05-06-2020

TIME SERIES PROJECT

AUSTRALIAN MONTHLY GAS PRODUCTION FORECAST

PRESENTED BY: SHILPA GIRIDHAR

CONTENTS

DESCRIPTION

Problem Statement

For this assignment, you are requested to download the **Forecast** package in R. The package contains methods and tools for displaying and analyzing univariate time series forecasts including exponential smoothing via state space models and automatic ARIMA modelling. Explore the "**gas (Australian monthly gas production)"** dataset in Forecast package to do the following:

[Hint code]

install.packages("forecast")

library(forecast)

data<- forecast::gas

Requirements

Perform the following :

1. **EDA (15 Marks)**
   - Read the data as a time series object in R. Plot the data (5 marks)
   - What do you observe? Which components of the time series are present in this dataset? (5 marks)
   - What is the periodicity of dataset? (5 marks)
     - HINT: Please use the dataset from January 1970 for your analysis.
     - Please partition your dataset in such a way that you have the data 1994 onwards in the test data.

2. **Modelling and Forecasting (40 Marks)**
   - 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? (15 marks)
   - Develop an initial forecast for next 20 periods. Check the same using the various metrics, after finalizing the model, develop a final forecast for the 12 time periods. Use both manual and auto.arima (Show & explain all the steps) (20 marks)

3. **Report the accuracy of the model (5 marks)**
   - Summarize your findings from the exercise in a concise yet actionable note

DATA EXPLORATION

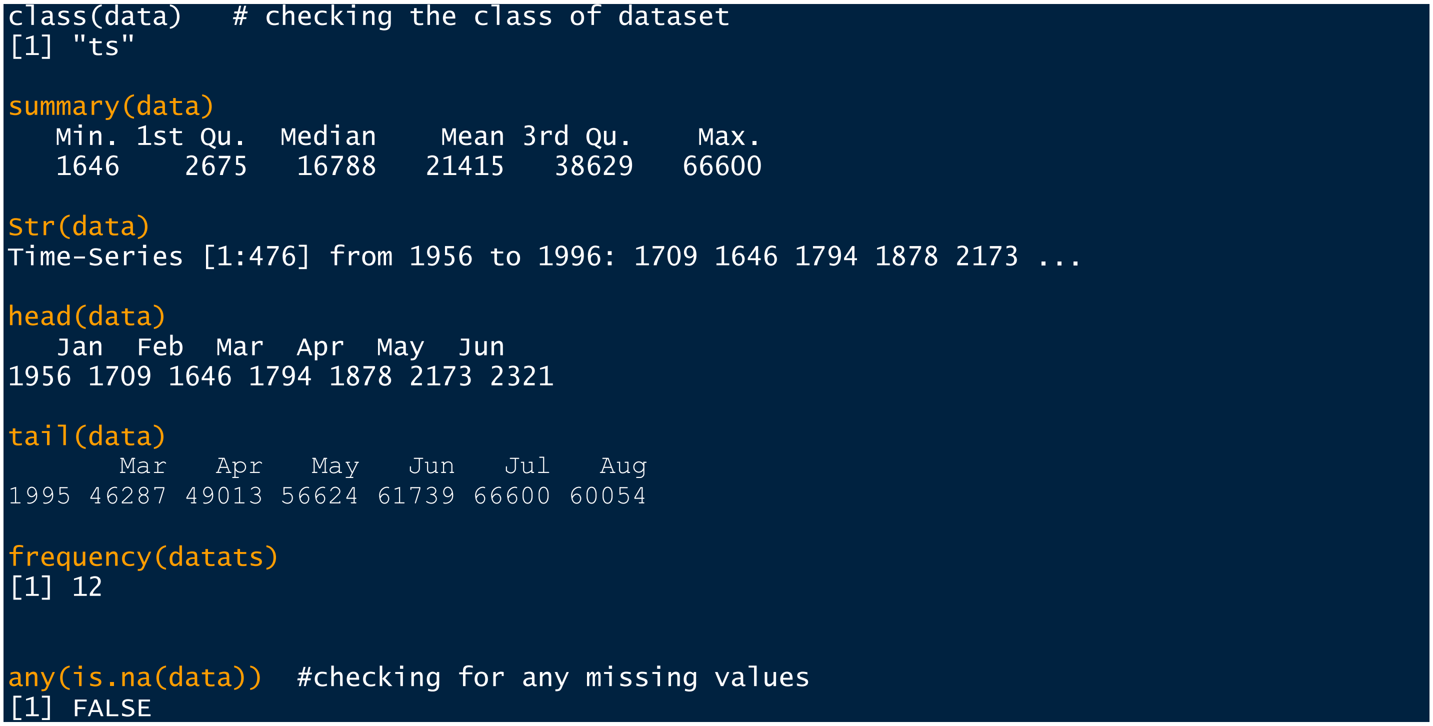
Reading Data, Finding Periodicity of Time Series Dataset

- The data shows that it’s a timeseries type dataset, there are 476 observations from year 1956 to 1996 and production ranging from 1660 to 66600.

- Performed the str and summary function.

- We can observe that there are no missing data.

- We also observe that the dataset has “Year” as row and “Months” as columns. We can run the frequency command and conclude that periodicity or the cyclicity is 12.

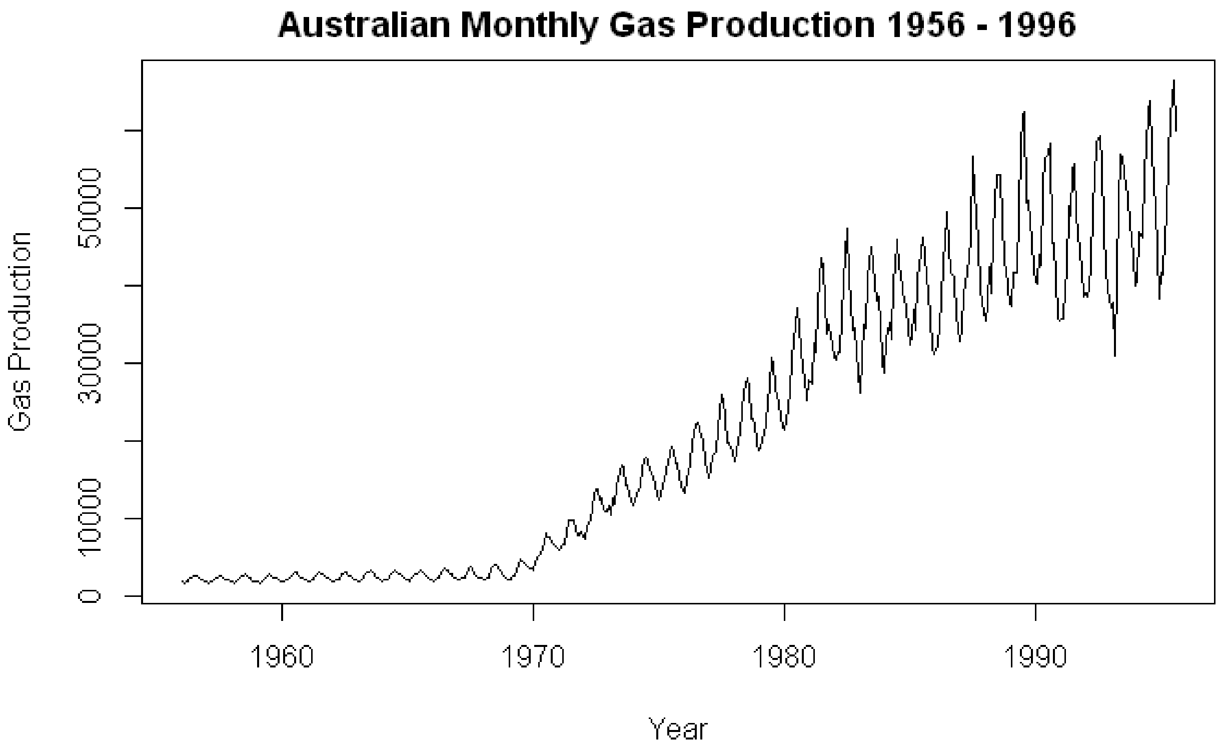


```
class(data)   # checking the class of dataset
[1] "ts"

summary(data)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1646    2675   16788   21415   38629   66600

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

head(data)
   Jan  Feb  Mar  Apr  May  Jun
1956 1709 1646 1794 1878 2173 2321

tail(data)
       Mar   Apr   May   Jun   Jul   Aug
1995 46287 49013 56624 61739 66600 60054

frequency(datats)
[1] 12


any(is.na(data))  #checking for any missing values
[1] FALSE
```

Finding Components – Stationery, Trend, and Seasonality

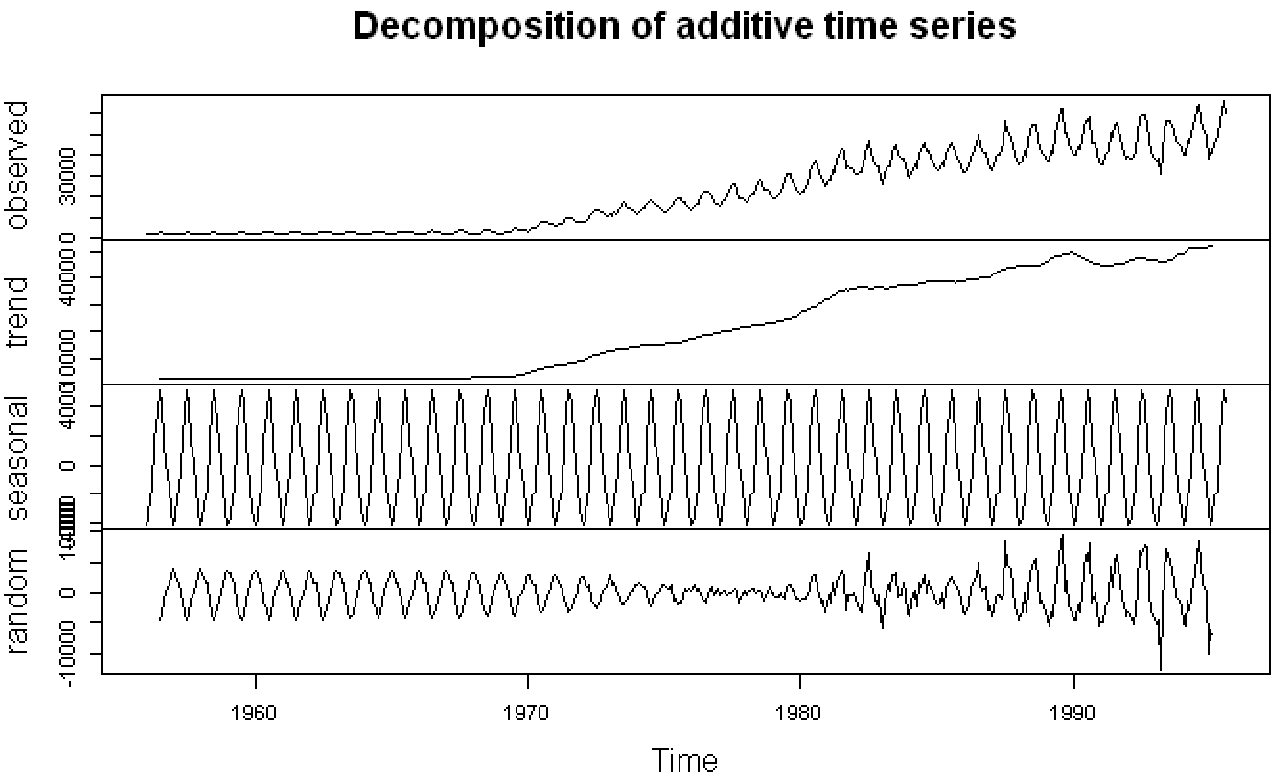
- Plotted timeseries data shows that there is change in trend from the year 1970 onwards and that the gas production has gradually increased from year 1970 onwards. The data has mostly a flat trend prior to yar 1970. Hence for practical forecast purposes we use filtered data from year 1970 to 1996.

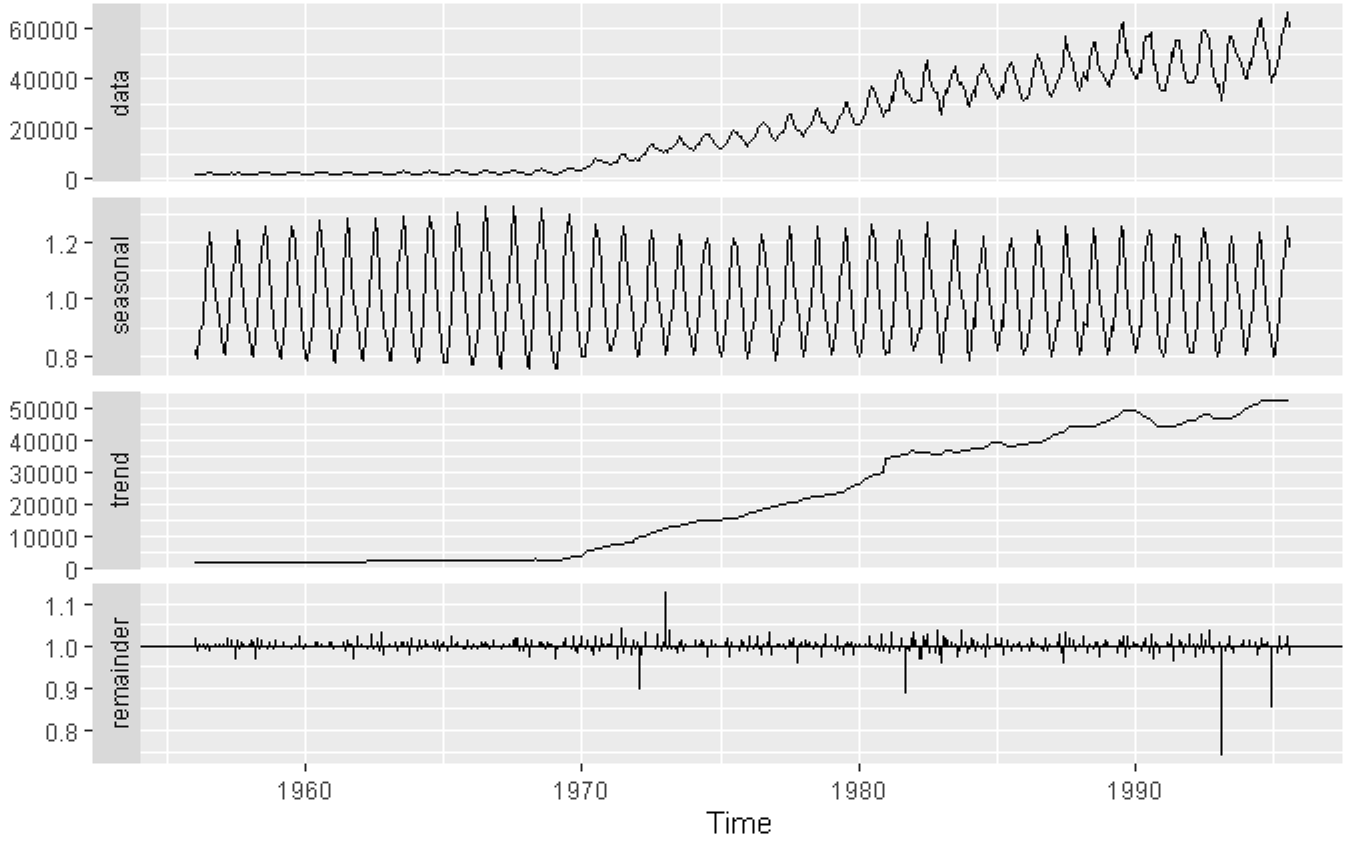
```
ts.plot(data,xlab = "Year", ylab= "Gas Production",
                main = " Australian Monthly Gas Production 1956 - 1996 ")
```



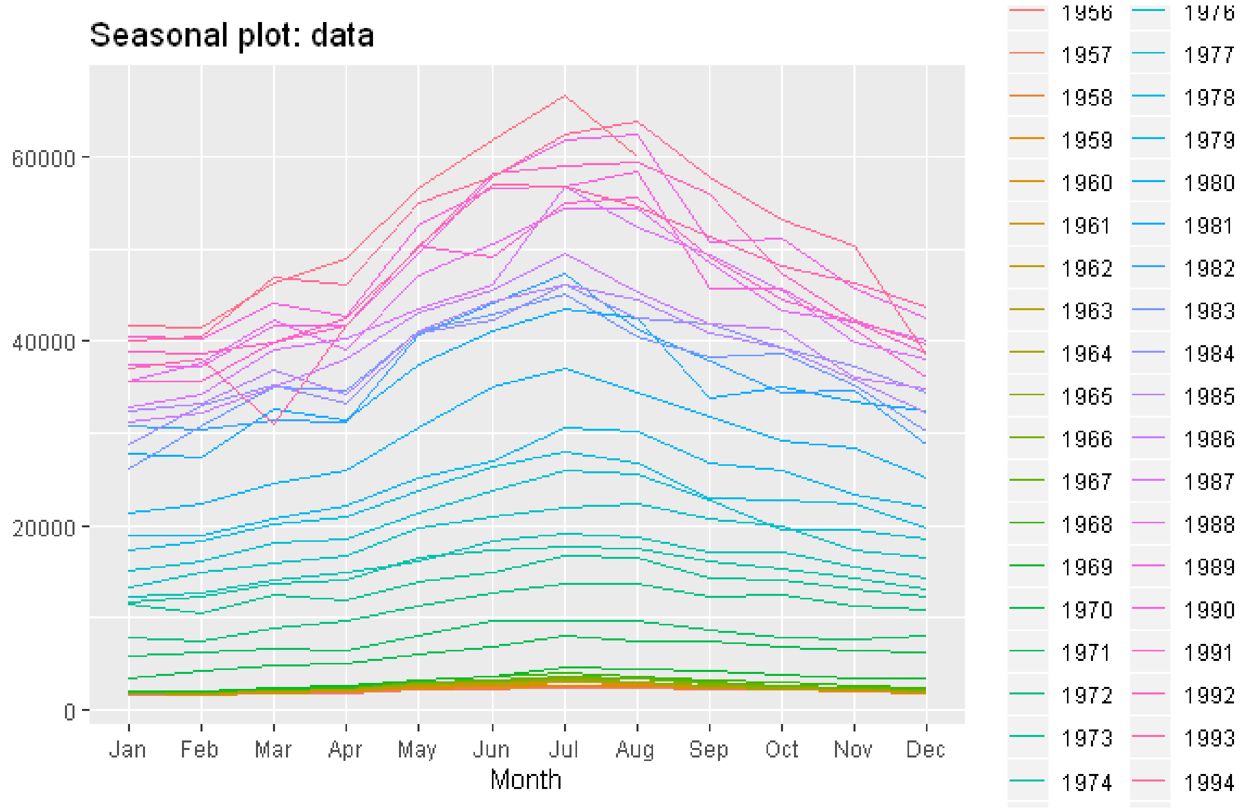
- We can see from this time series that there seems to be seasonal variation in the number of productions per month: there is a crest (or peaks) and trough in the cycle every year. To estimate the trend, seasonal and irregular components of this time series, we decompose the timeseries and plot it. We can also use “autoplot” function from “seasonal” library package which does similar plotting of the components

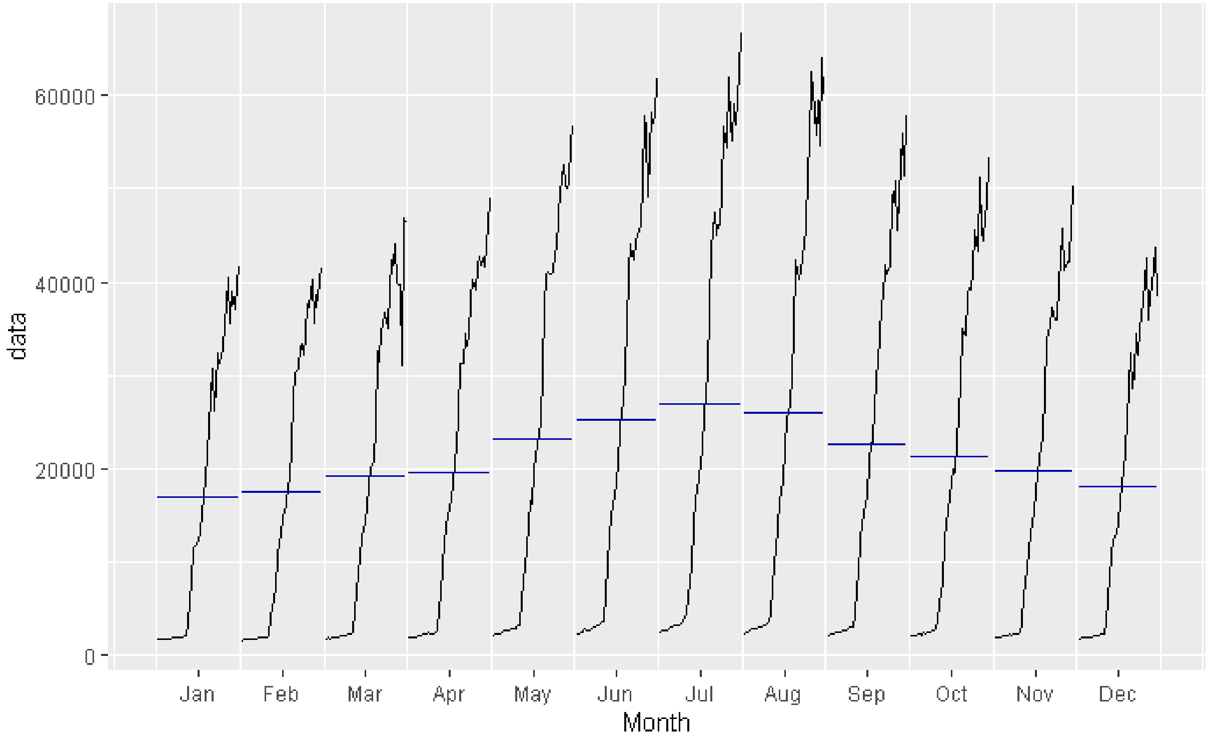
```
#estimate the trend, seasonal and irregular components of this time series
data_components <- decompose(data)
 plot(data_components)

 #using library(seasonal)
data %>% stl(s.window='periodic') %>% autoplot
seas(data) %>% autoplot
```


Decomposition of additive time series
observed
30000
0
trend
400000
10000
seasonal
4000
0
random
10000
0
-10000
1960
1970
1980
1990
Time


data
60000
40000
20000
0
seasonal
1.2
1.0
0.8
trend
50000
40000
30000
20000
10000
0
remainder
1.1
1.0
0.9
0.8
1960
1970
1980
1990
Time

- Using ggseasonplot and ggsubseriesplot functions we can plot a seasonal plot to observe seasonality in the dataset. This is a time plot except that the data are plotted against the seasons in separate years.





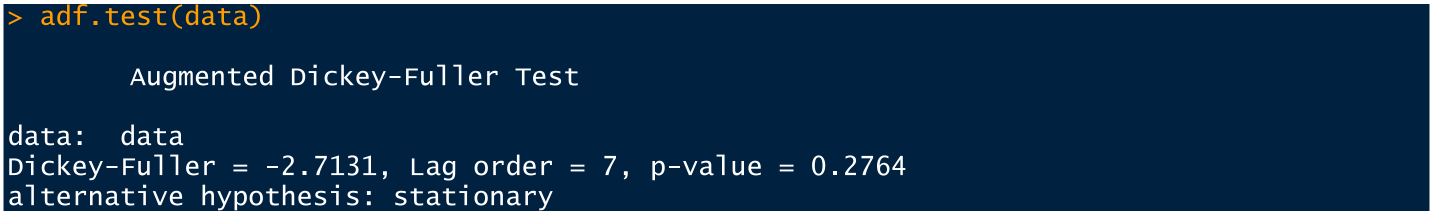
- **Check for Stationary** - Visual observations of decomposed plot shows that both trend and seasonality exists in this dataset. A stationary time series is one whose properties do not depend on the time at which the series is observed. Thus, time series with trends, or with seasonality, is **Not Stationary**.

- We can confirm the same by performing **Augmented Dickey-Fuller** test to check if the time series Stationary

  Null hypothesis Ho for ADF test: Time series non-stationary

  Alternative hypothesis Ha for ADF test: Time series is stationary

- Results show that p-value is greater than significant value of 0.05, hence Null Hypothesis is accepted. Thus with ADF test also, we can conclude that time series dataset is **Not Stationary.**



```
> adf.test(data)

	Augmented Dickey-Fuller Test

data:  data
Dickey-Fuller = -2.7131, Lag order = 7, p-value = 0.2764
alternative hypothesis: stationary
```

DATA PREPARATION

Filtering, Decomposing, Deseasoning

- Plotted timeseries data shows that there is change in trend from the year 1970 onwards and that the gas production has gradually increased from year 1970 onwards. The data has mostly a flat trend prior to yar 1970.
- For forecast purposes we use filtered data from year 1970 to 1996. The filtered series has 301 observations from 1971 to 1996.

```
#filter time series data
> datats <- ts(data, start=c(1970,12), end=c(1995,12), frequency=12)

> summary(datats)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1646    2279    3239    8794   14989   37060

> str(datats)
Time-Series [1:301] from 1971 to 1996: 1709 1646 1794 1878 2173 ...)
```

- To estimate the trend, seasonal and irregular components of this time series, we decompose the filtered timeseries and plot it. We can also use “autoplot” function from “seasonal” library package which does similar plotting of the components.

Additive Model: Observation= Trend + Seasonality + Error; Useful when seasonality variation is relatively constant over the time.

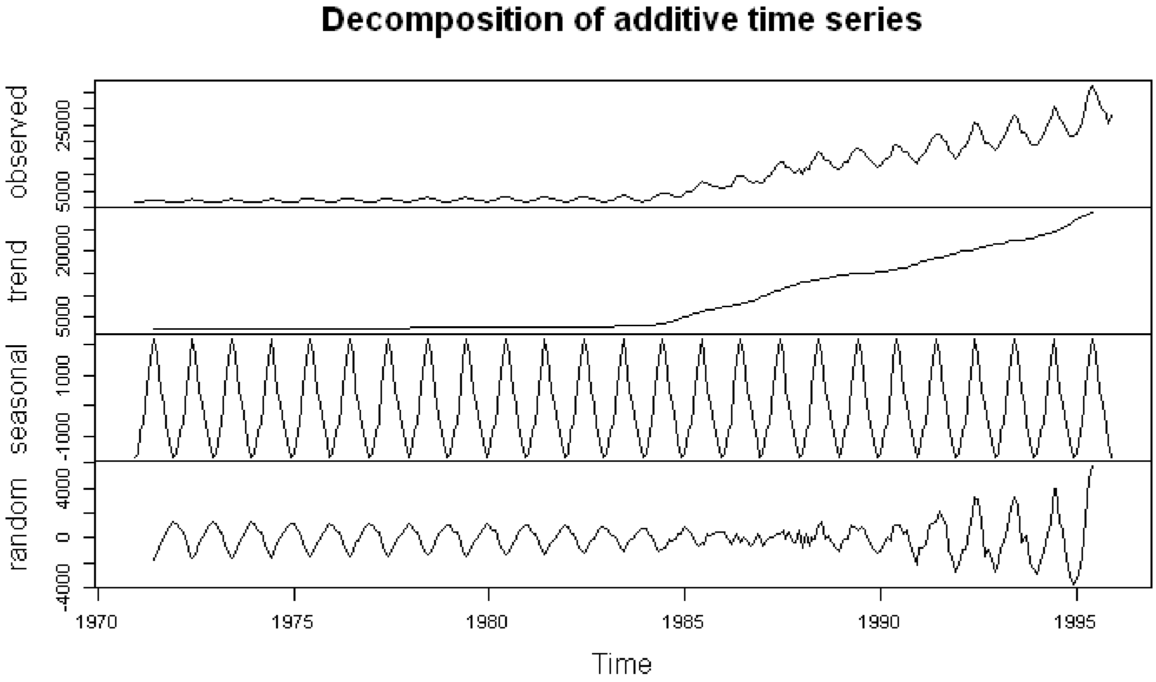
```
#estimate the trend, seasonal and irregular components of this time series
 datats_components <- decompose(datats, type = "additive")
 plot(datats_components)

#using library(seasonal)
 datats %>% stl(s.window='periodic') %>% autoplot
 seas(datats) %>% autoplot
```

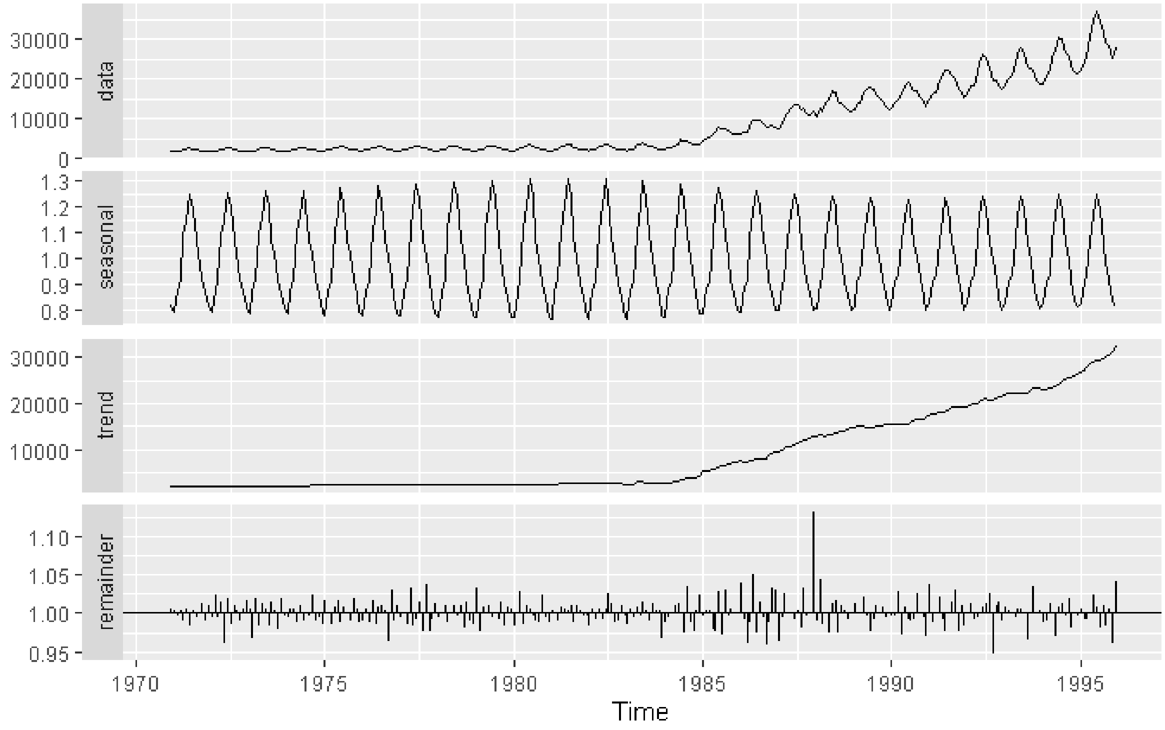
- Graph shows trend and seasonality exists. On performing the ADF test on filtered data set confirms it.

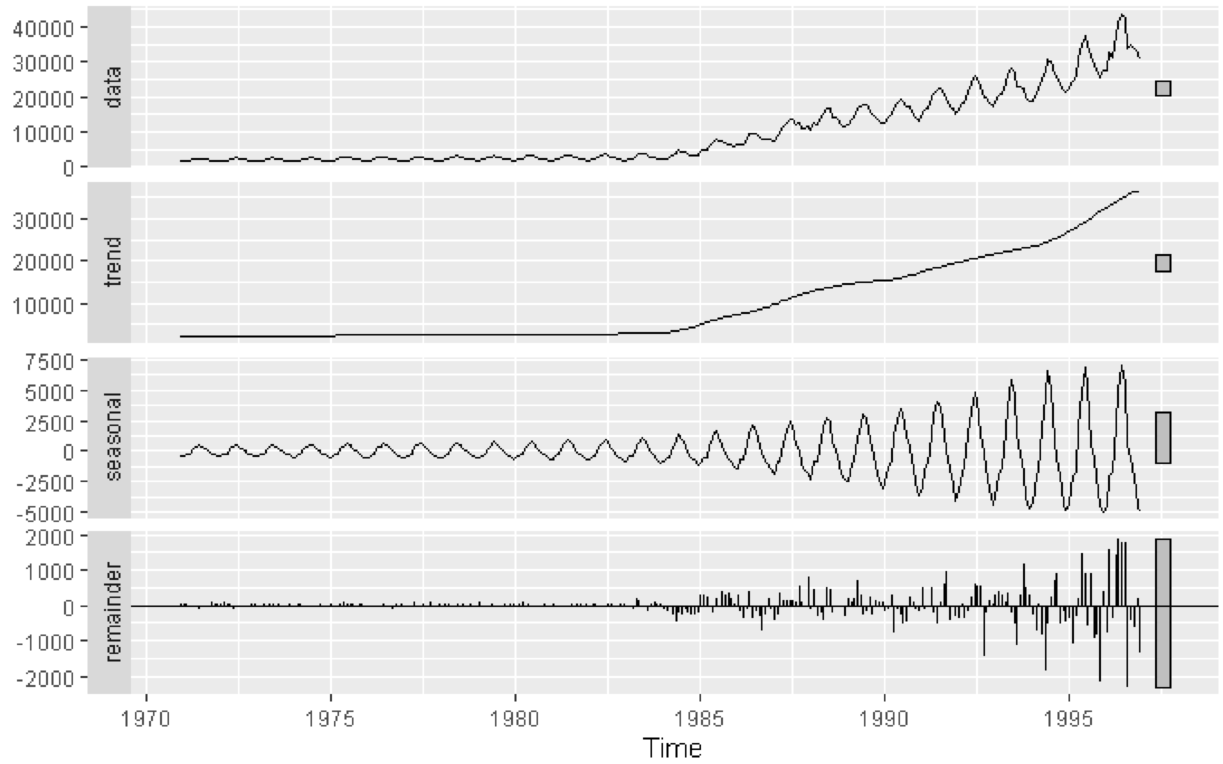
```
    Augmented Dickey-Fuller Test

data:  datats
Dickey-Fuller = 0.45786, Lag order = 6, p-value = 0.99
alternative hypothesis: stationary
```

We get following components from decomposition:

- Observed – the actual data plot
- Trend – the overall upward or downward movement of the data points
- Seasonal – any monthly/yearly pattern of the data points
- Random – unexplainable part of the data

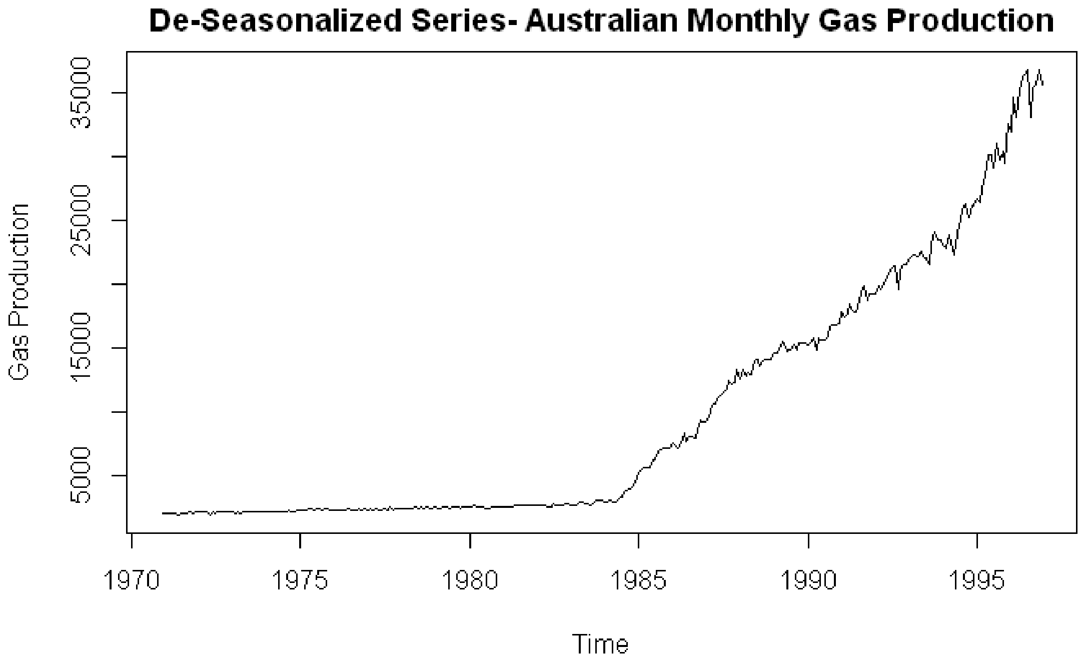




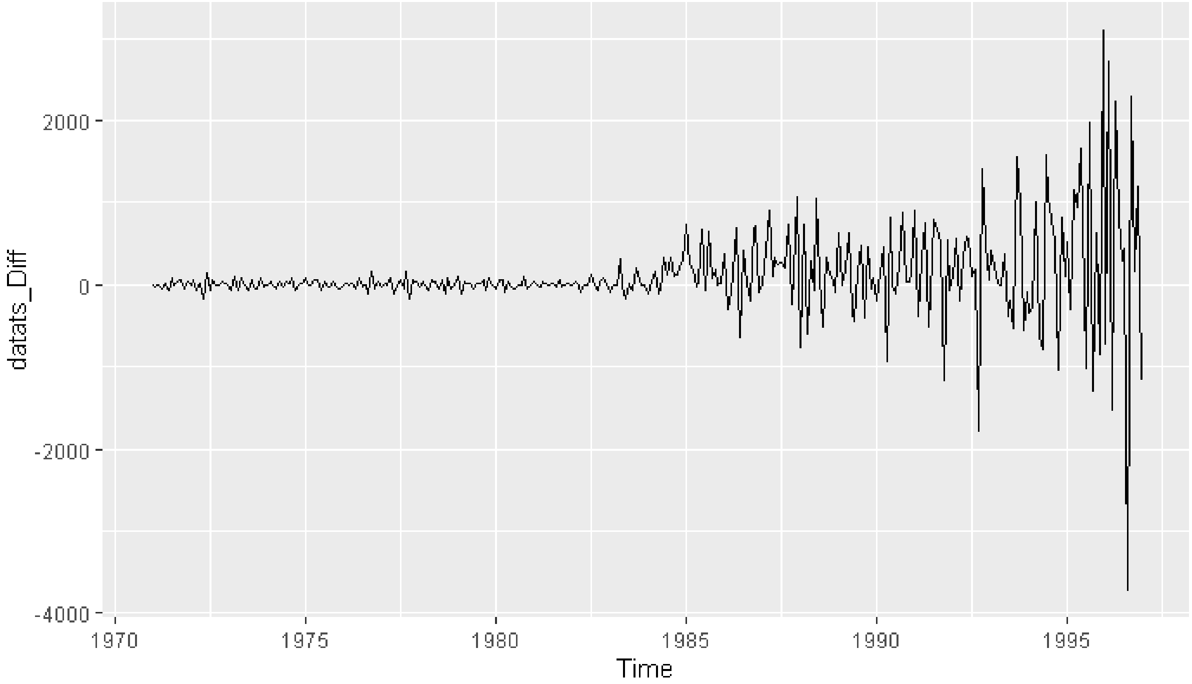


- Seasonal Adjustment – removing seasonality component; The seasonal part can be removed from the analysis and added later

```
datats_2_ds = seasadj(datats_1)
plot(datats_2_ds, ylab= "Gas Production", main = "De-Seasonalized Series- Australian
Monthly Gas Production") ## plotting the de-seasonalized data
```



- Seasonal Adjustment – removing seasonality component; The seasonal part can be removed from the analysis and added later



MAKING TIME SERIES STATIONARY AND NON-SEASONAL

[ref: https://datascienceplus.com/time-series-analysis-using-arima-model-in-r/]

- Before Forecasting method we will need to split deseasonalized dataset into test and train data
- One way to convert a non-stationary time series to a stationary time series (detrend the data) is by computing the differences between consecutive observations, also known as **Differencing**. We use
- If the data contains a trend, detrend the data
- The seasonal part can be removed from the analysis and added later, or it can be taken care of in the ARIMA model itself. To remove seasonality from the data, we subtract the seasonal component from the original series and then difference it to make it stationary. “ndiffs()” is used to determine the number of first differences required to make the time series non-seasonal

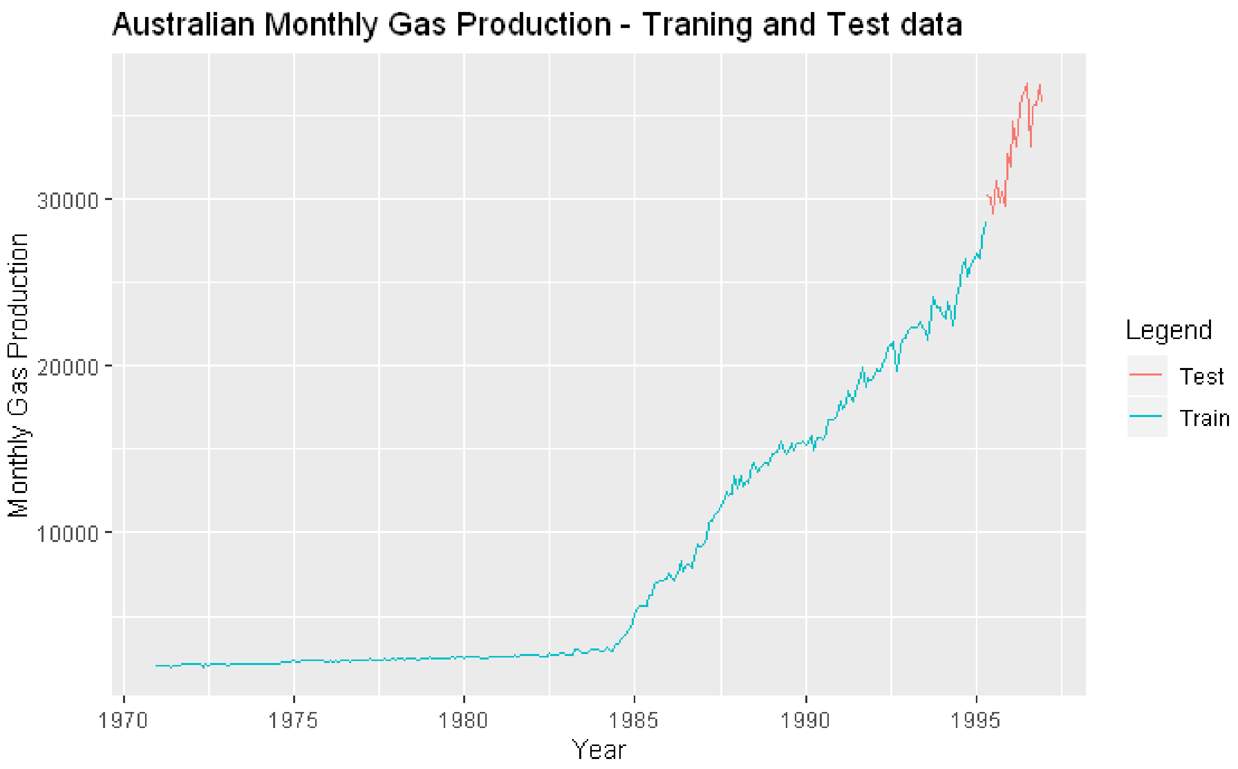
FORECAST

1. ACF and PACF
2. ARIMA and AUTO ARIMA

To forecast, first we need to De-seasonalize the series, split the data into train and test data, and then forecasted values needs to be estimated and adjusted by seasonality.

SPLITTING DE-SEASONALIZED DATA INTO TRAIN AND TEST DATA

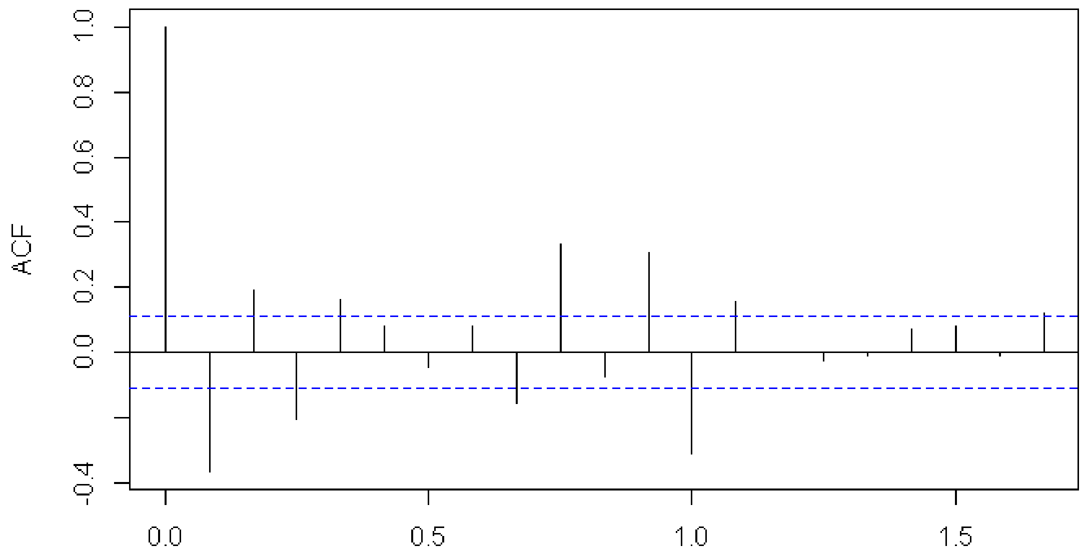
```
#prepare training data and test data on DESEASONALIZED DATA
datatsTrain <- window(datats_2_ds, start=c(1971,1), end=c(1995,4),frequency=12)
datatsTest <- window(datats_2_ds, start=c(1995,5),end=c(1996,12),frequency=12)
#.(last 4 months of 1995+8 months of 1996).


str(datatsTrain)
str(datatsTest)
head(datatsTest)
tail(datatsTest)

## Plotting the train and Test set
autoplot(datatsTrain, series="Train") +
 autolayer(datatsTest, series="Test") +
 ggtitle("Australian Monthly Gas Production - Traning and Test data") +
 xlab("Year") + ylab("Monthly Gas Production") +
 guides(colour=guide_legend(title="Legend"))
```

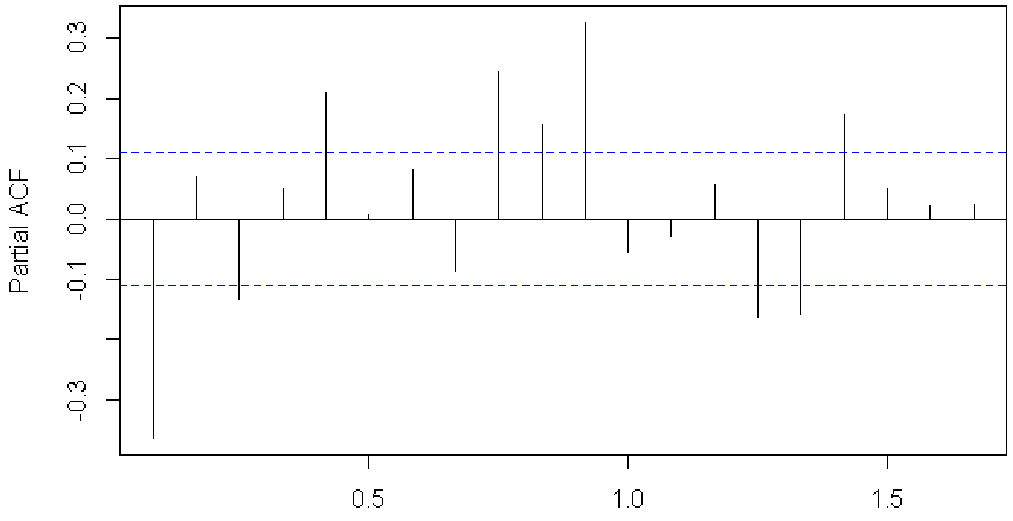


ACF AND PACF

- ACF plot describes how well the present value of the series is related with its past values. PACF is a partial auto-correlation function that finds correlation of the residuals by the earlier lag(s)) with the next lag value hence 'partial'.
- Auto Correlation Function (ACF) and Partial ACF plots on stationary data reveals which p and q values will be appropriate to fit model. ACF shows few lags, but overall series decaying to zero.
- Order q=3 is obtained from the ACF plot, this is the lag after which ACF crosses the upper confidence interval for the first time. Order p=2 is the lag value after which PACF plot crosses the upper confidence interval for the first time.

```
#Run correlation on stationary data
acf(datats_stationary1, lag.max=50)
pacf(datats_stationary1, lag.max=50)
```

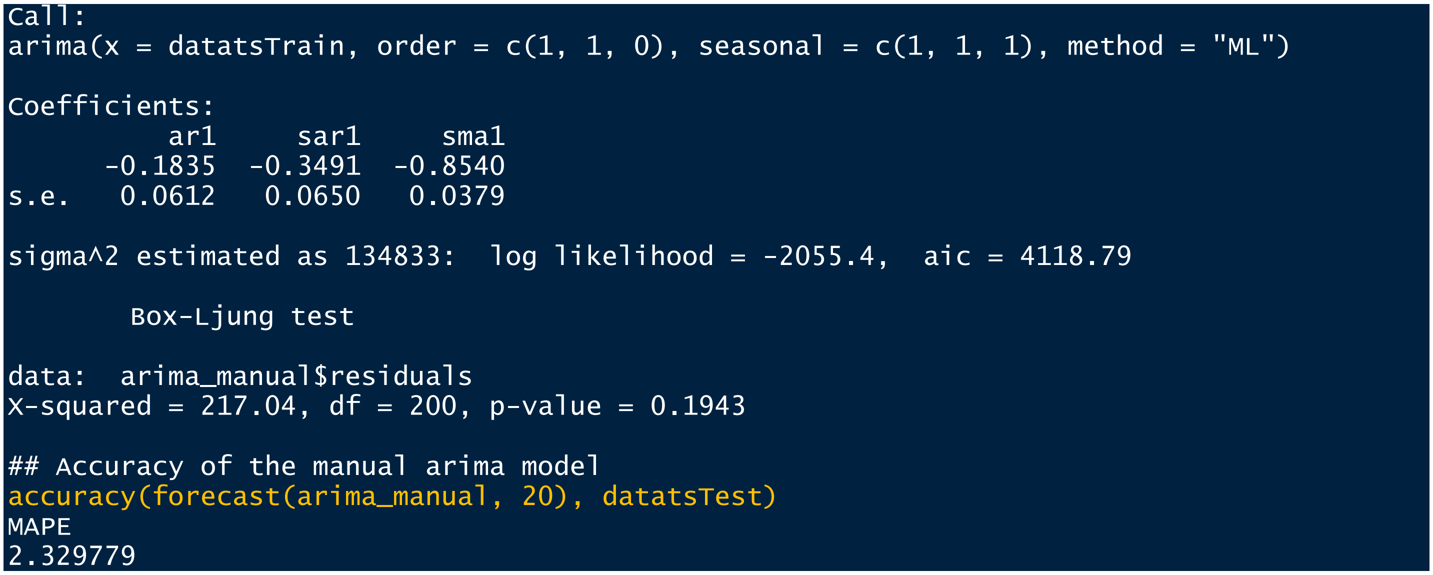




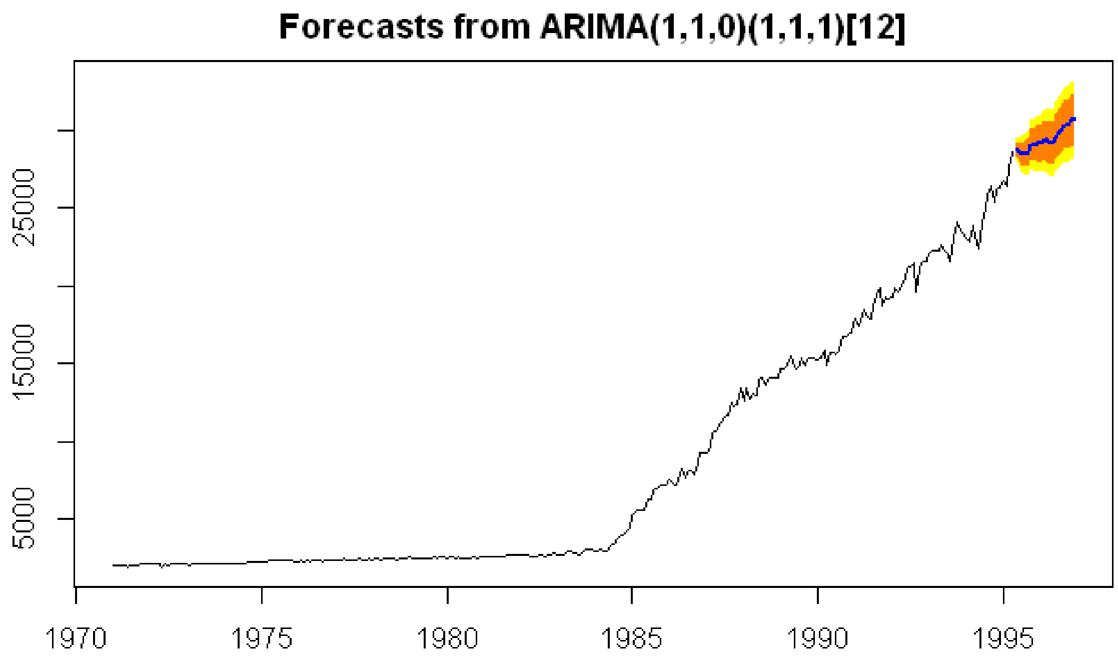
ARIMA AND AUTO ARIMA

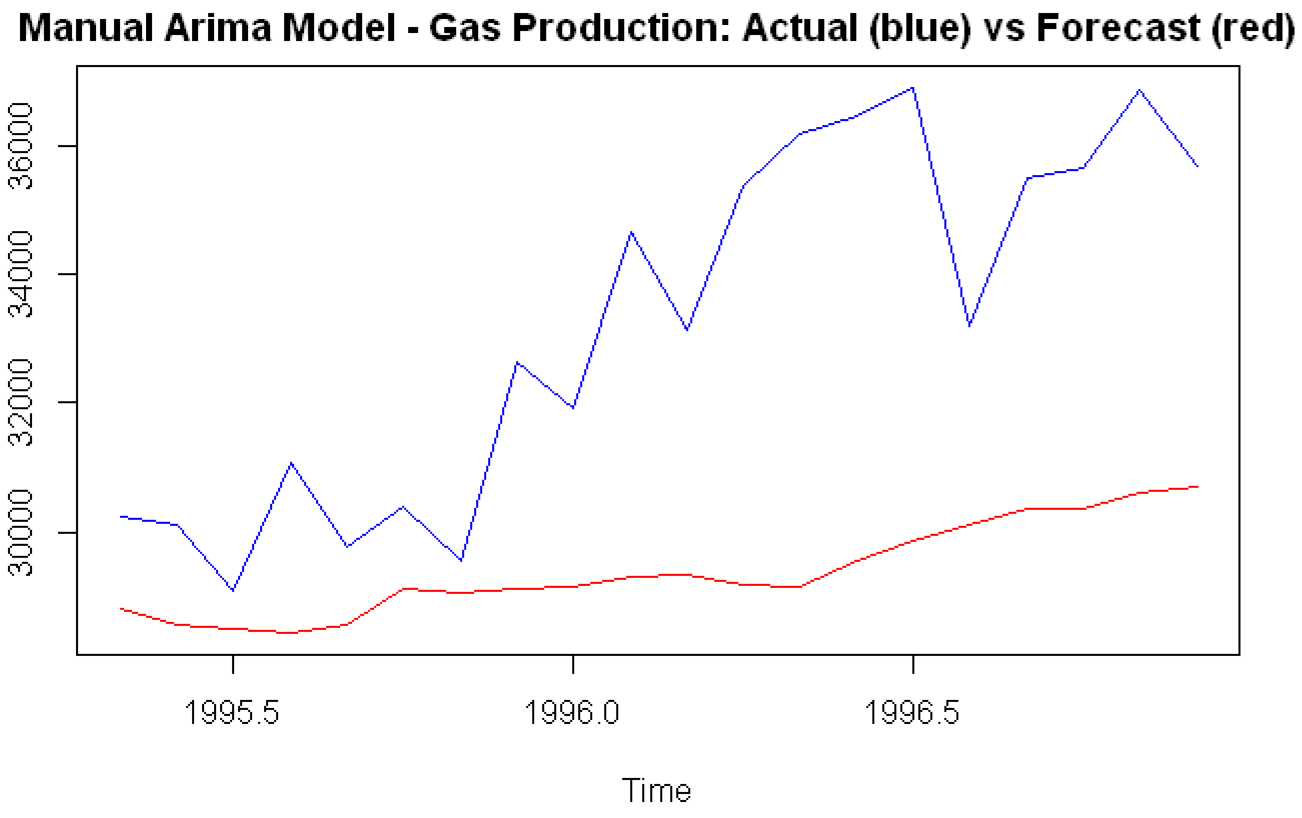
- ARIMA models are defined for stationary and univariate time series. Since the dataset is not stationary, we have to determine the number of differences required for time series to be made stationary.
- Order refer to values of [p,d,q]- p, d, and q which are non-negative integers that refer to the order of the autoregressive, integrated, and moving average parts of the model respectively. Method refer to "Maximum Likelihood".
- p= number of non-seasonal autoregressive terms, d= number of non-seasonal differences, q= number of non-seasonal moving average terms; and P=number of seasonal autoregressive (SAR) terms, D=number of seasonal differences, Q=number of seasonal moving average (SMA) terms
  The objective is to minimize the MAPE and AIC values. We will examine ARIMA model with different values.

- Manual ARIMA MODEL with [p,d,q] = [1,1,0] provides AIC value = 4118 and MAPE = 2.3;

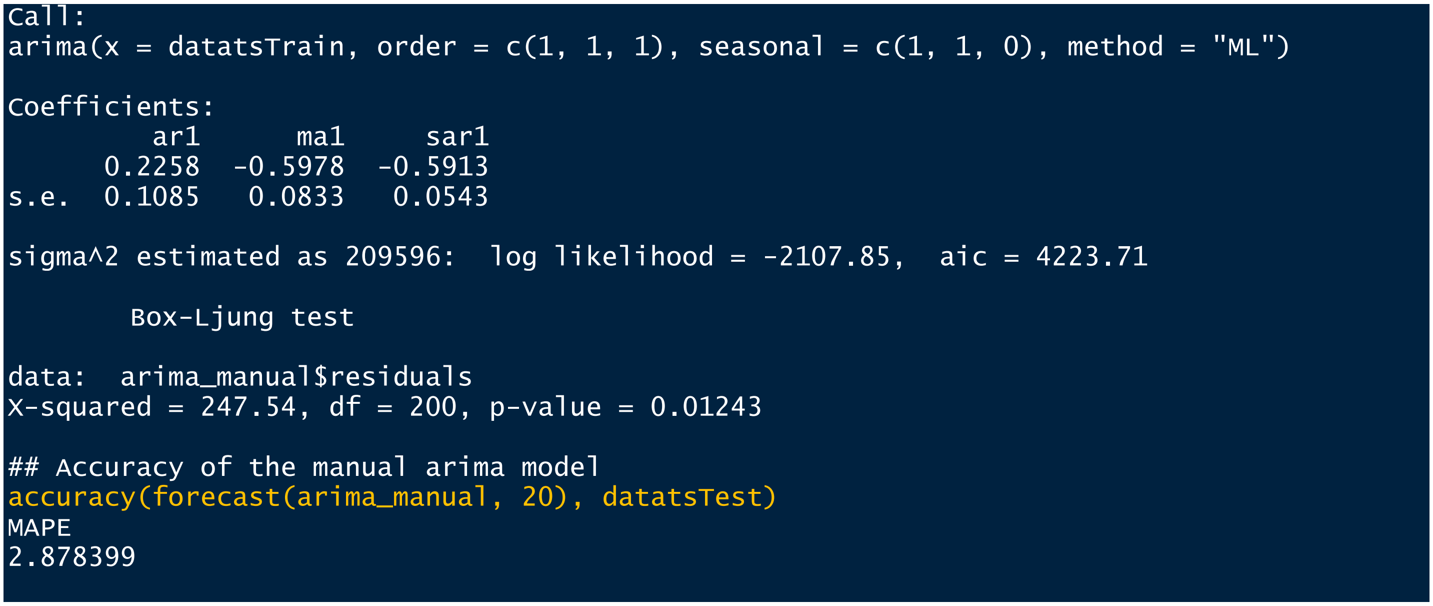


```
Call:
arima(x = datatsTrain, order = c(1, 1, 0), seasonal = c(1, 1, 1), method = "ML")

Coefficients:
          ar1     sar1     sma1
      -0.1835  -0.3491  -0.8540
s.e.   0.0612   0.0650   0.0379

sigma^2 estimated as 134833:  log likelihood = -2055.4,  aic = 4118.79

        Box-Ljung test

data:  arima_manual$residuals
X-squared = 217.04, df = 200, p-value = 0.1943

## Accuracy of the manual arima model
accuracy(forecast(arima_manual, 20), datatsTest)
MAPE
2.329779
```

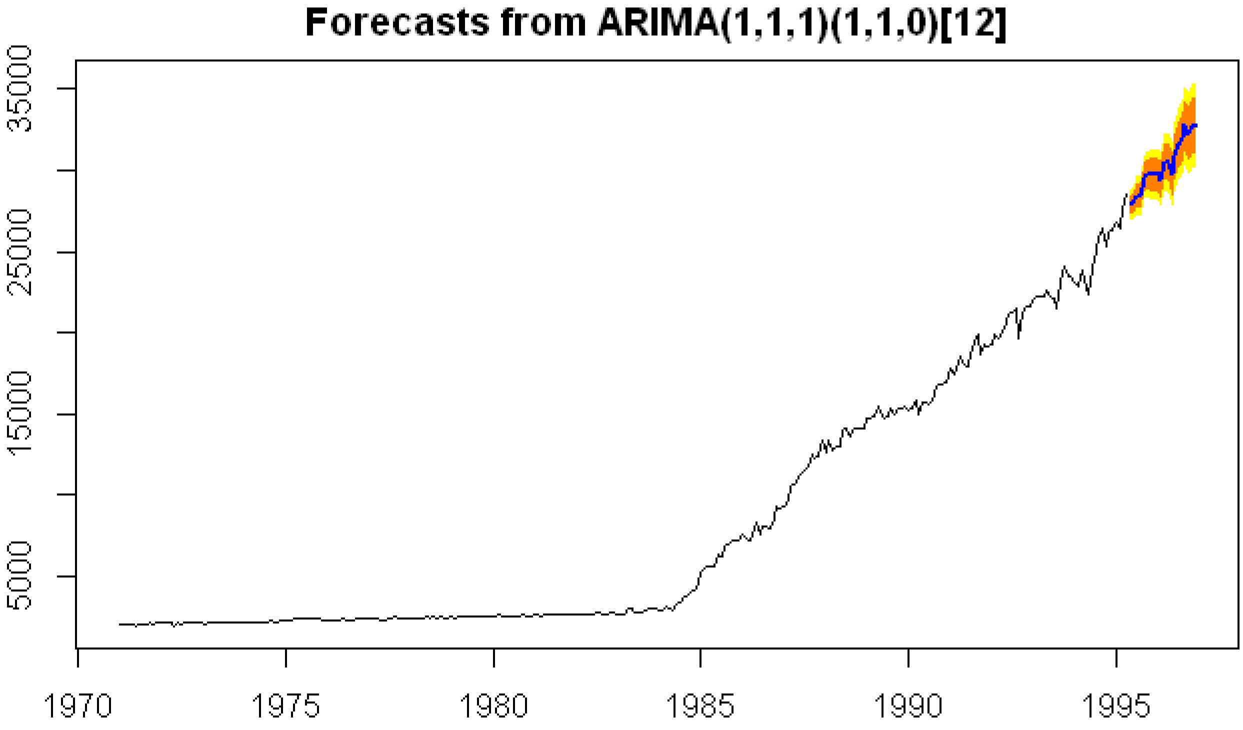

Forecasts from ARIMA(1,1,0)(1,1,1)[12]
5000
15000
25000
1970
1975
1980
1985
1990
1995

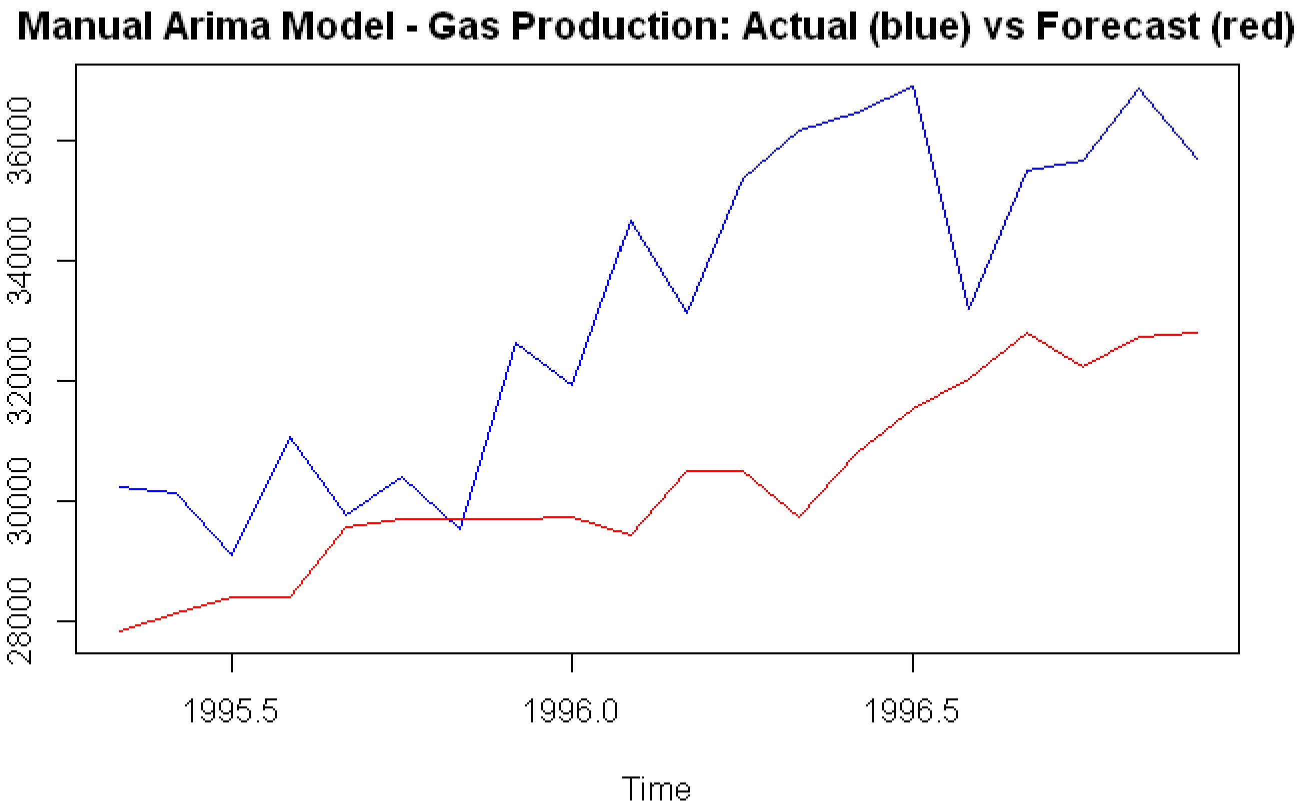

Manual Arima Model - Gas Production: Actual (blue) vs Forecast (red)
30000
32000
34000
36000
1995.5
1996.0
1996.5
Time

- Manual ARIMA MODEL with [p,d,q] = [1,1,1] provides AIC value = 3909 and MAPE = 2.8;

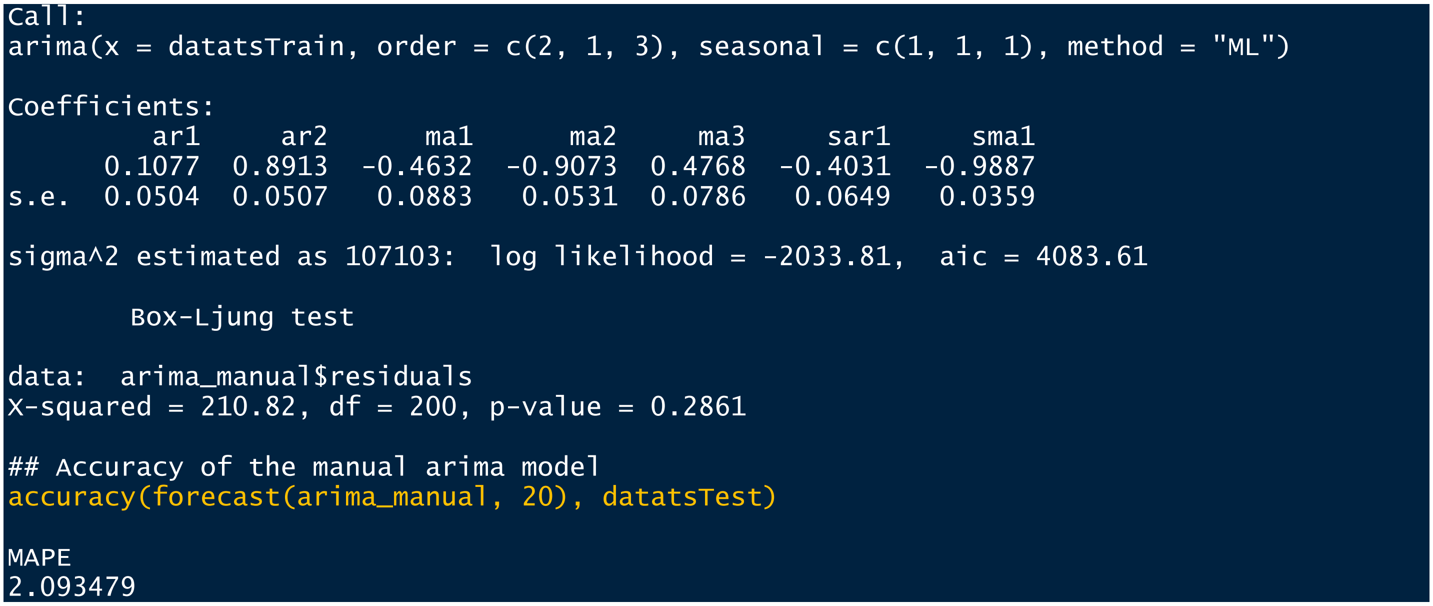


```
Call:
arima(x = datatsTrain, order = c(1, 1, 1), seasonal = c(1, 1, 0), method = "ML")

Coefficients:
         ar1      ma1     sar1
      0.2258  -0.5978  -0.5913
s.e.  0.1085   0.0833   0.0543

sigma^2 estimated as 209596:  log likelihood = -2107.85,  aic = 4223.71

        Box-Ljung test

data:  arima_manual$residuals
X-squared = 247.54, df = 200, p-value = 0.01243

## Accuracy of the manual arima model
accuracy(forecast(arima_manual, 20), datatsTest)
MAPE
2.878399
```



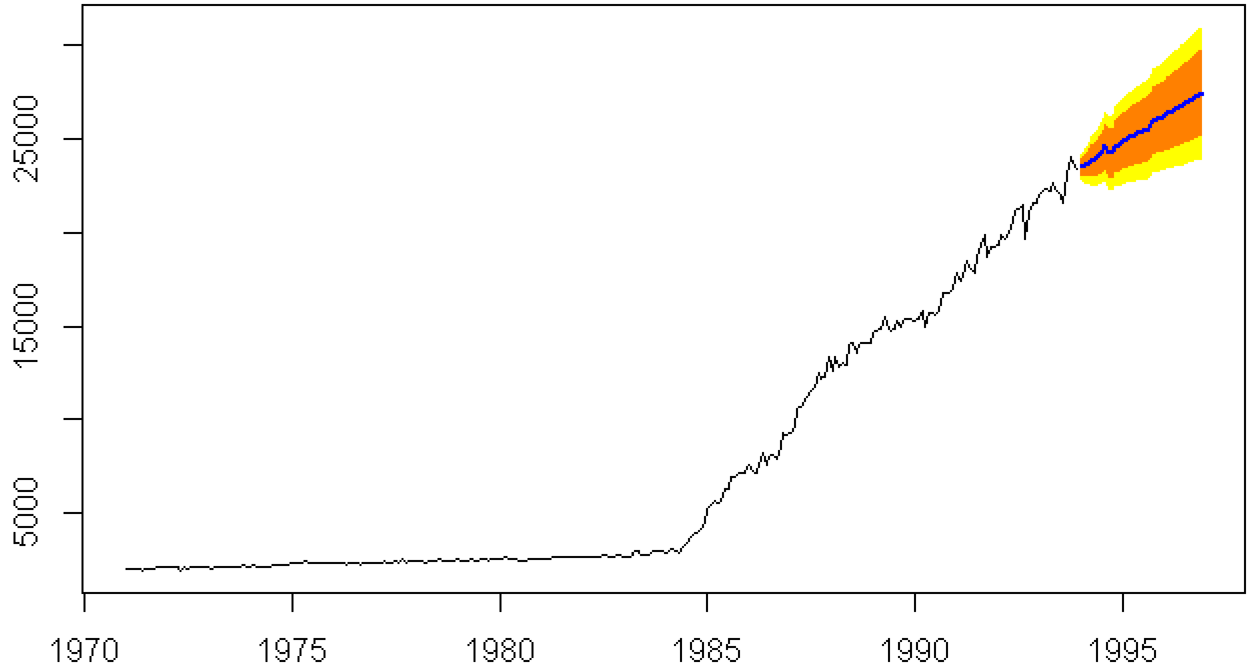


- Manual ARIMA MODEL with [p,d,q] = [0,1,0] provides AIC value = 4083 and MAPE = 2.0;

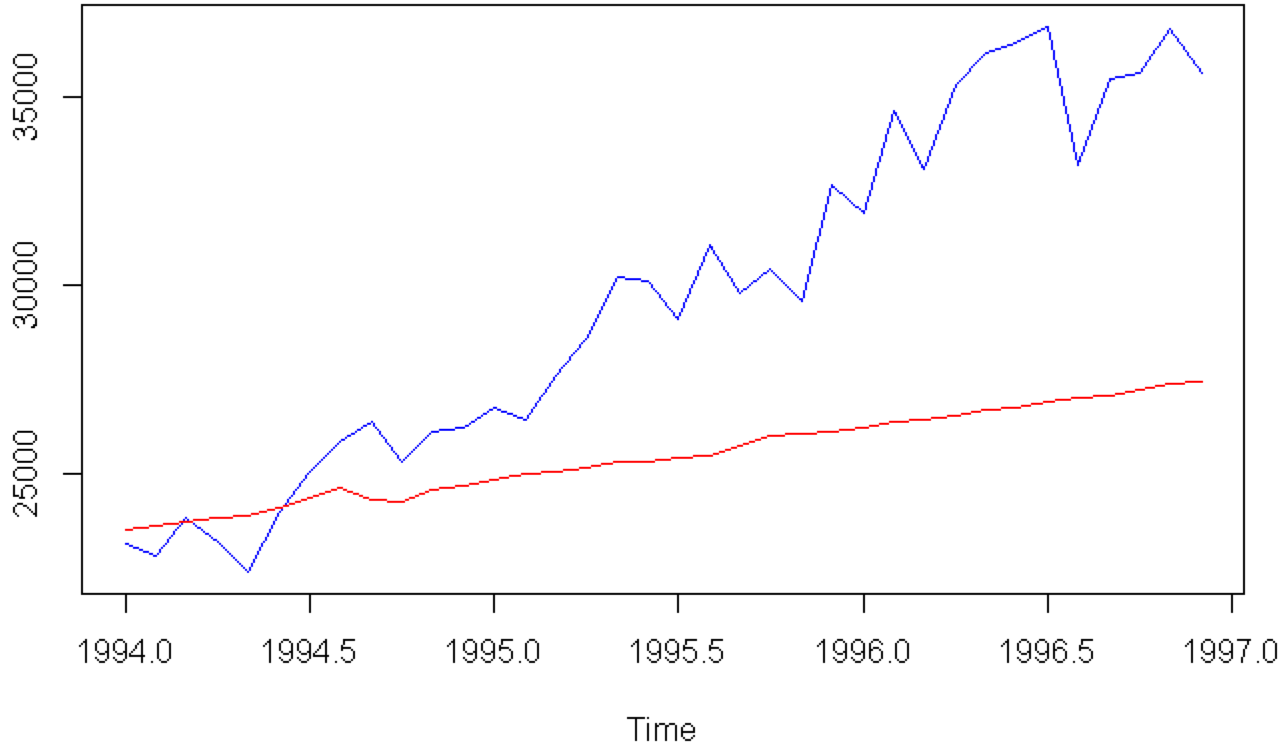


```
Call:
arima(x = datatsTrain, order = c(2, 1, 3), seasonal = c(1, 1, 1), method = "ML")

Coefficients:
         ar1     ar2      ma1      ma2     ma3     sar1     sma1
      0.1077  0.8913  -0.4632  -0.9073  0.4768  -0.4031  -0.9887
s.e.  0.0504  0.0507   0.0883   0.0531  0.0786   0.0649   0.0359

sigma^2 estimated as 107103:  log likelihood = -2033.81,  aic = 4083.61

        Box-Ljung test

data:  arima_manual$residuals
X-squared = 210.82, df = 200, p-value = 0.2861

## Accuracy of the manual arima model
accuracy(forecast(arima_manual, 20), datatsTest)

MAPE
2.093479
```

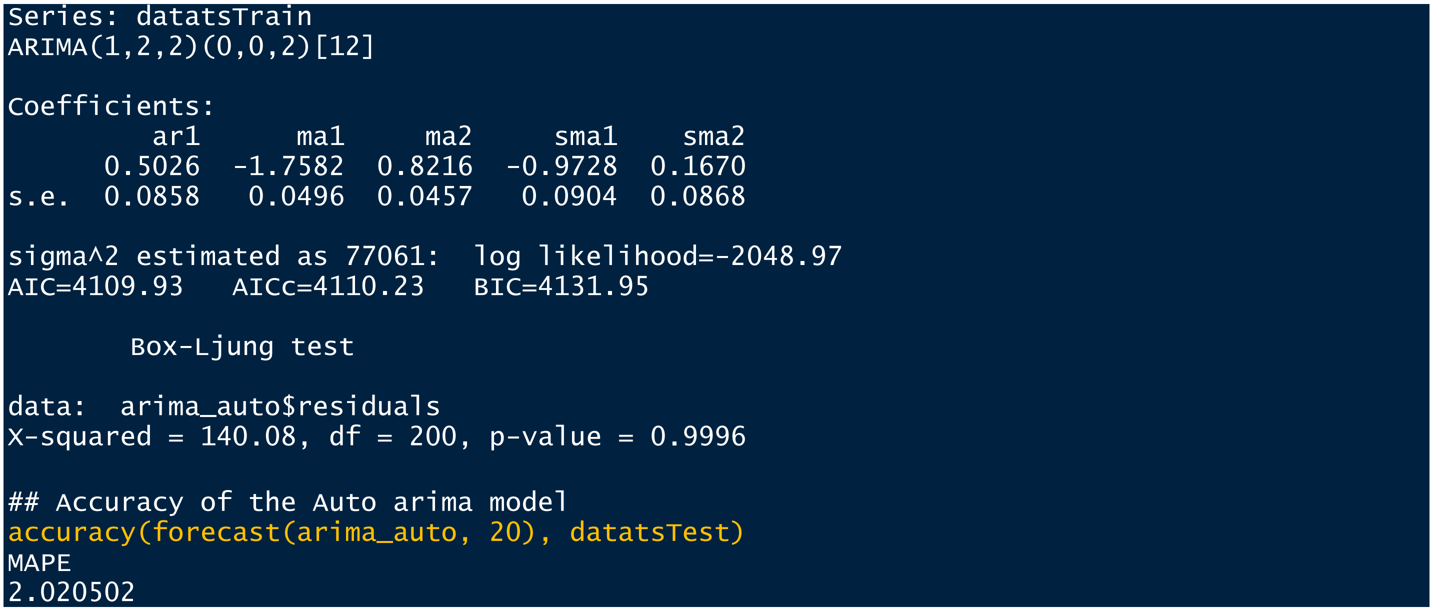
Forecasts from ARIMA(0,1,0)(1,1,1)[12]



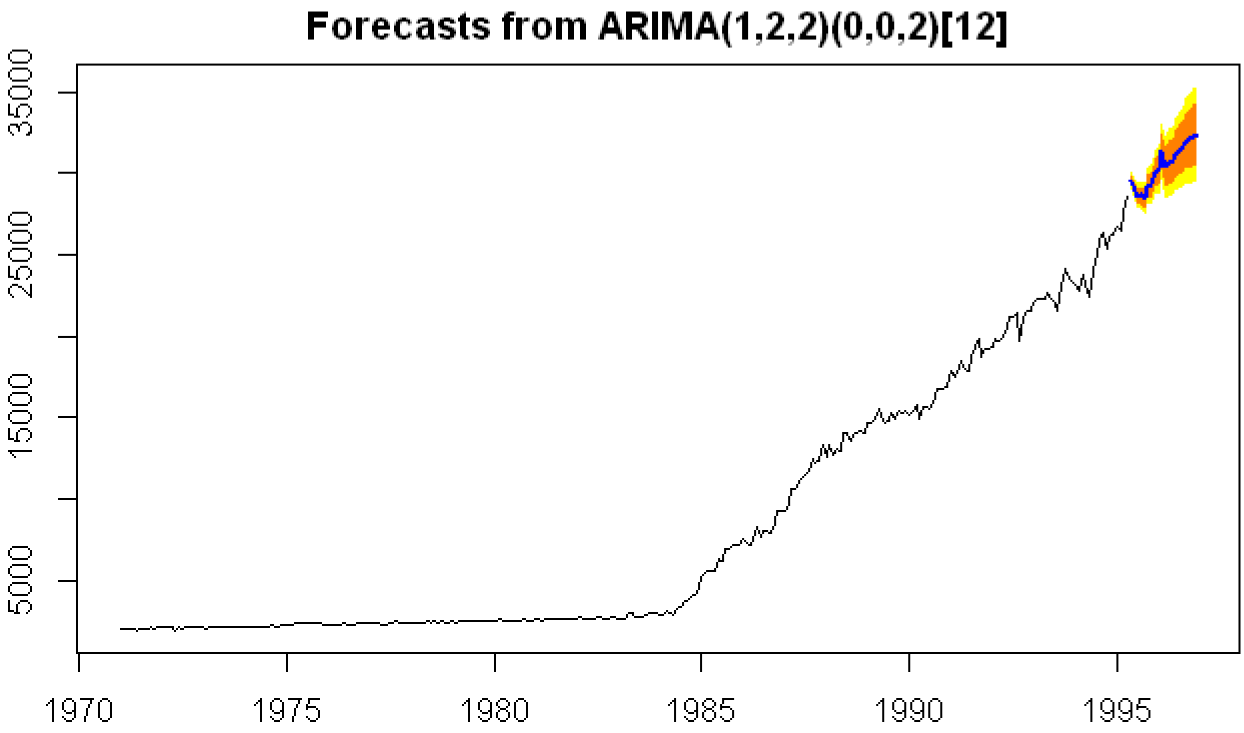
Manual Arima Model - Gas Production: Actual (blue) vs Forecast (red)

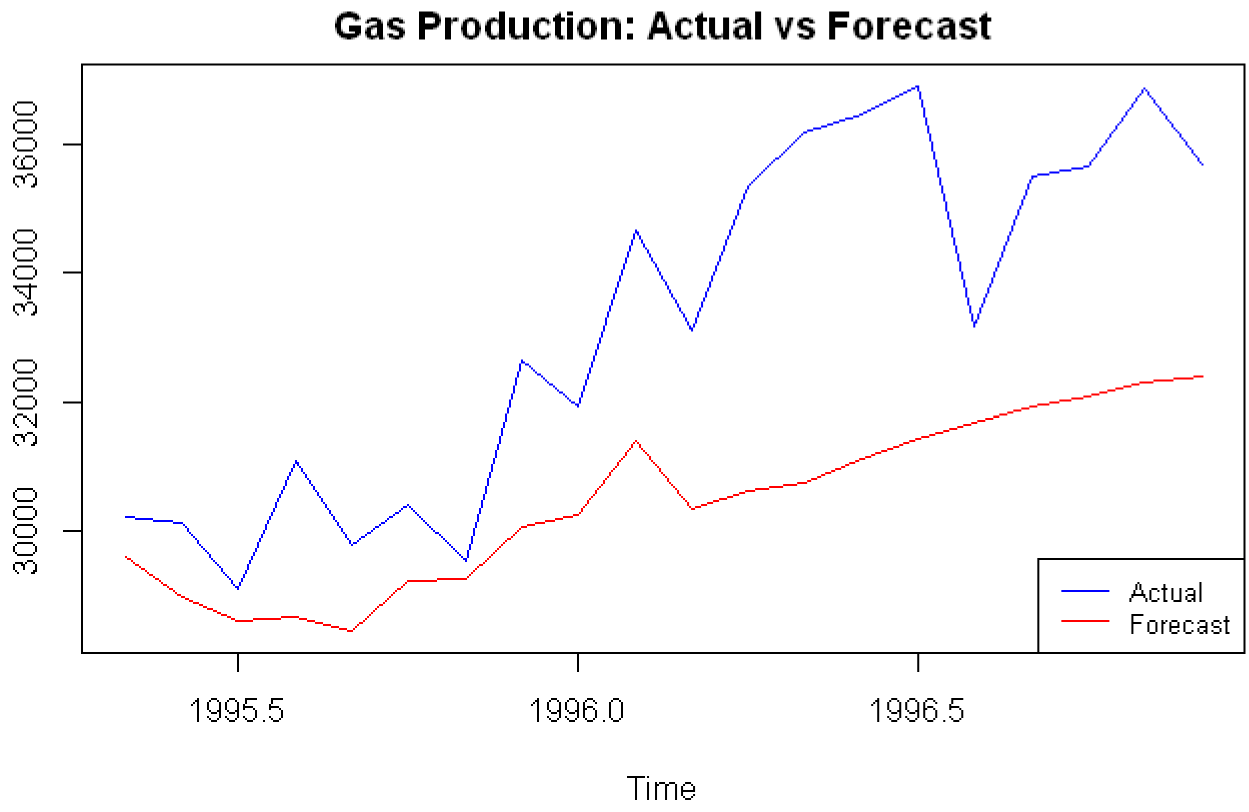


- Auto ARIMA MODEL provides non-seasonal p,d,q and seasonal P,D,Q terms as **(1,2,2) (0,0,2)** with AIC value = 3791 and MAPE = 2.0



```
Series: datatsTrain
ARIMA(1,2,2)(0,0,2)[12]

Coefficients:
         ar1      ma1     ma2     sma1    sma2
      0.5026  -1.7582  0.8216  -0.9728  0.1670
s.e.  0.0858   0.0496  0.0457   0.0904  0.0868

sigma^2 estimated as 77061:  log likelihood=-2048.97
AIC=4109.93   AICc=4110.23   BIC=4131.95

        Box-Ljung test

data:  arima_auto$residuals
X-squared = 140.08, df = 200, p-value = 0.9996

## Accuracy of the Auto arima model
accuracy(forecast(arima_auto, 20), datatsTest)
MAPE
2.020502
```




Gas Production: Actual vs Forecast
36000
34000
32000
30000
1995.5
1996.0
1996.5
Time
Actual
Forecast

MODEL COMPARISON

Going by comparison of MAPE value for all models, Auto ARIMA gives the lowest value of 2.0. AIC values ranges between 4000-4100 in both manual and auto ARIMA models. AUTO ARIMA model gives the best forecast.

.

	MAPE	AIC
MANUAL ARIMA MODEL1	2.32	4118
MANUAL ARIMA MODEL2	2.8	3909
MANUAL ARIMA MODEL3	2.3	4083
AUTO ARIMA MODEL	2.0	4110.23