Tractor sales Forecasting

GROUP MEMBERS:

SHRADDHA SHAHANE

APEKSHA SHETTY

PRIYANKA KURKURE

BHAVESH YADAV

VARUN SHAH

Background:

A tractor and farm equipment manufacturing company known as Power Horse, was established a several years after the World War II. The company has shown a continuous growth in its revenue from tractor sales since its establishment. However, over the years the company has been trying effortlessly to keep its production cost and inventory down because of variability in sales and tractor demand. The management at Power Horse is under great pressure from the shareholders and board to reduce the production cost.

Here we have to develop an ARIMA model to forecast the sale of tractors for next 3 years.

1.Data Analysis:

In this paper, we extracted the data set from https://github.com/sam2015/Machine_Learning/blob/master/Time%20series/Tractor_Ivy_Time%20series/TractorSales.csv

The dataset consists of 144 observations having the total month wise sales data of Tractors for a period of past 12 years.

1.1 Density Plot:

R Code:

```
library(e1071)
par(mfrow=c(1, 1))  # divide graph area in 2 columns
plot(density(data$Number.of.tractors.sold), main="Density Plot: Month.year",
ylab="Frequency", sub=paste("Skewness:", round(e1071::skewness(data$Month.yea
r), 2)))  # density plot for 'Month.year'

## Warning in mean.default(x): argument is not numeric or logical: returning
## NA

## Warning in Ops.factor(x, mean(x)): '-' not meaningful for factors

polygon(density(data$Number.of.tractors.sold), col="red")
```

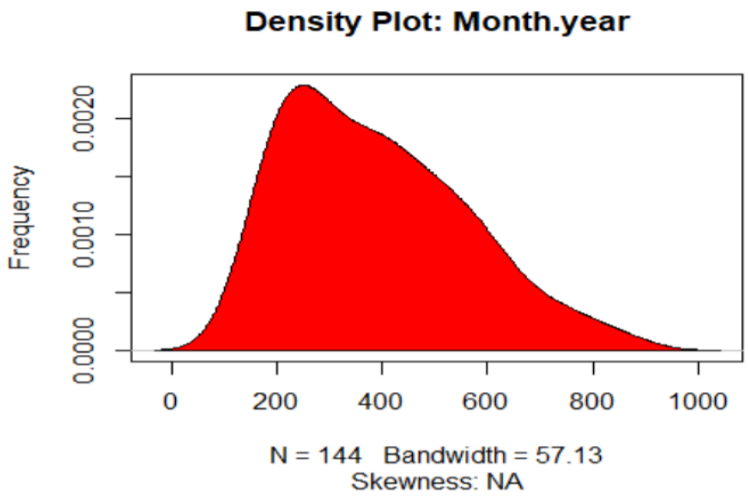


Figure 1.1.1

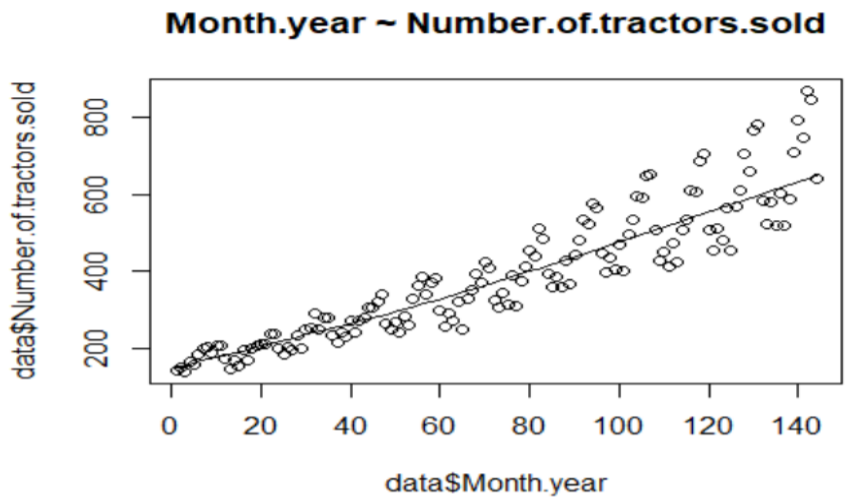
The above is a density plot for Month.year The Density Plot visualizes the distribution of data over a continuous interval or time. Here the plot checks if the response variable is close to normality.

It can be seen from the above plot, that the peak is at about 0.0020 at x=220. It can be inferred that about 0.2% of values are around 220.

Scatter Plot:

R Code:

```
scatter.smooth(x=data$Month.year, y=data$Number.of.tractors.sold, main="Month
.year ~ Number.of.tractors.sold")
```



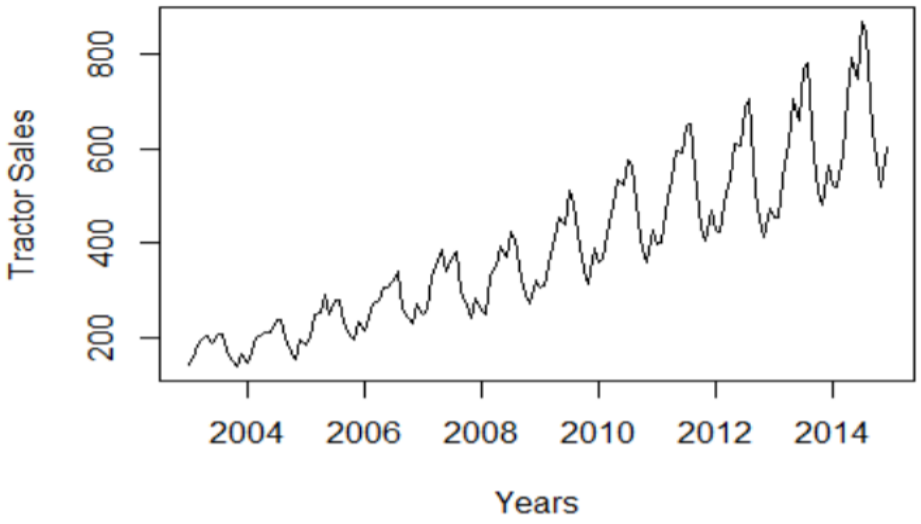
The above is the Scatter plot to visualize the linear relationship between the predictor and response. The X-axis represents the Month.year and the Y-axis represents the Number.of.tractors.sold.

It can be seen from the above plot that there exists a linear relationship between X and Y as the pattern of X– and Y-values resembles a line, with a uphill (with a positive slope) from left to right.

Time Series:

R Code:

```
data<-read.csv("C:\\Users\\APEKSHA\\Downloads\\Tractor sales (1).csv")
data = ts(data[,2],start = c(2003,1),frequency = 12)
plot(data, xlab='Years', ylab = 'Tractor Sales')
```



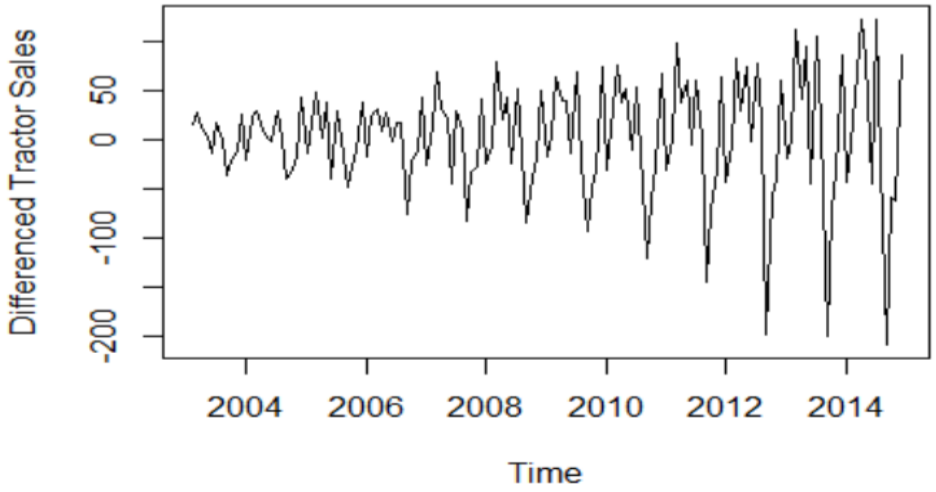
Plotting the tractor sales data as time series. The X-axis represents the Year and the Y-axis represents the Tractor sales.

It can be seen from the above trend, that there is a strong increasing trend, with strong seasonality. There is no evidence of any cyclic behavior here.

Make Data Stationary on Mean(Removing the Trend):

R Code:

```
plot(diff(data),ylab='Differenced Tractor Sales')
```



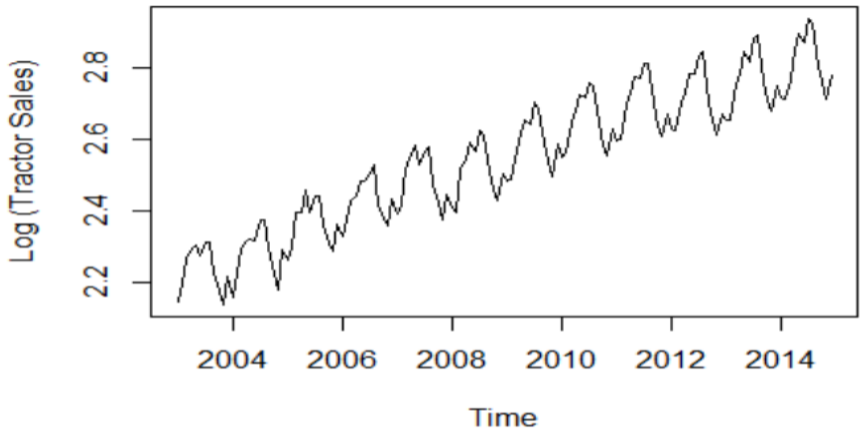
Here in the above plot we are differencing the data to make data stationary on mean. The X-axis represents the time and Y-axis represents the Differenced Tractor sales.

It can be seen from the above trend that it has no trend, seasonality or cyclic behavior.

Make Data Stationary on Variance:(Log Transformation)

R Code:

```
plot(log10(data),ylab='Log (Tractor Sales)')
```



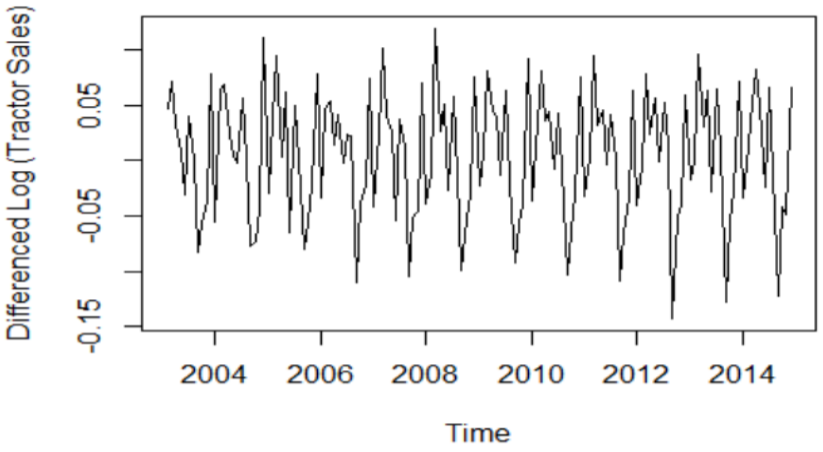
Plotting the log(tractor sales) data as time series. The X-axis represents the Time and the Y-axis represents the Tractor sales.

It can be seen from the above trend, that there is a strong increasing trend, with strong seasonality. There is no evidence of any cyclic behavior here.

Make data stationary on both mean and variance:

R Code:

```
plot(diff(log10(data)), ylab='Differenced Log (Tractor Sales)')
```



So now the above series looks stationary on both mean and variance.

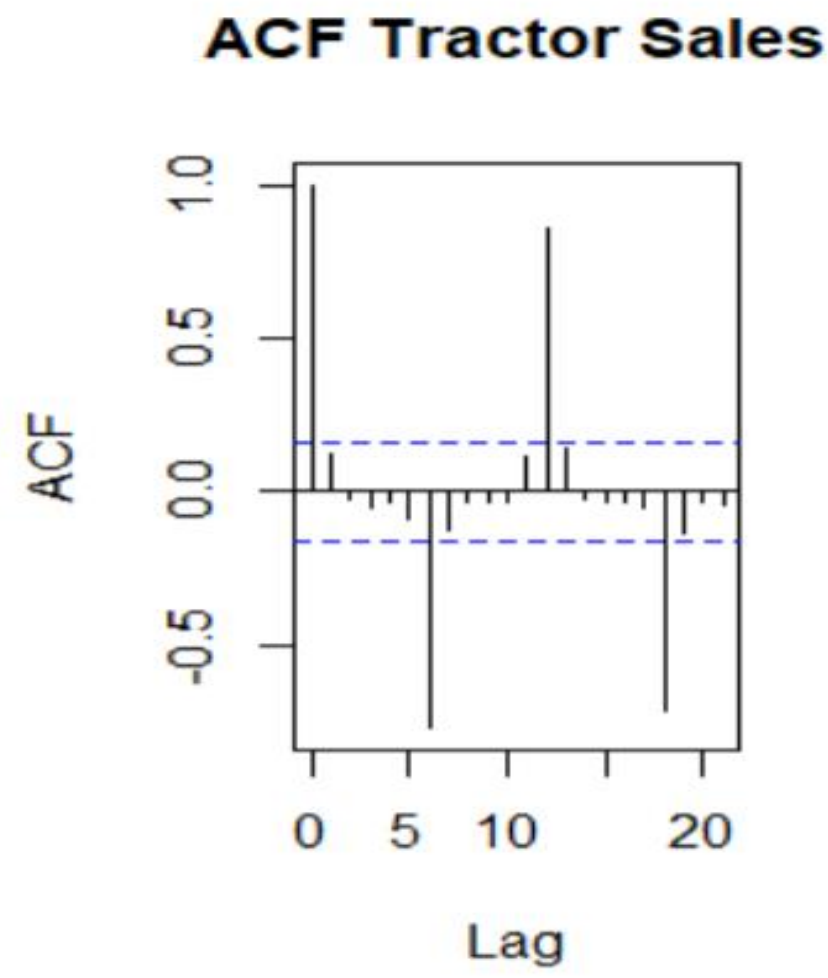
This also gives us the clue that the integrated part of our ARIMA model will be equal to 1 as 1st difference is making the series stationary.

2. ARIMA MODELS:

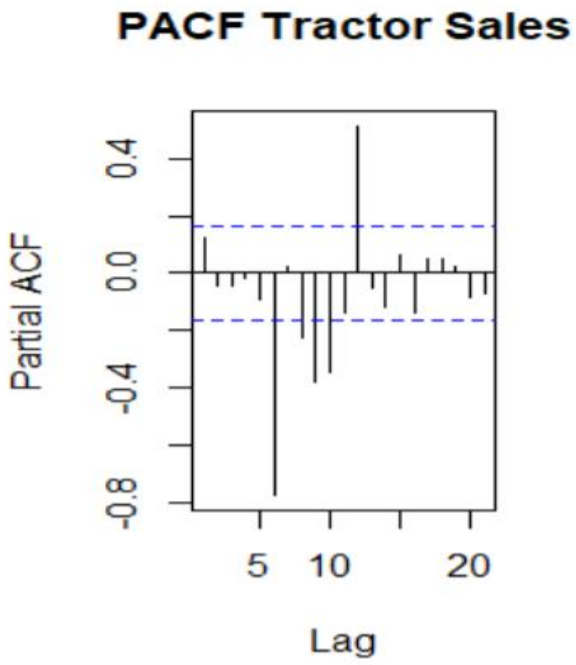
2.1 ACF:

R Code:

```
par(mfrow = c(1,2))
acf(ts(diff(log10(data))),main='ACF Tractor Sales')
pacf(ts(diff(log10(data))),main='PACF Tractor Sales')
```



PACF:



In time series analysis, the partial autocorrelation function (**PACF**) gives the partial correlation of a time series along with its own lagged values. PACF contrast with the autocorrelation function, which does not control for other lags.

Observation by ACF and PACF plots:

Since, there are enough spikes in the plots outside the insignificant zone (dotted horizontal lines) we can conclude that the residuals are not random.

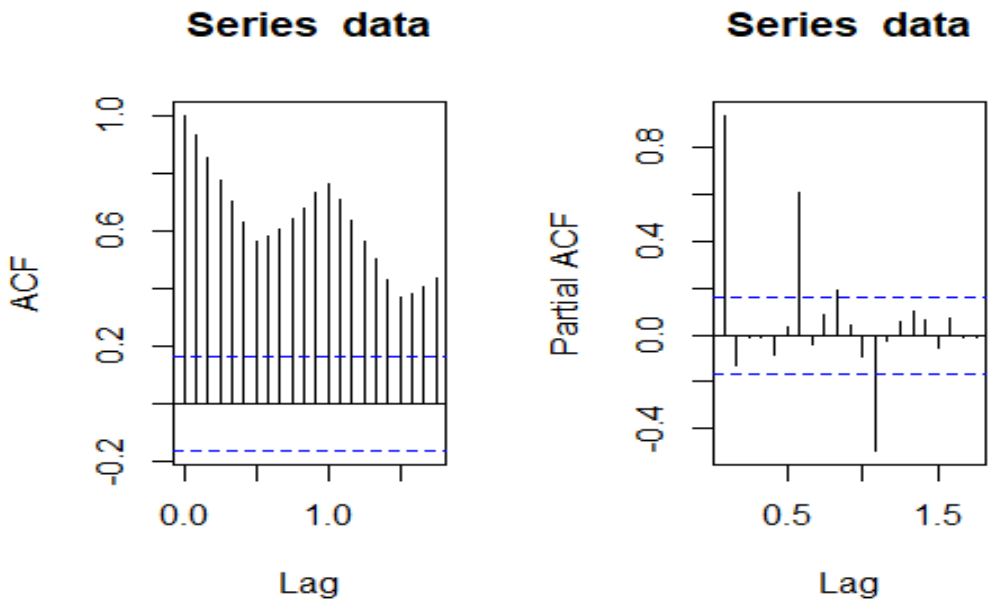
This implies that there is juice or information available in residuals to be extracted by AR and MA models.

Also, there is a seasonal component available in the residuals at the lag 12 (represented by spikes at lag 12). This makes sense since we are analyzing monthly data that tends to have seasonality of 12 months because of patterns in tractor sales.

Series Data:

R Code:

```
acfRes <- acf(data) # autocorrelation
pacfRes <- pacf(data)  # partial autocorrelation
```



On using the ACF and PACF functions, it generates the series data by default

#11 Identification of the best fit ARIMA model

R Code:

```
library(tseries)
ARIMAfit = auto.arima(log10(data), approximation=FALSE,trace=FALSE)
summary(ARIMAfit)

## Series: log10(data)
## ARIMA(0,1,1)(0,1,1)[12]
##
## Coefficients:
##           ma1     sma1
##       -0.4047  -0.5529
## s.e.   0.0885   0.0734
##
## sigma^2 estimated as 0.0002571:  log likelihood=354.4
```

```
## AIC=-702.79   AICc=-702.6   BIC=-694.17
##
## Training set error measures:
##                        ME       RMSE        MAE         MPE      MAPE
## Training set 0.0002410698 0.01517695 0.01135312 0.008335713 0.4462212
##                    MASE       ACF1
## Training set 0.2158968 0.01062604
```

Forecasting the sales using the best fit ARIMA model

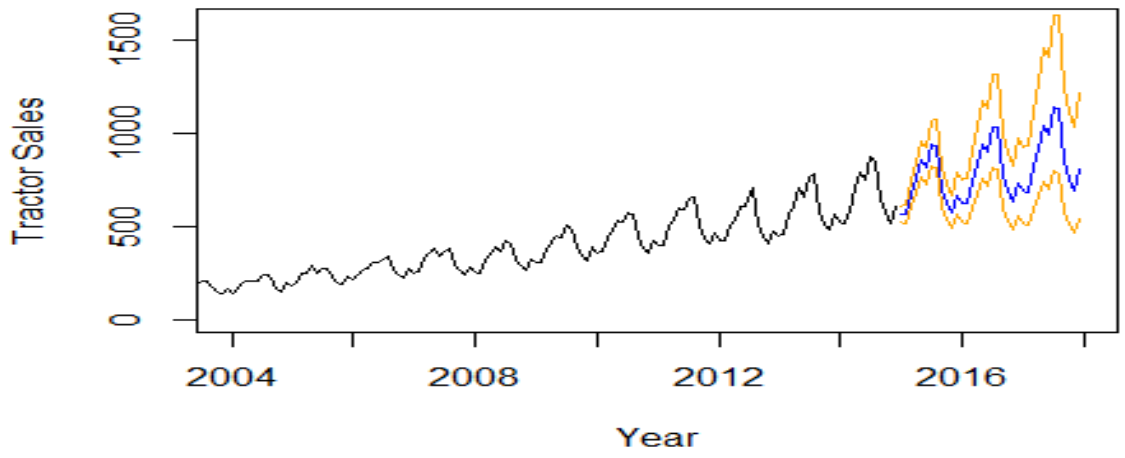
R Code:

```
par(mfrow = c(1,1))
pred = predict(ARIMAfit, n.ahead = 36)
pred
```

```
## $pred
##           Jan      Feb      Mar      Apr      May      Jun      Jul
## 2015 2.754168 2.753182 2.826608 2.880192 2.932447 2.912372 2.972538
## 2016 2.796051 2.795065 2.868491 2.922075 2.974330 2.954255 3.014421
## 2017 2.837934 2.836948 2.910374 2.963958 3.016213 2.996138 3.056304
##           Aug      Sep      Oct      Nov      Dec
## 2015 2.970585 2.847264 2.797259 2.757395 2.825125
## 2016 3.012468 2.889147 2.839142 2.799278 2.867008
## 2017 3.054351 2.931030 2.881025 2.841161 2.908891
##
## $se
##             Jan        Feb        Mar        Apr        May        Jun
## 2015 0.01603508 0.01866159 0.02096153 0.02303295 0.02493287 0.02669792
## 2016 0.03923008 0.04159145 0.04382576 0.04595157 0.04798329 0.04993241
## 2017 0.06386474 0.06637555 0.06879478 0.07113179 0.07339441 0.07558934
##             Jul        Aug        Sep        Oct        Nov        Dec
## 2015 0.02835330 0.02991723 0.03140337 0.03282229 0.03418236 0.03549035
## 2016 0.05180825 0.05361850 0.05536960 0.05706700 0.05871534 0.06031866
## 2017 0.07772231 0.07979828 0.08182160 0.08379608 0.08572510 0.08761165
```

Hence as stated in the problem statement we have forecasted the tractor sales for the next THREE years using the ARIMA Model.

```
plot(data,type='l',xlim=c(2004,2018),ylim=c(1,1600),xlab = 'Year',ylab = 'Tra
ctor Sales')
lines(10^(pred$pred),col='blue')
lines(10^(pred$pred+2*pred$se),col='orange')
lines(10^(pred$pred-2*pred$se),col='orange')
```



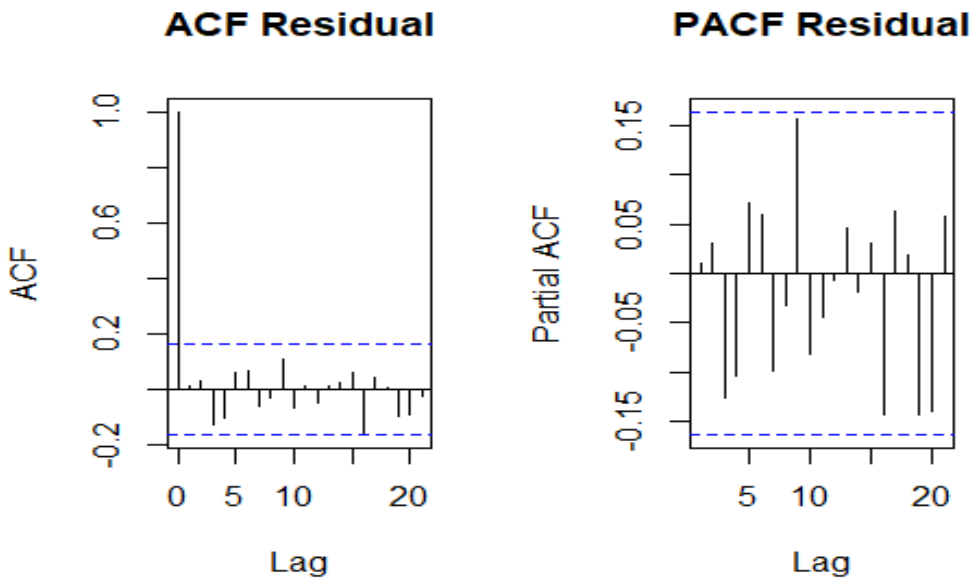
The above given output shows the forecasted values of the tractor sales in blue. Also, the range of expected error which is twice the standard deviation is drawn with orange lines on one of the sides of predicted blue line.

Assumptions while forecasting: Forecasting for a period of THREE years is a big task. The major assumption here is that the underlining patterns in the time series will continue to stay the same as predicted in the model. In a short-term forecasting model, for example, a couple of business quarters or a year, is usually a good idea to forecast with reasonable accuracy. A long-term model like the one above needs to be calculated with frequent interval of time i.e. 6 months. The main reason behind this is to incorporate the new information available with the passage of time in the model.

```
#13 Plotting ACF and PACF for residuals of ARIMA model to check if there's
more information left for extraction.
```

R Code:

```
par(mfrow=c(1,2))
acf(ts(ARIMAfit$residuals),main='ACF Residual')
pacf(ts(ARIMAfit$residuals),main='PACF Residual')
```



With time series data, it is highly likely that the value of a variable observed in the current time will be similar to its value in the previous period, or even the period before that, and so on. Therefore, when fitting a regression model to time series data, it is common to find autocorrelation in the residuals.

In the above graphs we can see that Autocorrelation Function shows no seasonality and no trend because the residual points are below the p value.