vecx2[,1]-vecx2[,2])/vecx2[,1])

MAPEx2\*100

MAPE= 3.95%

3.95% seems to be decent enough so now we can build our final model using the full data.

* + 1. Final Model

Because our training model is performing good now, we can make our model using full data.

modelfinalx2<-auto.arima(datanew1,seasonal = TRUE)

finalforcastx2<-forecast(modelfinalx2,h=12)

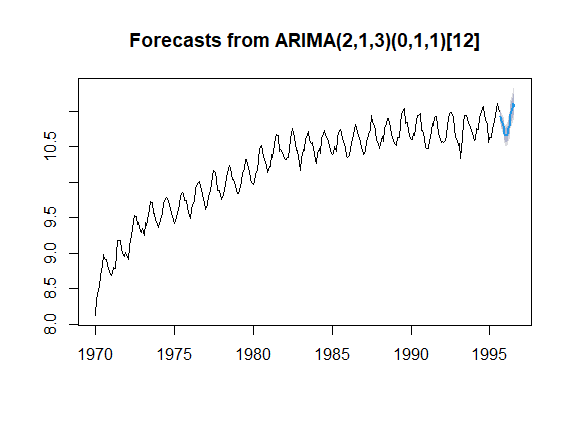


Figure 32- Final Forecast

* + 1. Original values of forecast

Jan Feb Mar Apr May Jun Jul

1995

1996 42656.06 43541.93 46880.96 48757.08 57662.68 62551.92 66712.93

Aug Sep Oct Nov Dec

1995 55821.79 52642.98 48946.71 44141.64

1996 64659.90

1. Accuracy of Model

Training set error of Train data Model

* 1. Model 1

Training set error measures:

ME RMSE MAE MPE MAPE MASE ACF1

Training set -0.002097938 0.04877562 0.03578238 -0.0224173 0.3539363 0.8888596 0.01566407

* 1. Model 2

Training set error measures:

ME RMSE MAE MPE MAPE MASE ACF1

Training set -0.00556439 0.04447802 0.02931977 -0.05497589 0.2812754 0.2669094 -0.01852382

* 1. Model 3

Training set error measures:

ME RMSE MAE MPE MAPE MASE ACF1

Training set -0.01112353 0.05173519 0.03772915 -0.1139543 0.3720522 0.3434633 -0.03151261

Training set error of Final Model

* 1. Model 1

Training set error measures:

ME RMSE MAE MPE MAPE MASE ACF1

Training set -0.002265717 0.04897111 0.03566277 -0.02400631 0.3515483 0.8842029 0.01629762

* 1. Model 2

Training set error measures:

ME RMSE MAE MPE MAPE MASE ACF1

Training set -0.006152884 0.0453045 0.0298803 -0.06067421 0.2860116 0.2753709 -0.01945022

* 1. Model 3

Training set error measures:

ME RMSE MAE MPE MAPE MASE ACF1

Training set -0.00967906 0.05149535 0.03727247 -0.09919849 0.365719 0.3434957 -0.02852354

* 1. MAPE values of fitted training, forecasted test and fitted final model data

|  |  |  |  |
| --- | --- | --- | --- |
|  | Model 1 | Model 2 | Model 3 |
| Fitted training | 3.598595 | 2.969181 | 3.830021 |
| Forecasted test | 2.885176 | 3.563371 | 3.957853 |
| Fitted Final | 3.590679 | 3.030505 | 3.78305 |

* 1. Interpretation

Section 4.1, 4.2, 4.3 shows various training set error of Model 1, Model 2, Model 3 on training data set.

These error values are of transformed data

ME- is minimum of Model 3

RMSE- is minimum of Model 2

MAE- is minimum of Model 2

MPE- is minimum of Model 1

MAPE- is minimum of Model 2

MASE- is minimum of Model 1

Section 4.4, 4.5, 4.5 shows various training set error of Model 1, Model 2, Model 3 on Final model.

These error values are of transformed data

ME- is minimum of Model 1

RMSE- is minimum of Model 2

MAE- is minimum of Model 2

MPE- is minimum of Model 1

MAPE- is minimum of Model 2

MASE- is minimum of Model 1

SO, according to these values model 2, model 1 both are performing good

According to section 4.7 Model 2 has least MAPE value of fitted Final and fitted Training data and then Model 1.

But MAPE of forecasted test data is least for Model 1.

* 1. Best model

Model 2 i.e. SARIMA model is relatively showing better result on fitted values but its losing little bit at forecasting side but its not very drastic.

On other hand Model 1 i.e. Manual ARIMA it has optimum results on fitted values and it is showing best result in forecasting.

So, both models are good but my suggestion is to use Model 1 i.e. Manual ARIMA.