**Time Series Assignment**

**–Australia Monthly Gas Production Forecasting**

In this problem we would be exploring the Australia Monthly Gas Production dataset available in the library “forecast” of R.

• Read the data as a time series object in R. Plot the data

• What do you observe? Which components of the time series are present in this dataset?

• What is the periodicity of dataset?

• Is the time series Stationary? Inspect visually as well as conduct an ADF test? Write down the null and alternate hypothesis for the stationarity test? De-seasonalise the series if seasonality is present?

• Develop an ARIMA Model to forecast for next 12 periods. Use both manual and auto.arima (Show & explain all the steps)

• Report the accuracy of the model

**SOLUTION:**

**1. Phase I – EDA**

**# Load the library forecast**

**# Store gas dataset in to a new object**

TSdata<-gas

**# Let’s check the class of the dataset to be sure that it is a time series data set.**

class(TSdata)

- Given time series is univariate, has only one variable.

**# Find the start of the series, end of the series, frequency and cycle.**

**# To print start of the series**

start(TSdata)

**# To print end of the series**

end(TSdata)

**# To print frequency of the series**

frequency(TSdata)

**# To print the cycle across**

cycle(TSdata)

* This series stars from 1956, 1st Month i.e. January. Series end at 1995 ,8th Month i.e August.
* Frequency of the data is 12 which implies that this is a monthly series .
* Cycle indicates that all monthly values are available from 1956 Jan till 1995 August.
* Data set does not have any missing value

**2. Phase II – Data Preparation**

**2.1 Data Exploration and Visualization**

**## Let’s start the data exploration step with the summary function**

summary(TSdata)

* Summary output, because of the difference between median and maximum value, might be read as if data is skewed.
* However, for a time series, such interpretation should be made with a caution as time series may have two additional components i.e. trend and seasonality. Hence this difference could be because of these two components.
* For further analysis, data is aggregated at a quarter and yearly level and then plot is made for the original series along with the series aggregated at a quarter and year level.

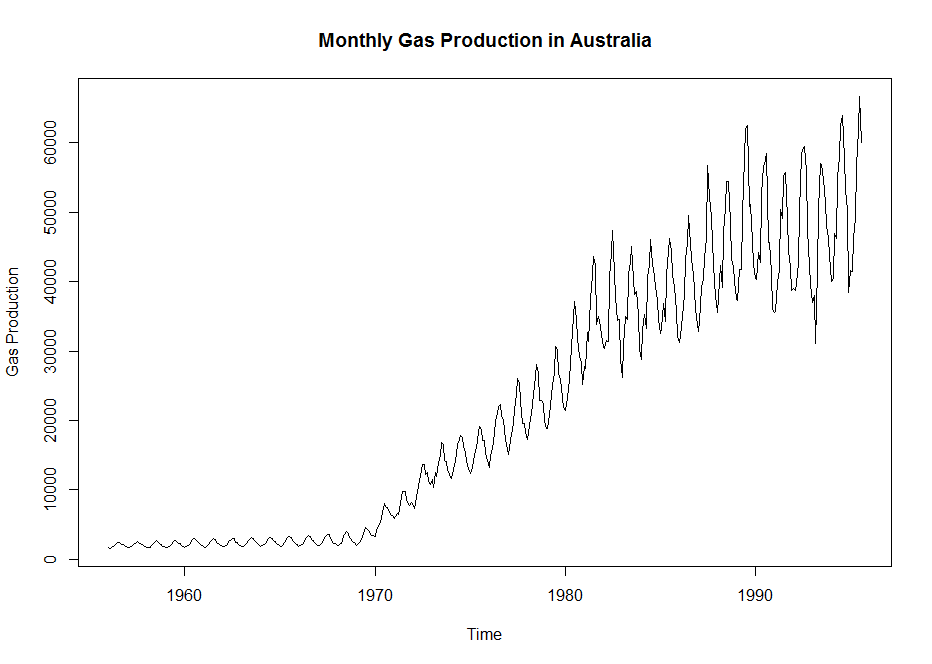
**## Aggregation at a Quarter and Year Level**

TSdata.qtr <- aggregate(TSdata, nfrequency=4)

TSdata.yr <- aggregate(TSdata, nfrequency=1)

**## Plots**

plot.ts(TSdata, main = "Monthly Gas Production in Australia", xlab = "Time", ylab = "Gas Production")



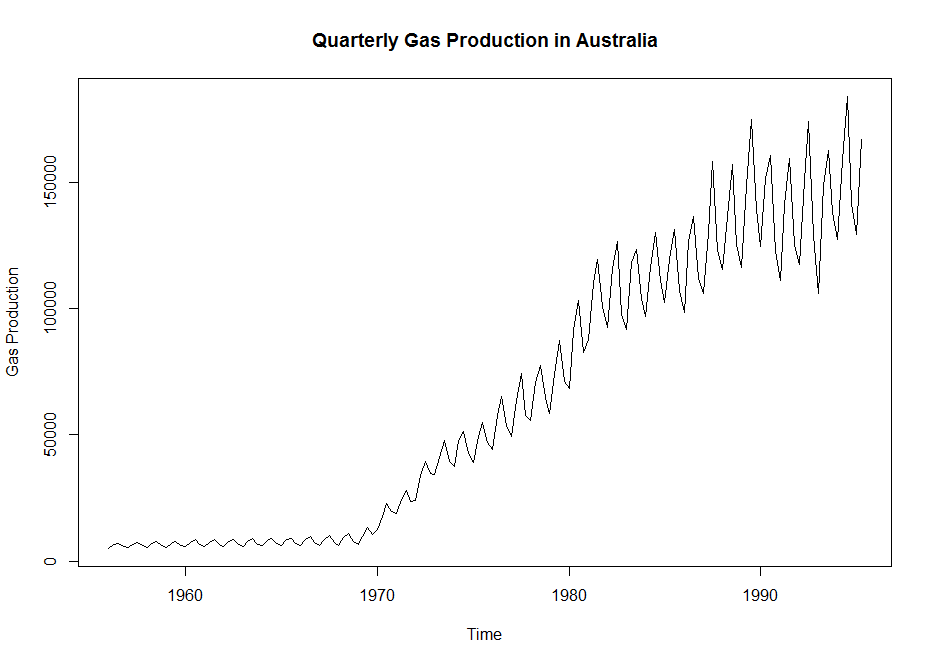
The above graph shows an increasing trend. It does indicate some seasonality; however, we will separately make a seasonality graph to study the same.

Starting around 1970, gas production started to increase and by 1980 had become extremely variable month to month.

**## Quarter plot**

plot.ts(TSdata.qtr, main = "Quarterly Gas Production in Australia",

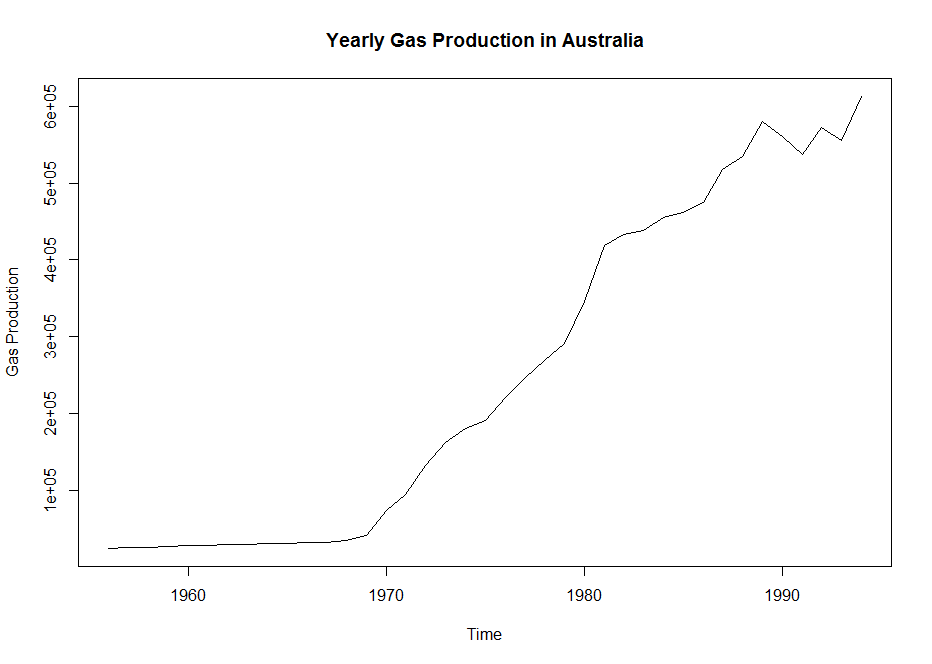
xlab = "Time", ylab = "Gas Production")



Quarter plot shows some clear indication of seasonality

**## Yearly plot**

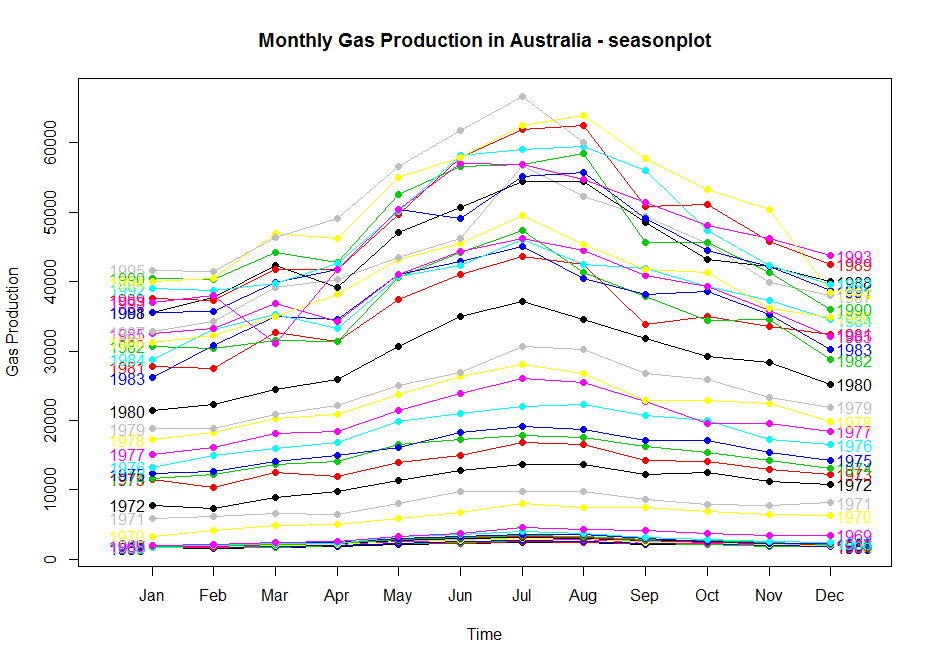
plot.ts(TSdata.yr, main = "Yearly Gas Production in Australia", xlab = "Time", ylab = "Gas Production")



**Yearly plot has clear indication of trend in the data set**

**## Below are Seasonality Plot for further analysis**

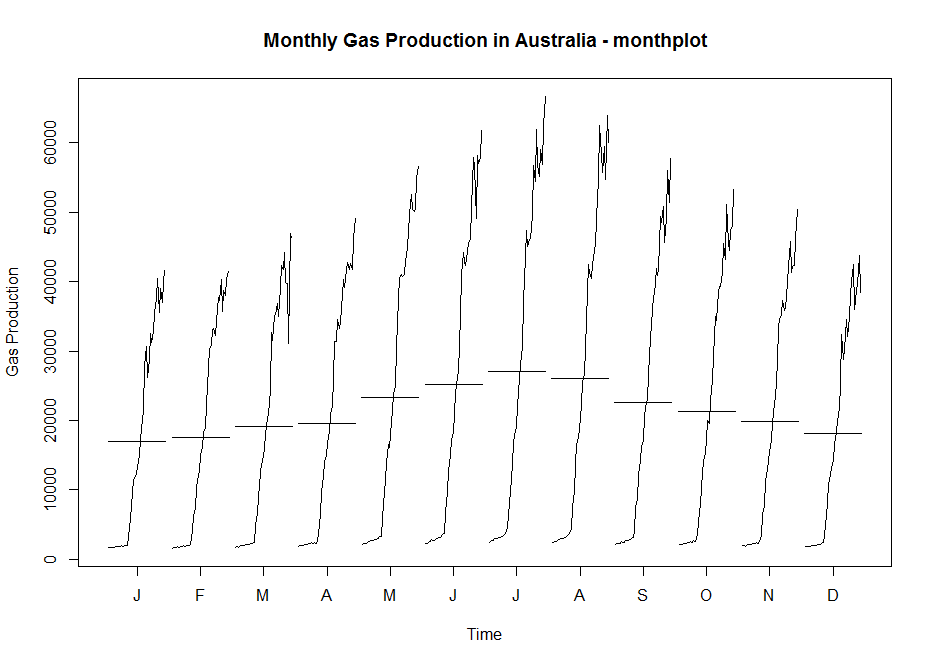
seasonplot(TSdata, year.labels = TRUE, year.labels.left=TRUE, col=1:40, pch=19, main = "Monthly Gas Production in Australia - seasonplot", xlab = "Time", ylab = "Gas Production")



* For the initial years, seasonality was not prevalent.
* However over years, seasonality is visible from May to October with July showing the peak value across all years. Series has clear semi-annual seasonality.

**## Monthly gas production**

monthplot (TSdata, main = "Monthly Gas Production in Australia - monthplot", xlab = "Time", ylab = "Gas Production")



**2.2 Decomposition of time series**

A time series decomposition is procedure which transform a time series into multiple different time series. The original time series is often computed (decompose) into 3 sub-time series:

Seasonal: patterns that repeat with fixed period of time.

Trend: the underlying trend of the metrics.

Random: (also call “noise”, “Irregular” or “Remainder”) Is the residuals

of the time series after allocation into the seasonal and trends time series.

Other than above three component there is Cyclic component which occurs after long period of time

**Additive or multiplicative decomposition?**

To get a successful decomposition, it is important to choose between the additiveor multiplicative model. To choose the right model we need to look at the time series.

• The additive model is useful when the seasonal variation is relatively constant

over time.

• The multiplicative model is useful when the seasonal variation increases over time.

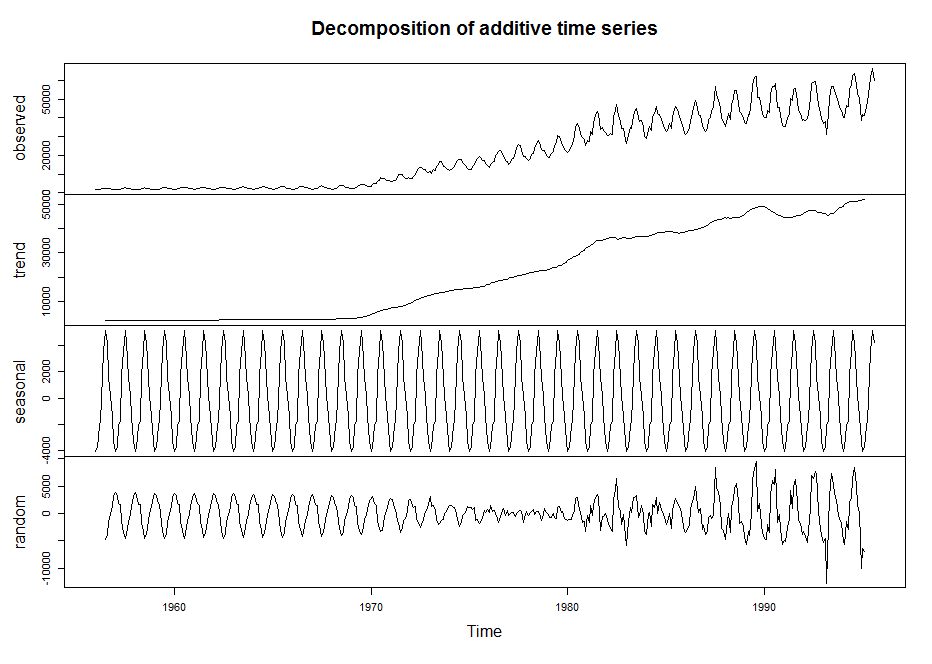
**## How to visually differentiate an additive and Multiplicative Model ?**

In an additive model, the amplitude of both the seasonal and irregular variations do not vary as the level of the trend rises or falls.

* Decomposition is a tool that we can separate different components in a time series data so we can see trend, seasonality, and random noises individually.
* As seasonality pattern does not increases with time in our series, hence the series is assumed to be additive.

decompgas = decompose(TSdata, type="additive")

plot (decompgas)



* Above decomposition clearly shows the gas production is trending upward with clear semi-annual seasonality.

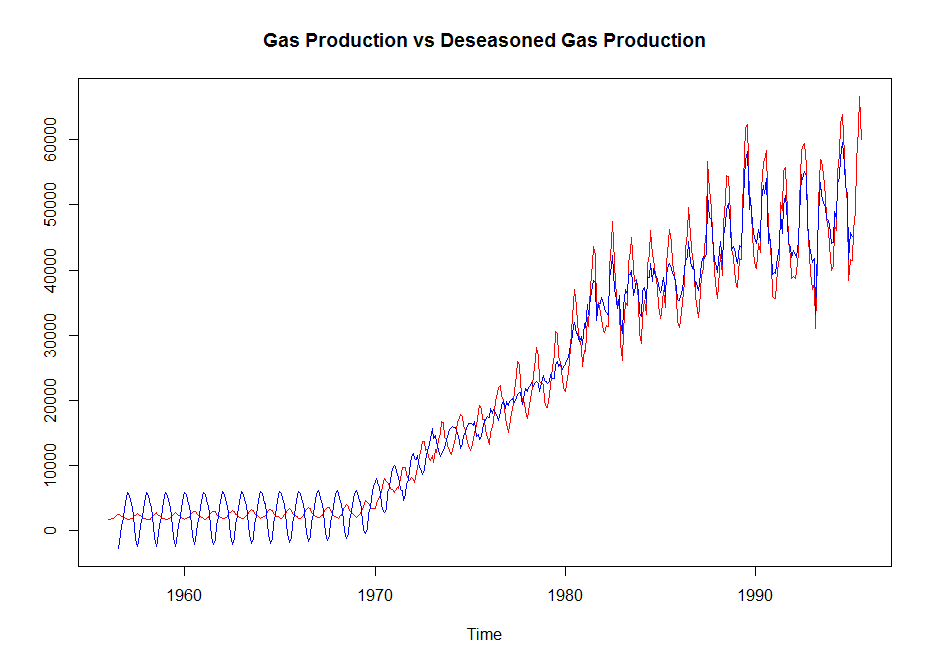
**2.3 Depersonalised the Time Series**

If the focus is on figuring out whether the general trend of production is up, we deseasonalize, and possibly forget about the seasonal component. However, if you need to forecast the production in next month, then you need take into account both the secular trend and seasonality.

* As the series is additive, trend and random component of the series are added to deasonalise the series. Then desasonalised and original data set are plotted to study the trend.

Deseason\_gas <- (decompgas$trend+decompgas$random)

ts.plot(TSdata, Deseason\_gas, col=c("red", "blue"), main="Gas Production vs Deseasoned Gas Production")



Above plot show original series in Red and de-seasoned production in Blue, we can see that there is increasing trend of production.

**2.4 Data Preparation – Training and Testing Dataset**

Before starting with the decomposition, let’s extract series from 1970 onwards as this was the period gas production started increasing, i.e. showing a trend.

TSdata\_new <- ts(TSdata, start=c(1970,1),end=c(1995,8), frequency=12)

**## Divide data into test and train**

DataATrain <- window(TSdata\_new, start=c(1970,1), end=c(1993,12), frequency=12)

DataATest <- window(TSdata\_new, start=c(1994,1), frequency=12)

Test and train dataset have been created to ensure that at least one full cycle datais available in the test dataset.

**3. Phase III – Model Planning and Building**

**3.1 Model Planning**

**3.1.1 Check for stationary time series**

**### Dickey–Fuller test**

Statistical tests make strong assumptions about your data.They can only be used to inform the degree to which a null hypothesis can be acceptedor rejected. The result must be interpreted for a given problem to be meaningful.

Nevertheless, they can provide a quick check and confirmatory evidence that your time series is stationary or non-stationary.

Null Hypothesis (H0): If accepted, it suggests the time series has a unit root,

meaning it is non-stationary. It has some time dependent structure.

Alternate Hypothesis (H1): The null hypothesis is rejected;

it suggests the time series does not have a unit root, meaning it is stationary.

It does not have time-dependent structure.

p-value > 0.05: Retain the null hypothesis (H0), the data has a unit root and is

non-stationary. p-value <= 0.05: Reject the null hypothesis (H0), the data does

not have a unit root and is stationary.

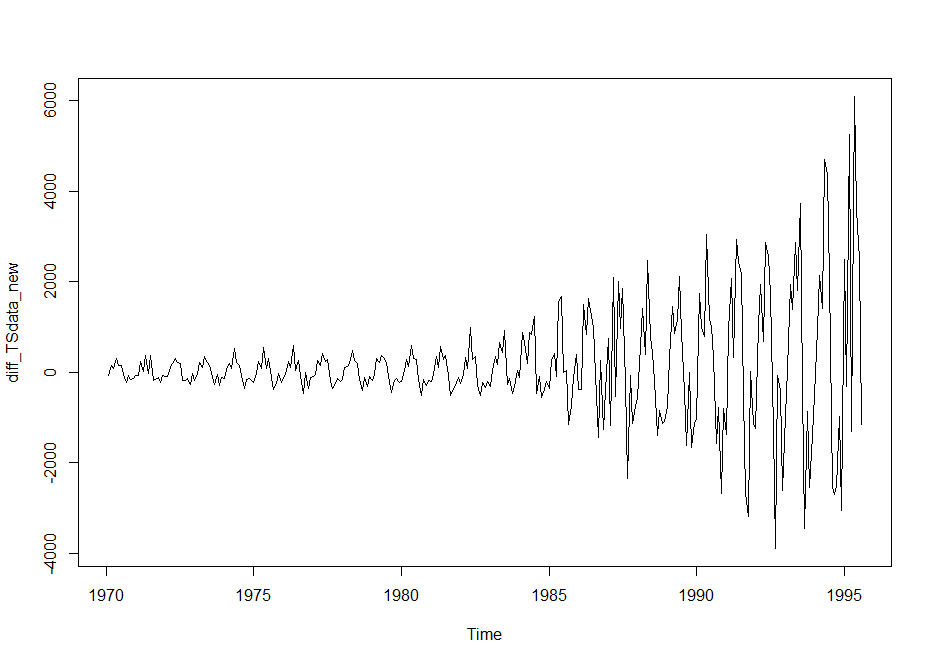
library(tseries)

adf.test(TSdata\_new)

* Null Hypothesis is retained , hence gas data is non-stationary; the average gas production changes through time.
* As for the Arima model it is mandatory for a series to be stationary, hence the next step is to perform difference transformation and observe through the plot if the series is stationary or not

diff\_TSdata\_new <- diff(TSdata\_new)

plot(diff\_TSdata\_new)



From the plot, series does look stationary.

**Let’s perform Dicky Fuller test on the differenced series to confirm the same.**

adf.test(diff\_TSdata\_new)

In adf.test(diff\_TSdata\_new) : p-value smaller than printed p-value

From the above ADF test the Null Hypothesis is rejected.

The time series of differences (above) does appear to be stationary in mean and variance,

as the level of the series stays roughly constant over time, and the variance of the

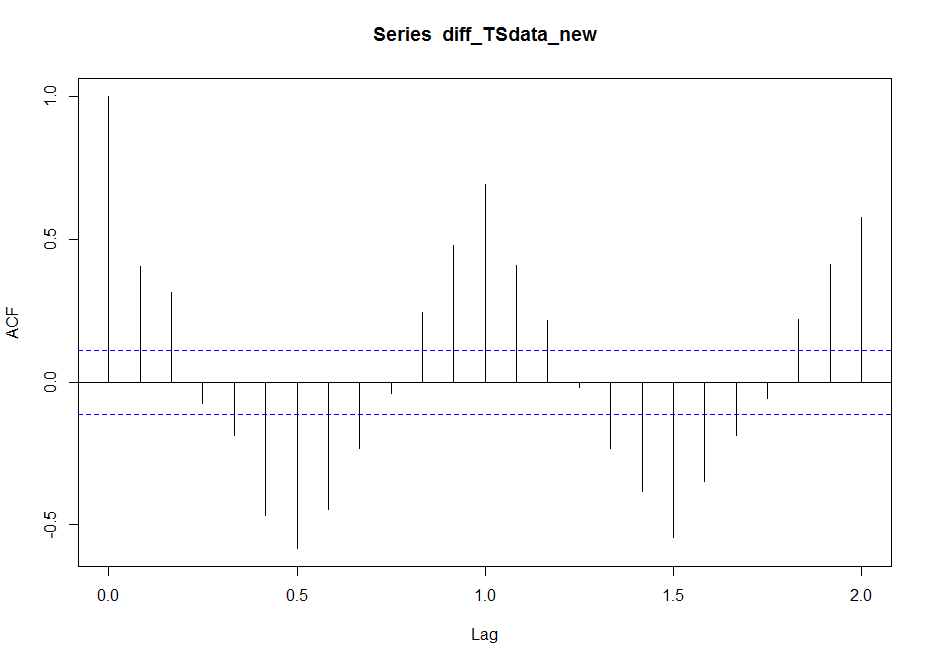
series appears roughly constant over time.

**### ACF and PACF (performing to check the stationary data and autocorrelation)**

The function Acf computes an estimate of the autocorrelation function of a (possibly multivariate) time series.

Function Pacf computes an estimate of the partial autocorrelation function of a (possibly multivariate) time series.

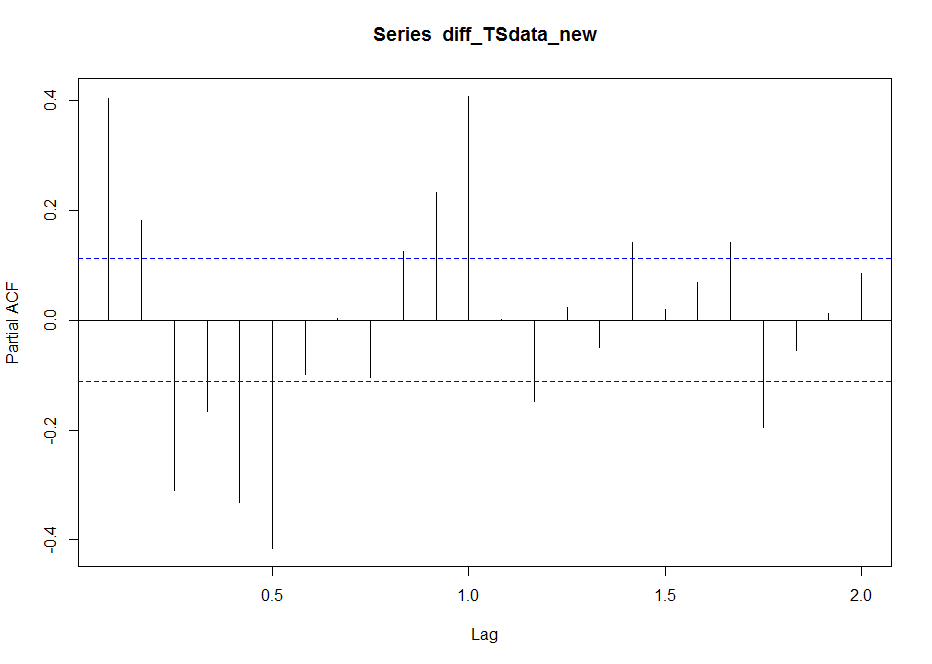
acf(diff\_TSdata\_new)



From the above graph, looks like autocorrelation is significant over all lags except for lag 4 and lag 10, in first ten lags. ACF plots can help in determining the order of the MA (q) model.

**### To confirm if the correlation in the later lags is not because of mutual correlation, let’s draw the partial correlation plot**

pacf(diff\_TSdata\_new)



Even the partial correlation plot shows that all lags are significant.

PACF plots are useful when determining the order of the AR(p) model.

As multiple lags are significant, it is not possible to tentatively identify the numbers of AR and/or MA terms that are needed.

**### Hence let’s start with Auto Arima model.**

Model Development

Auto Arima

* Exponential smoothing methods are useful for making forecasts, and make no assumptions about the correlations between successive values of the time series.
* While exponential smoothing methods do not make any assumptions about correlations between successive values of the time series, in some cases you can make a better predictive model by taking correlations in the data into account.

Autoregressive Integrated Moving Average (ARIMA) models include an explicit statistical model for the irregular component of a time series, that allows for non-zero autocorrelations in the irregular component. ARIMA models are defined for stationary time series.

**As the series has seasonality, hence in the auto arima function seasonality is assumed to be true.**

TSdat.arima.fit.train <- auto.arima(DataATrain, seasonal=TRUE)

TSdat.arima.fit.train

Akaike information criteria (AIC) and Baysian information criteria (BIC). are closely related and can be interpreted as an estimate of how much information would be lost if a given model is chosen.

When comparing models, one wants to minimize AIC and BIC. ARIMA(1,1,1)(0,1,1)[12] is seasonal ARIMA. [12] stands for number of periods in season, i.e. months in year in this case. (1,0,0) stands for seasonal part of model.

This model had first order autoregressive nonseasonal, first order difference nonseasonal , first order moving average nonseasonal, first order seasonal difference term, and first order seasonal moving average terms with a lag component of twelve months.

**### Arima**

# Let’s fit the autoarima output using the Arima function

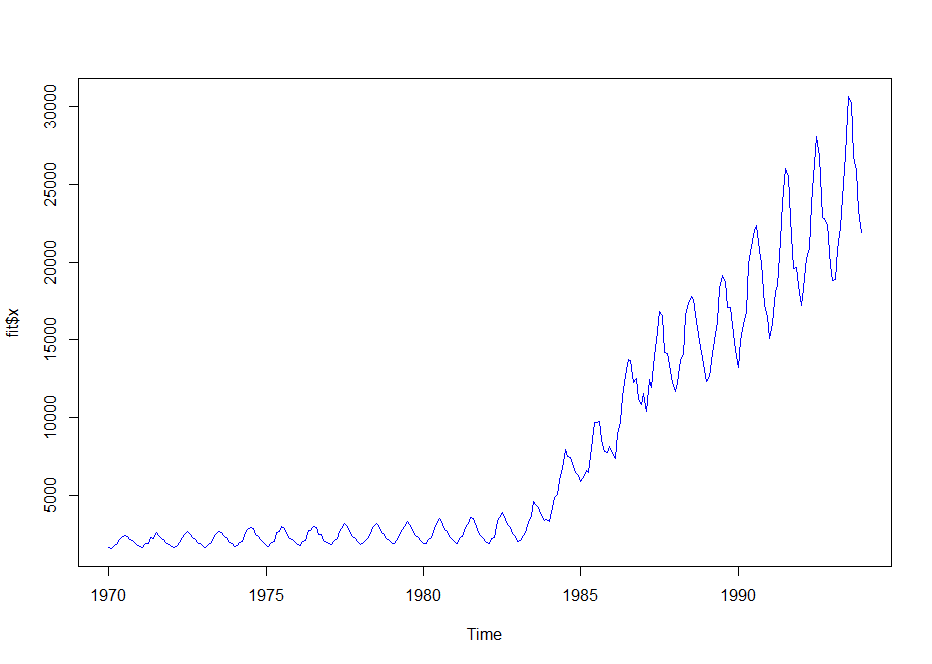
fit <- Arima(DataATrain, c(1, 1, 1),seasonal = list(order = c(0, 1, 1), period = 12))

fit

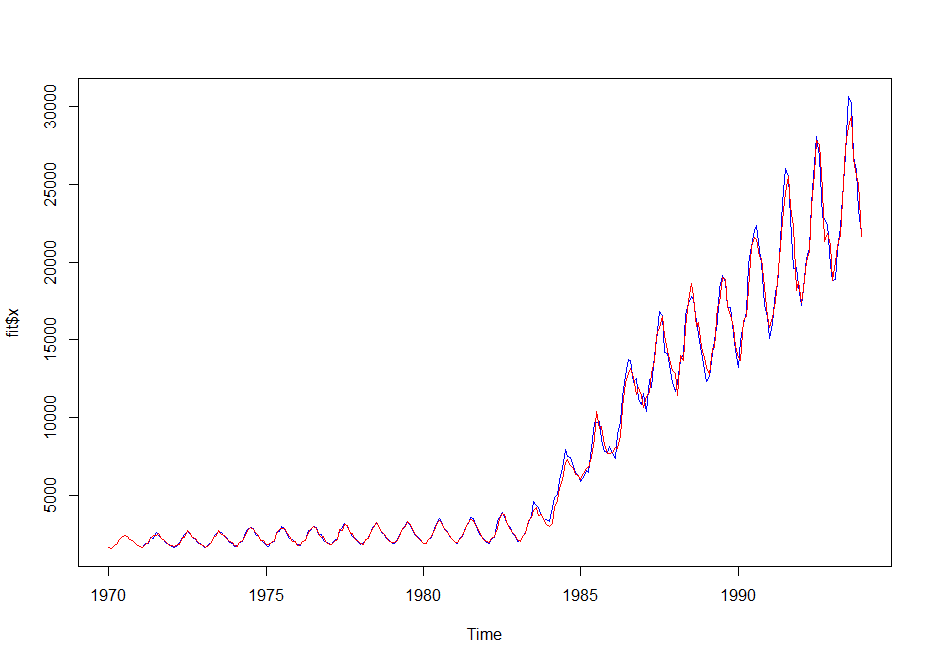
arima(x = DataATrain, order = c(1, 1, 1), seasonal = list(order = c(0, 1, 1),

period = 12))

plot(fit$x,col="blue")



lines(fit$fitted,col="red",main="Production : Actual vs Forecast")



**### Box-Ljung Test**

* To check is residual are independent
* H0: Residuals are independent
* Ha: Residuals are not independent

Box.test(fit$residuals, type = c("Ljung-Box"))

Conclusion: Do not reject H0: Residuals are independent

**# Now the model is valid,**

**# let’s check the model performance on the train dataset**

VecA1 <- cbind(fit$fitted,fit$x)

MAPEA\_train <- mean(abs(VecA1[,1]-VecA1[,2])/VecA1[,1])

MAPEA\_train

# Let’s forecast the holdout sample using the above model.

# Period is considered as 20 because we have 20 periods in the holdout sample

Arimafcast <- forecast(fit, h=20)

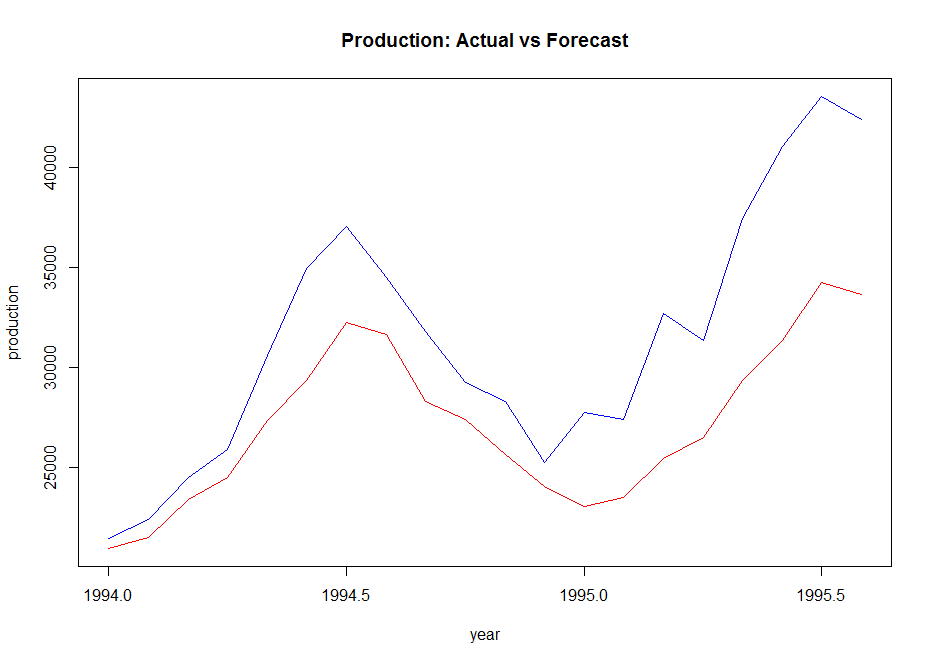
VecA2 <- cbind(DataATest,Arimafcast)

MAPEA\_holdout <- mean(abs(VecA2[,1]-VecA2[,2])/VecA2[,1])

MAPEA\_holdout

ts.plot(VecA2[,1],VecA2[,2], col=c("blue","red"),xlab="year", ylab="production",

main="Production: Actual vs Forecast")



From the above plot it can be seen that there is a difference between the actual and forecast model hence there is a scope to further improve the model.

Some tips to improve the model are provided in section 4 of this document.

**# Final Model**

In time series, model creation is a two-step process.

In the first step, data is divided into train and holdout sample.

Model is prepared using the train dataset and validated using the holdout sample.

Once the model is finalised, then the final model is prepared on the complete dataset.

Output of this model is used for a real forecast i.e. forecast for an unknown period.

Final\_model <- auto.arima(TSdata\_new, seasonal=TRUE)

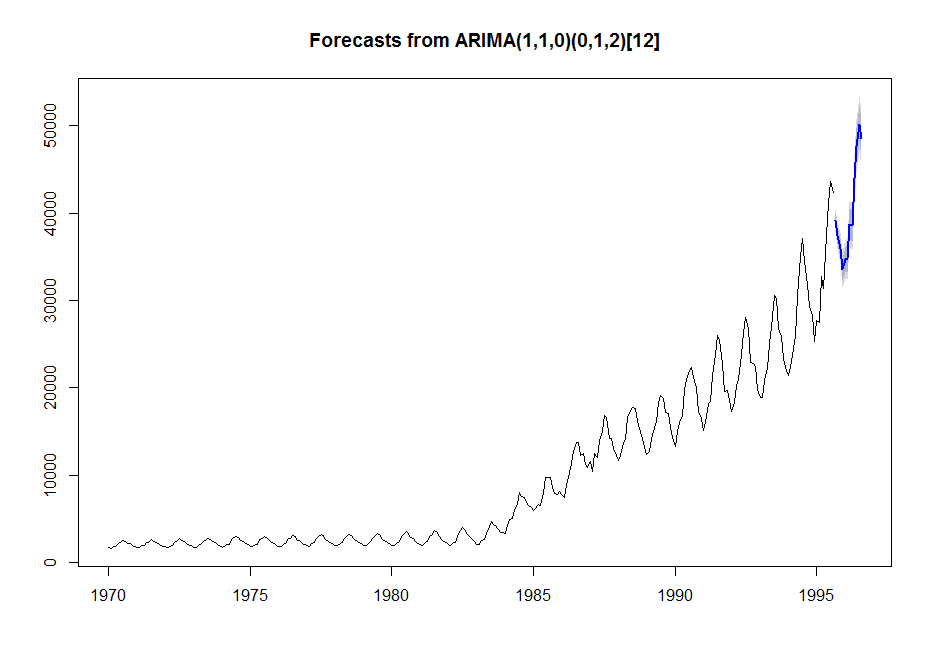
Final\_model

Box.test(Final\_model$residuals, type = c("Ljung-Box"))

**### Now forecast for 12 months, unknown period.**

Final\_forecast <- forecast(Final\_model, h=12)

plot(Final\_forecast)



**#### Phase IV Next Steps for Model Refining**

* Now we have built a robust model. Can this model be refined further?

Following steps are recommended to refine the model further

✓ Log transformation of the original dataset as log transformed data is expected to give

better result

✓ As it is clearly visible in the final model, Arima model equation is different when

the entire dataset was considered for the final model, which clearly indicates that

dataset for the later periods has seasonality and trend which should be further

analysed. Hence rather than considering dataset from 1970, probably a refined model

should be built considering the period from 1980.