Australian Monthly Gas Production Report 2019

Nov 23, 2019

Australian Monthly Gas Production Report

Authored By: Deepti Lobo

CONTENTS

Project Objective

To understand and analyze the factors that influence the Australian monthly gas production dataset, present in the Forecast package.

- 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.

Known Facts

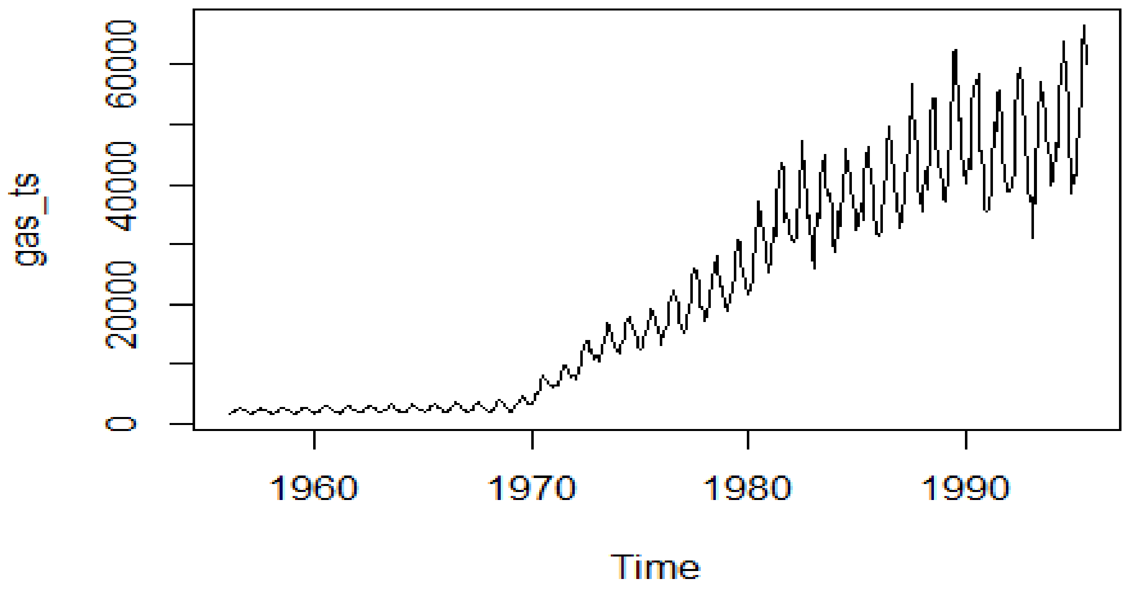
The gas time series dataset contains the Australian monthly gas production data. The data available here, is for the period of 1956 - 1995. We have around 476 observations here. It is a monthly data series. This dataset is drawn from the forecast package. The timeseries is univariate, has only one variable.

Assumptions

- **Data should be stationary** - by stationary it means that the properties of the series doesn’t depend on the time when it is captured. A white noise series and series with cyclic behavior can also be considered as stationary series.
- **Data should be univariate** - ARIMA works on a single variable. Auto-regression is all about regression with the past values.

Exploratory Data Analysis (EDA)

```
### Convert this into timeseries object
gas_ts = ts(gas,start = c(1956,1),frequency = 12)

#Plot the data set
plot(gas_ts)
```



From the graph we can see that the data is spread from 1956 to 1995. The data has upward trend and seasonality.

```
#Class of the dataset
class(gas_ts)

## [1] "ts"

### To print start of the series
start(gas_ts)

## [1] 1956    1

### To print end of the series
end(gas_ts)

## [1] 1995    8
```

```
### To print frequency of the series
frequency(gas_ts)

## [1] 12

cycle(gas_ts)

##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 1956   1   2   3   4   5   6   7   8   9  10  11  12
## 1957   1   2   3   4   5   6   7   8   9  10  11  12

|
## 1994   1   2   3   4   5   6   7   8   9  10  11  12
## 1995   1   2   3   4   5   6   7   8
```

This series starts from 1956, 1st month (i.e) January and ends at 1995, 8th month (i.e) August. Frequency of the data is 12, which implies that this is a monthly series. From the cycle we can say that all the monthly values are available from Jan,1956 to Aug,1995. The dataset does not have any missing values.

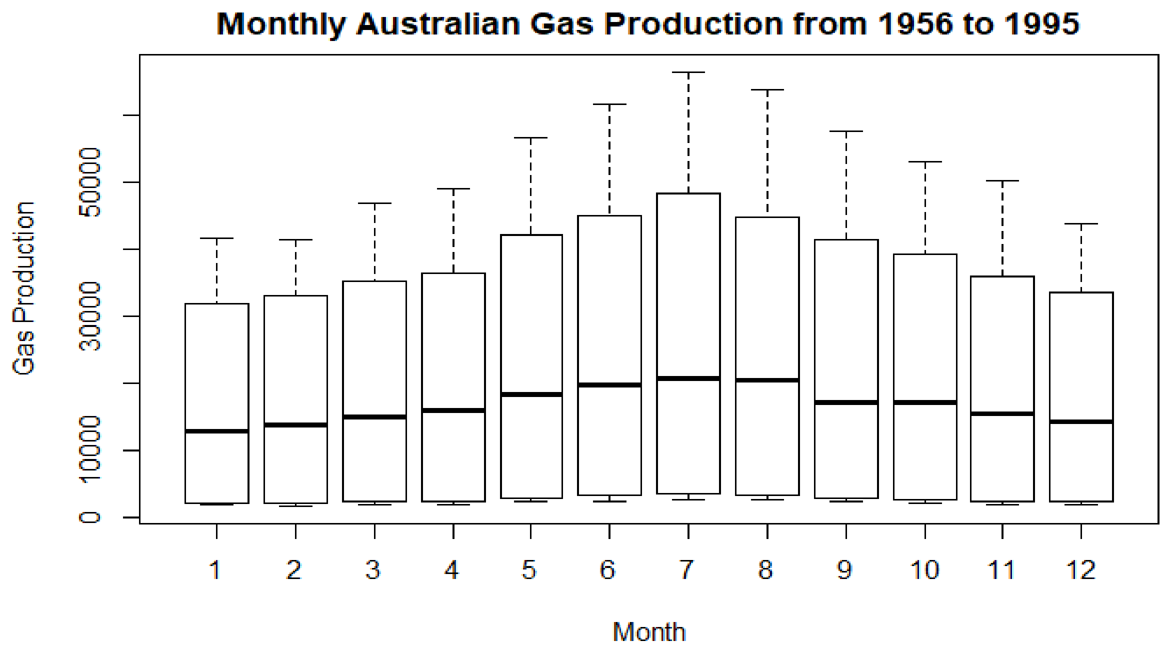
Descriptive Analysis

```
summary(gas_ts)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1646    2675   16788   21415   38629   66600
```

The difference between the median and max value, might be read as if the data is skewed. However, for a time series such an interpretation should be made with caution, as time series may have two additional components (i.e) trend and seasonality. Hence the difference could be because of these two components.

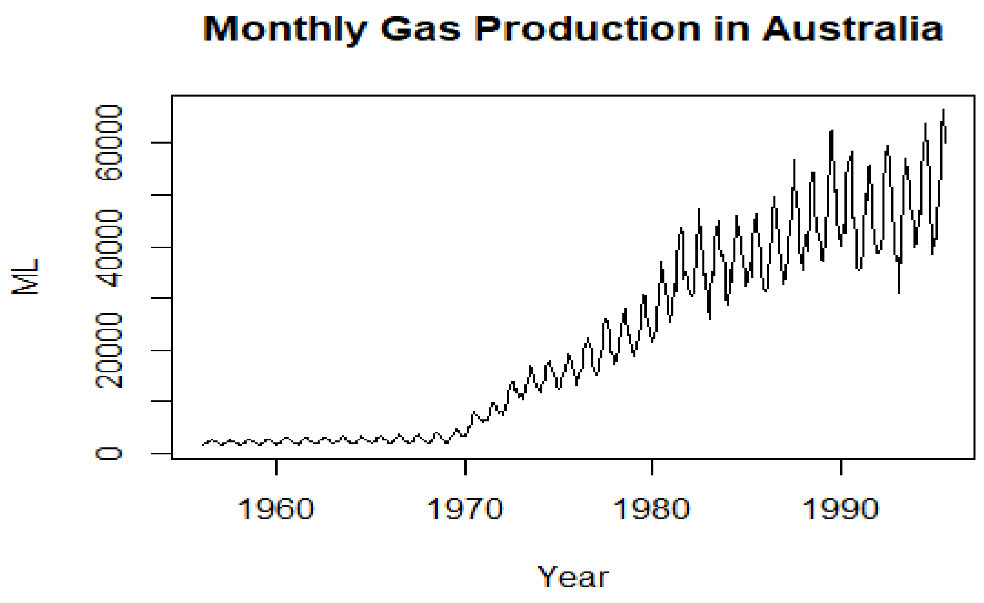
Data Visualization

```
#Box plot function to see any seasonal effects
boxplot(gas_ts~cycle(gas_ts), xlab = "Month", ylab = "Gas Production", main =
"Monthly Australian Gas Production from 1956 to 1995")
```



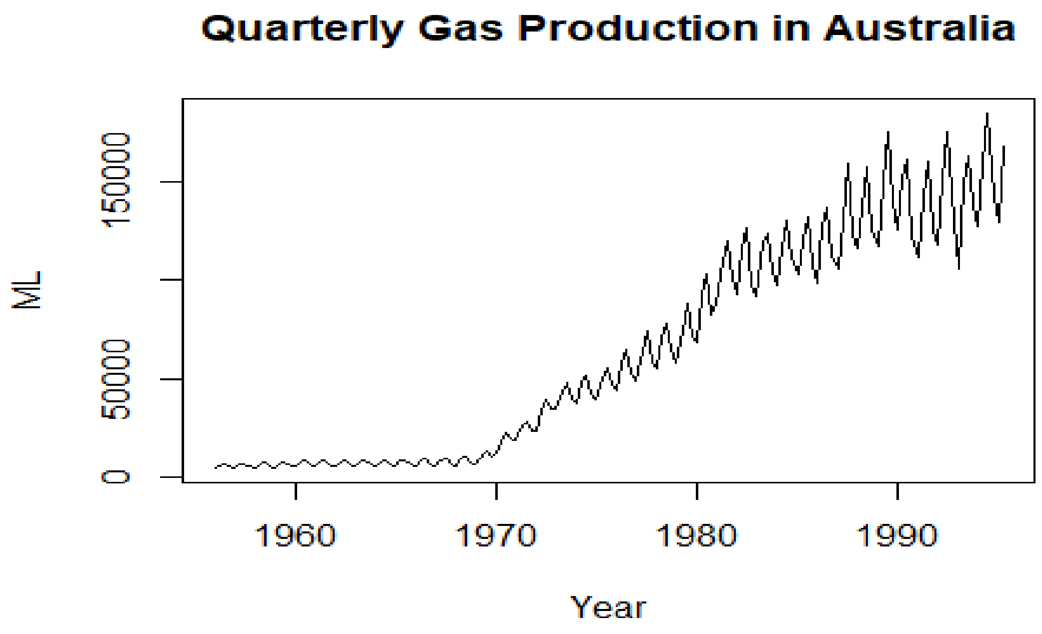
From the initial inference we can see that the gas production seems to gradually increase and then decrease like a bell curve indicating seasonality.

```
### Aggregation at a Quarter and Year Level
gas_ts_qtr = aggregate(gas_ts, nfrequency=4)
gas_ts_yr = aggregate(gas_ts, nfrequency=1)

### Plots

plot.ts(gas_ts, main = "Monthly Gas Production in Australia", xlab = "Year",
ylab = "ML")
```



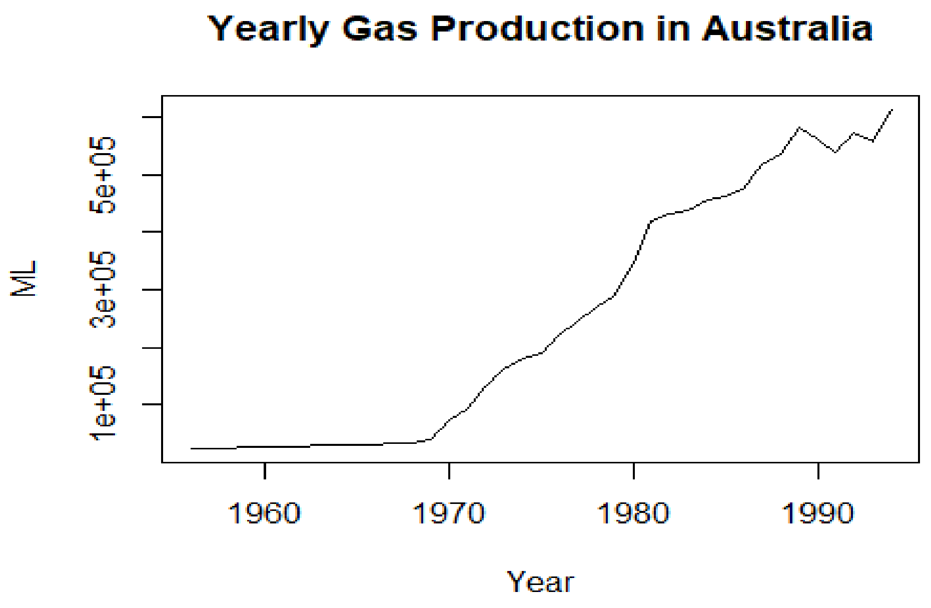
The monthly graph shows an increase in production from 1970 onwards along with seasonality.

```r
plot.ts(gas_ts_qtr, main = "Quarterly Gas Production in Australia", xlab = "Y
ear", ylab = "ML")
```



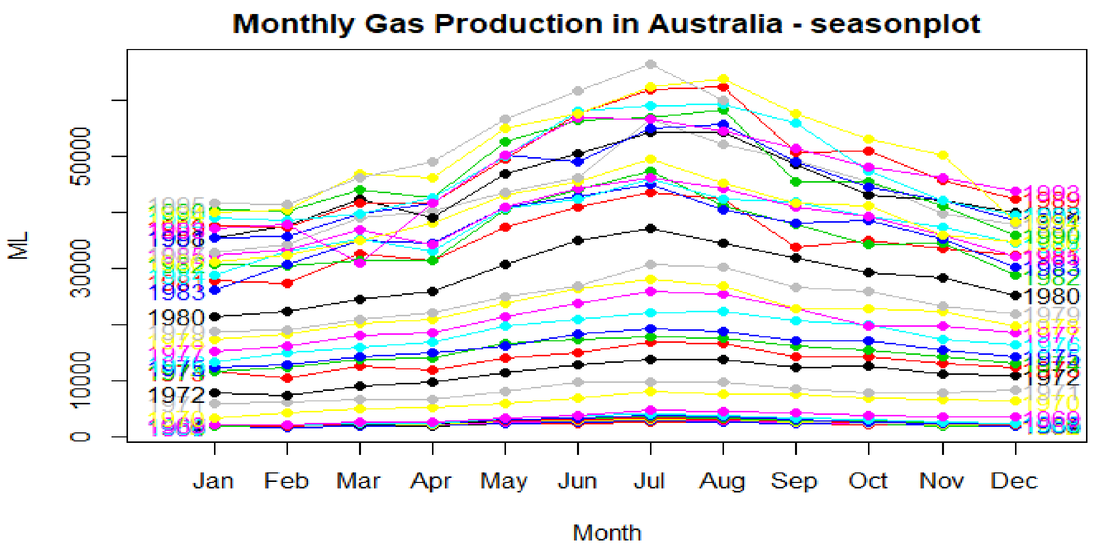
The quarterly graph shows an increase in production from 1970 onwards along with seasonality.

```
plot.ts(gas_ts_yr, main = "Yearly Gas Production in Australia", xlab = "Year"
, ylab = "ML")
```



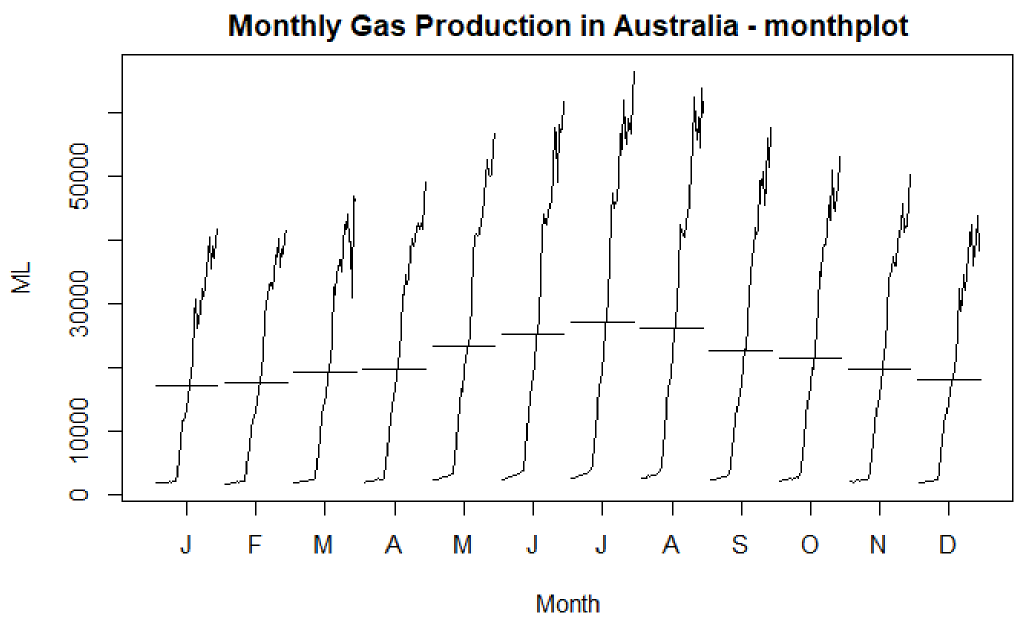
The yearly graph shows an increase in production from 1970 onwards.

```
### Seasonality Plot for further analysis

seasonplot(gas_ts, year.labels = TRUE, year.labels.left=TRUE, col=1:40, pch=1
9, main = "Monthly Gas Production in Australia - seasonplot", xlab = "Month",
ylab = "ML")
```



The plot shows that the production has gradually increased over the years. There seems to be a gradual increase from February till July and then a decrease till December.

```
monthplot(gas_ts, main = "Monthly Gas Production in Australia - monthplot", x
lab = "Month", ylab = "ML")
```

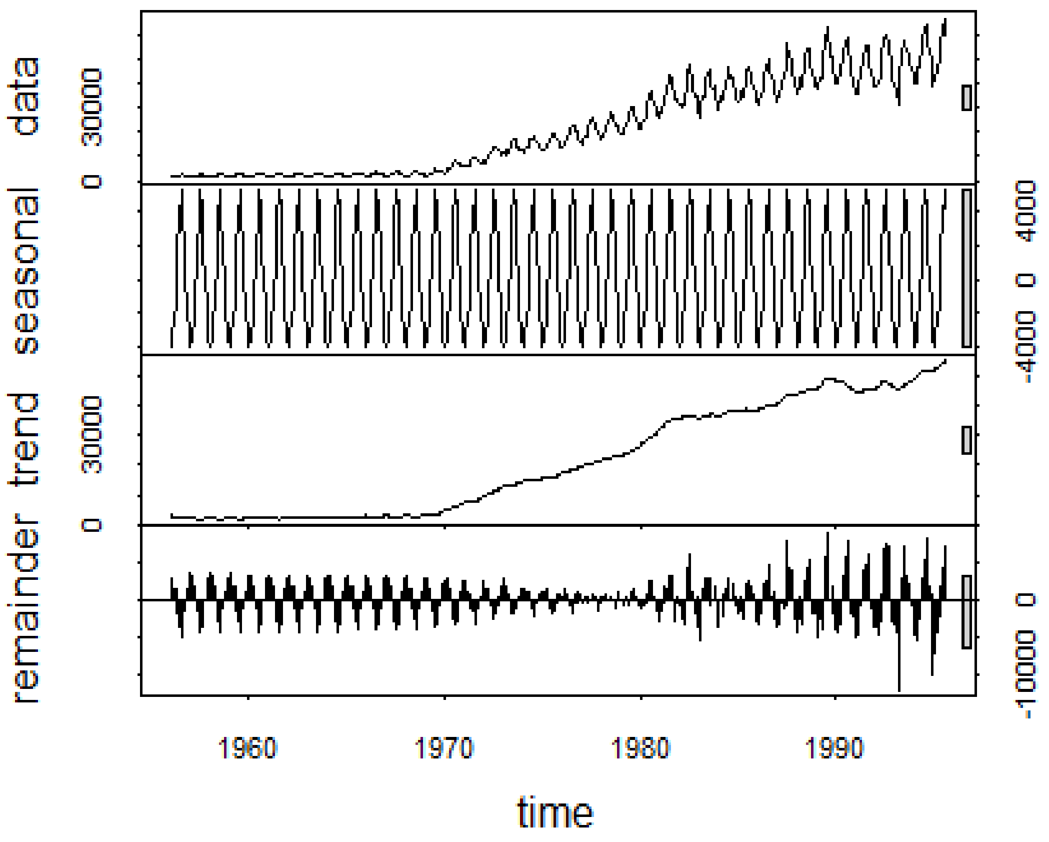


The season plot shows an increase in production up to July and then on a decrease. The monthly plot also similarly shows the peak production in July every year.

Decomposition

STL is a versatile and robust method for decomposing time series. STL is an acronym for "Seasonal and Trend decomposition using Loess".

```
decomp = stl(gas_ts, s.window = "periodic")
plot(decomp)
```



Here we have 4 components:

- **Data** - the actual data plot.
- **Seasonal** - There seems to be a uniform monthly pattern of the data points.
- **Trend** - There is an upward movement of the data points.
- **Remainder** - This is the unexplainable part of the data. Also called as white noise.

Check for stationarity

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, are not stationary - the trend and seasonality will affect the value of the time series at different times.

1. **Test stationarity of the time series (ADF)**

In order to test the stationarity of the time series, let's run the Augmented Dickey-Fuller Test using the adf.test() function from the tseries R package.

First set the hypothesis test:

- The null hypothesis

  H0 : that the time series is non stationary.

- The alternative hypothesis

  HA : that the time series is stationary

```
# Ho : Non-Stationary
# Ha : Stationary

adf.test(gas_ts, alternative = "stationary")

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

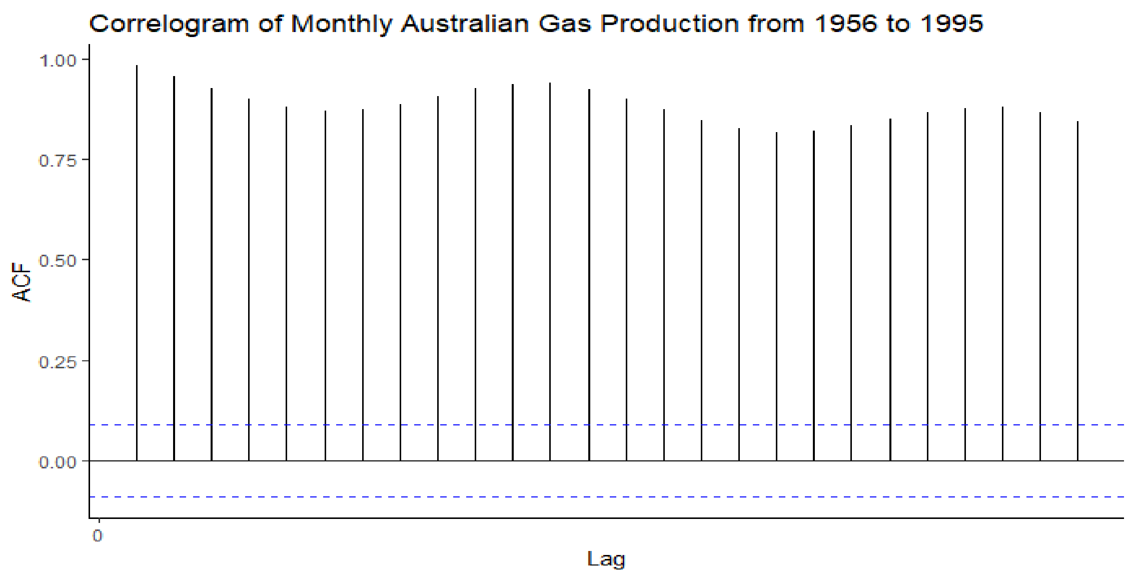
As a rule of thumb, where the p-value is less than 5%, we have a strong evidence against the null hypothesis, so we reject the null hypothesis. As per the test results above, the p-value is 0.2764 which is >0.05. Therefore, we fail to reject the null hypothesis, the time series is non stationary.

2. Test stationarity of the time series (Autocorrelation)

Another way to test for stationarity is to use autocorrelation. We will use autocorrelation function ACF) from the base stats R package. This function plots the correlation between a series and its lags (previous observations) with a 95% confidence interval in blue. If the autocorrelation crosses the dashed blue line, it means that specific lag is significantly correlated with current series.

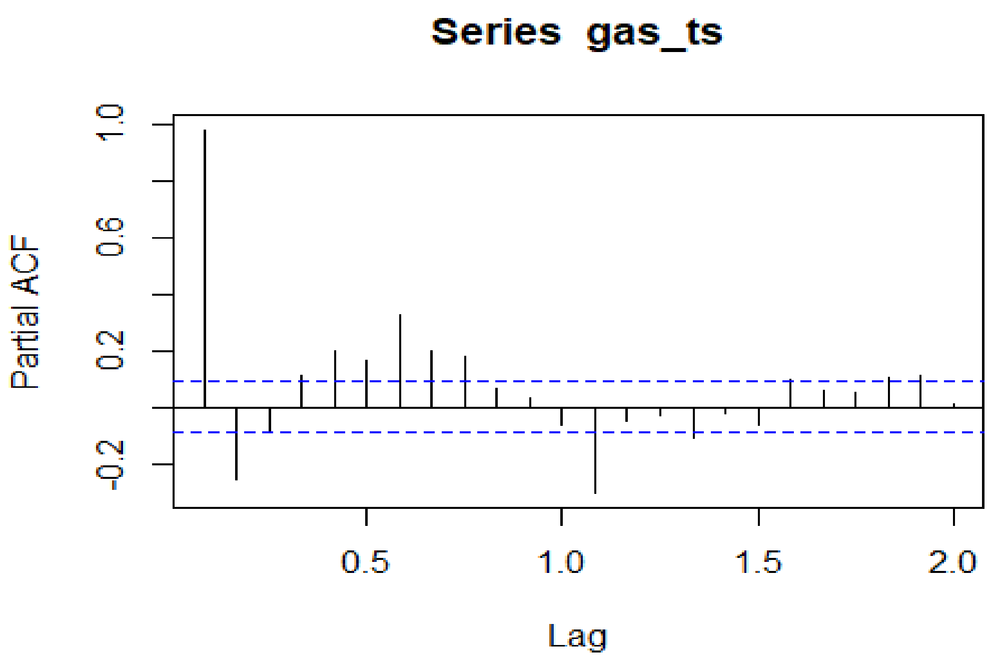
ACF and PACF plots

```
autoplot(acf(gas_ts,plot=FALSE))+ labs(title="Correlogram of Monthly Australi
an Gas Production from 1956 to 1995") + theme_classic()
```



From the ACF plot, we can see that they are significant autocorrelation with many lags in our demand series. The correlation peaks at Lag (0.0), Lag (1.0) and Lag (2.0) indicating seasonality. Since all the values are above the dotted lines, q = 0,1,2,3,4,5,6,7,8,9,10,11.

```
pacf(gas_ts, lag.max = 24)
```

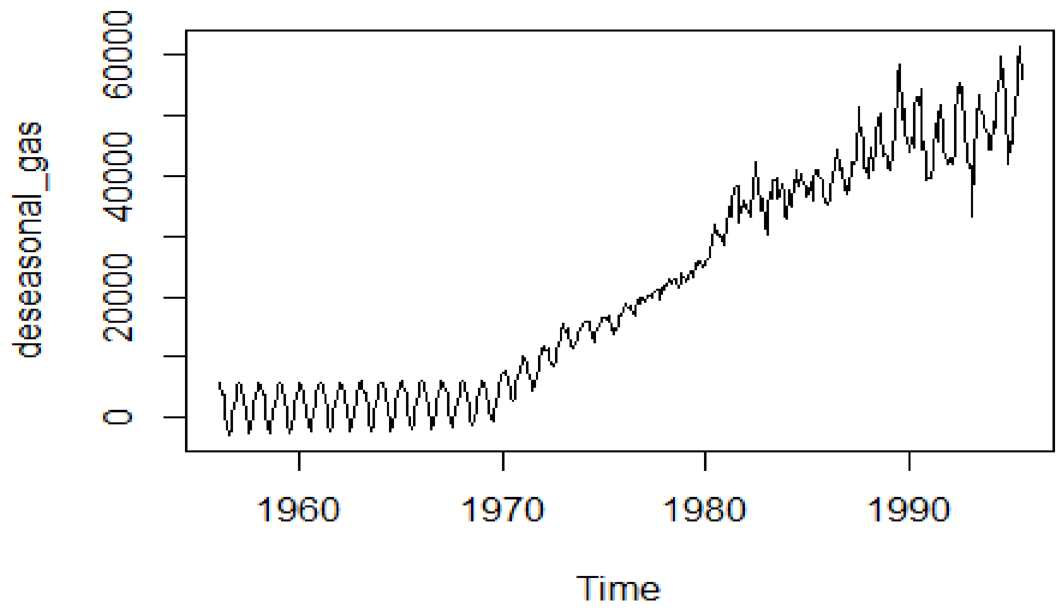


PACF plot shows that there could be monthly seasonality since the plot peaks at intervals of 12. The values above the dotted lines give the values, p = 1,2,4,5,6,7,8,9.

De-seasonalizing the time series data

Let's remove seasonality, seasadj returns seasonally adjusted data constructed by removing the seasonal component.

```
deseasonal_gas=seasadj(decomp)
plot(deseasonal_gas)
```



You can see that the seasonal variation has been removed from the seasonally adjusted time series. The seasonally adjusted time series now just contains the trend component and a remainder component (white noise).

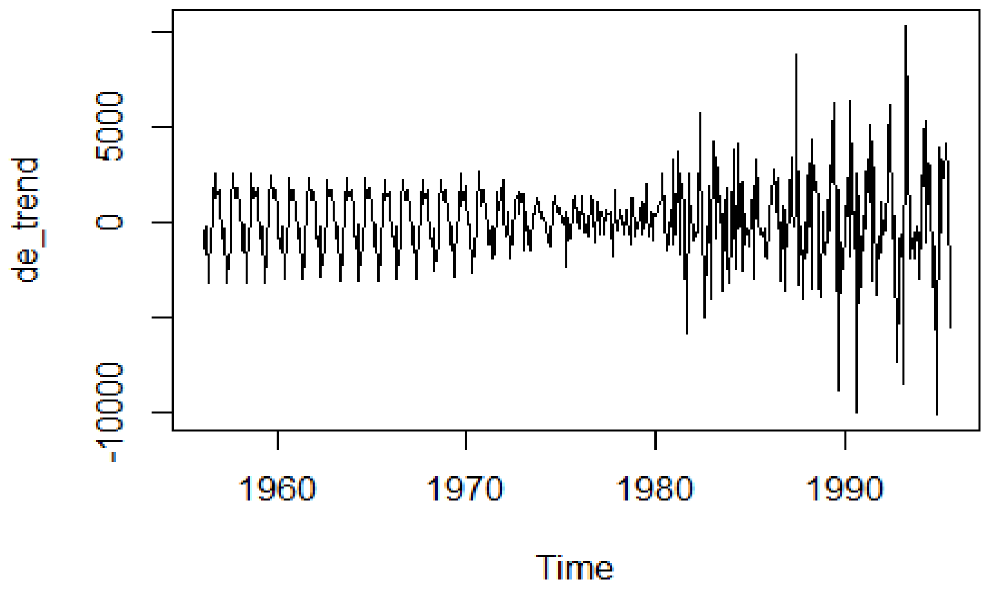
```
adf.test(deseasonal_gas, alternative = "stationary")

##
##  Augmented Dickey-Fuller Test
##
## data:  deseasonal_gas
## Dickey-Fuller = -2.4228, Lag order = 7, p-value = 0.3992
## alternative hypothesis: stationary
```

We have run the ADF test again and the p-value is 0.3992 > 0.05. The series is non stationary.

Differencing the time series data

Let's remove trend also. We use differencing to remove the trend from the time series.

```
de_trend = diff(deseasonal_gas, differences = 1)
plot(de_trend)
```



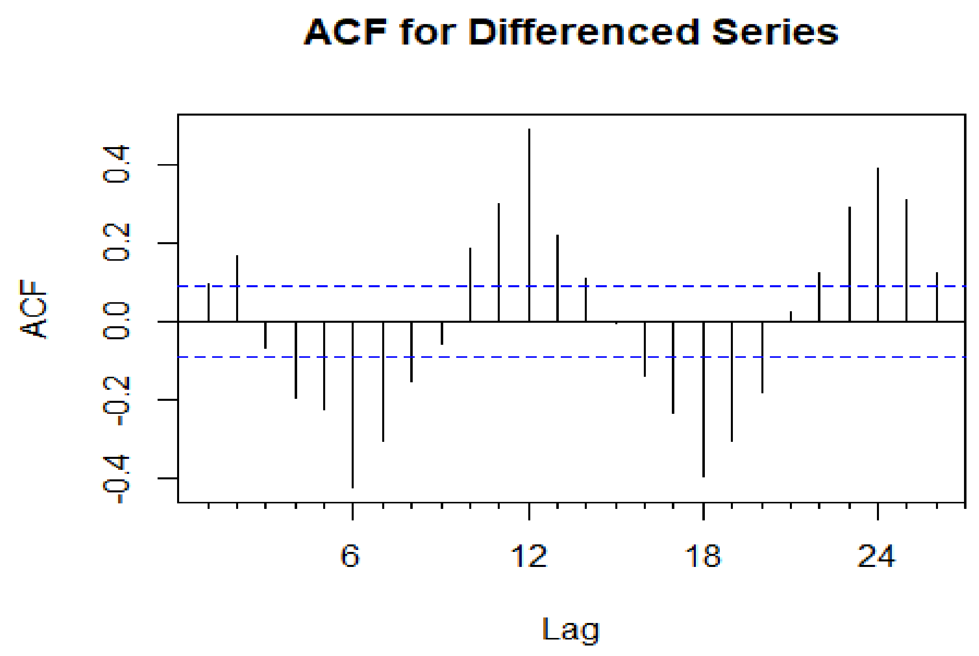
You can see that the seasonal and trend variation has been removed from the adjusted time series. The time series now just contains the remainder component.

```
#Re-run the test to see if the series is stationary
adf.test(de_trend, alternative = "stationary")

## Warning in adf.test(de_trend, alternative = "stationary"): p-value smaller
## than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  de_trend
## Dickey-Fuller = -18.14, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

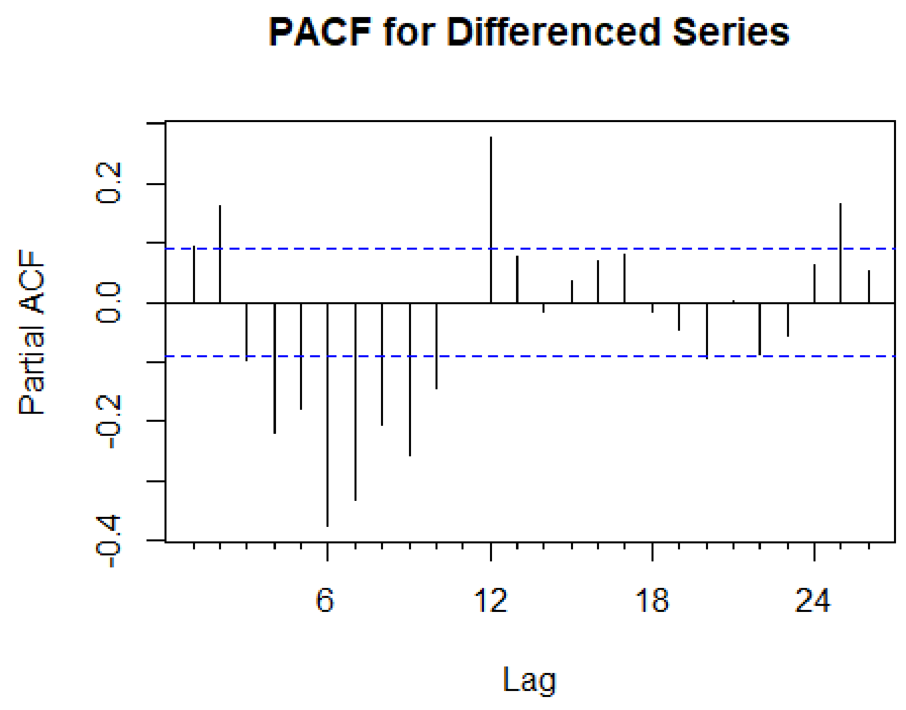
The ADF test was run after de-seasonalizing the series. It returned a p-value of 0.3992. We further went ahead and removed the trend. It gave a p-value of 0.01. As the p-value is <0.05, we can reject the Null Hypothesis. The differenced demand is stationary.

ACF and PACF for diff time series

```
Acf(de_trend, main='ACF for Differenced Series') #q = 1,3,4,5,6,7,9,10,11
```



```
Pacf(de_trend, main='PACF for Differenced Series') #p= 2,4,5,6,7,8,9,10,12
```



```
#d = 1
```

From the ACF plot, q = 1,3,4,5,6,7,9,10,11. PACF shows, p=2,4,5,6,7,8,9,10,12.

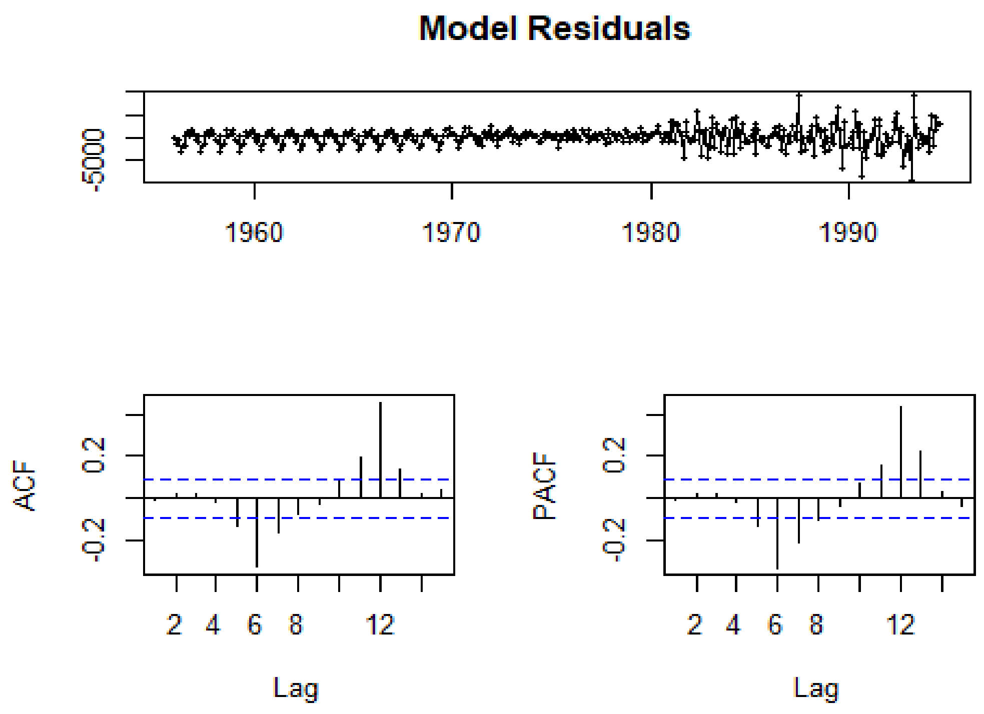
Splitting into training and test sets

```
gasTStrain = window(deseasonal_gas, start=1956, end=c(1994,8))
gasTStest= window(deseasonal_gas, start=c(1994,9), end=c(1995,8))
```

ARIMA Modeling

ARIMA is the abbreviation for Auto Regressive Integrated Moving Average. Auto Regressive (AR) terms refer to the lags of the differenced series, Moving Average (MA) terms refer to the lags of errors and I is the number of differences used to make the time series stationary. As we have seen, the series is univariate, and we have removed the seasonality and trend to make the series as stationary. We can go ahead and build the model.

```
# order : p(pacf)-AR,d(diff order)-I,q(acf)-MA
gasARIMA = arima(gasTStrain, order=c(2,1,4))
gasARIMA

##
## Call:
## arima(x = gasTStrain, order = c(2, 1, 4))
##
## Coefficients:
##           ar1     ar2     ma1      ma2      ma3      ma4
##       -0.1034  0.4164  0.1084  -0.3833  -0.2022  -0.2488
## s.e.   0.1694  0.1169  0.1708   0.1092   0.0553   0.0513
##
## sigma^2 estimated as 3805098:  log likelihood = -4164.87,  aic = 8343.75

tsdisplay(residuals(gasARIMA), lag.max=15, main='Model Residuals')
```

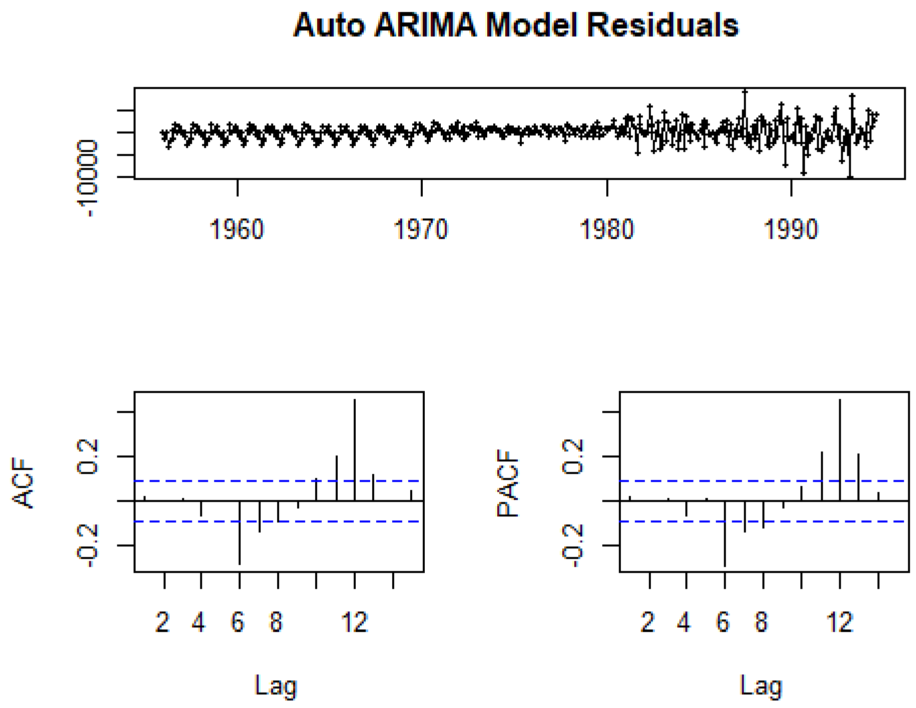


After testing for multiple combinations, we have chosen the order as, c(2,1,4), p = 2, d = 1 and q = 4. The AIC value is 8343.75.

Auto ARIMA Model

Auto-ARIMA uses brute force and tries different combinations of p, q, and d and then returns the best model after evaluation.

```
fit<-auto.arima(gasTStrain, seasonal=FALSE)
fit

## Series: gasTStrain
## ARIMA(1,1,5) with drift
##
## Coefficients:
##          ar1      ma1     ma2      ma3      ma4      ma5     drift
##       0.5012  -0.5666  0.1011  -0.2236  -0.0402  -0.1335  109.4506
## s.e.  0.0913   0.0928  0.0624   0.0684   0.0691   0.0526   25.1449
##
## sigma^2 estimated as 3718914:  log likelihood=-4156.17
## AIC=8328.33   AICc=8328.65   BIC=8361.43

tsdisplay(residuals(fit), lag.max=15, main='Auto ARIMA Model Residuals')
```



Auto ARIMA also fits the different p = 1 and q = 5 parameters for the model, but has a slightly lower AIC, 8328.33.

Ljung box test

The Ljung Box test is a way to test for the absence of serial autocorrelation, up to a specified lag k.

H0: Residuals are independent

Ha: Residuals are not independent

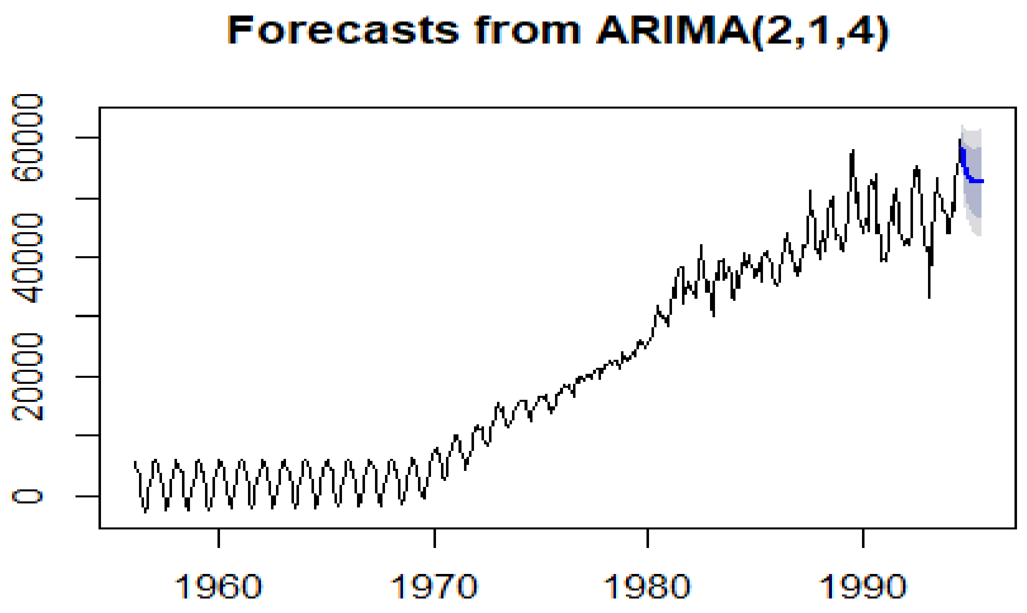
```
Box.test(gasARIMA$residuals)

##
##  Box-Pierce test
##
## data:  gasARIMA$residuals
## X-squared = 0.059609, df = 1, p-value = 0.8071

Box.test(fit$residuals)

##
##  Box-Pierce test
##
## data:  fit$residuals
## X-squared = 0.1637, df = 1, p-value = 0.6858
```

Arima model gives the p-value = 0.8071. The Auto Arima model gives the p-value = 0.6858. Both the models return a value of, p-value > 0.05. The residuals are independent.

Forecasting using the ARIMA model

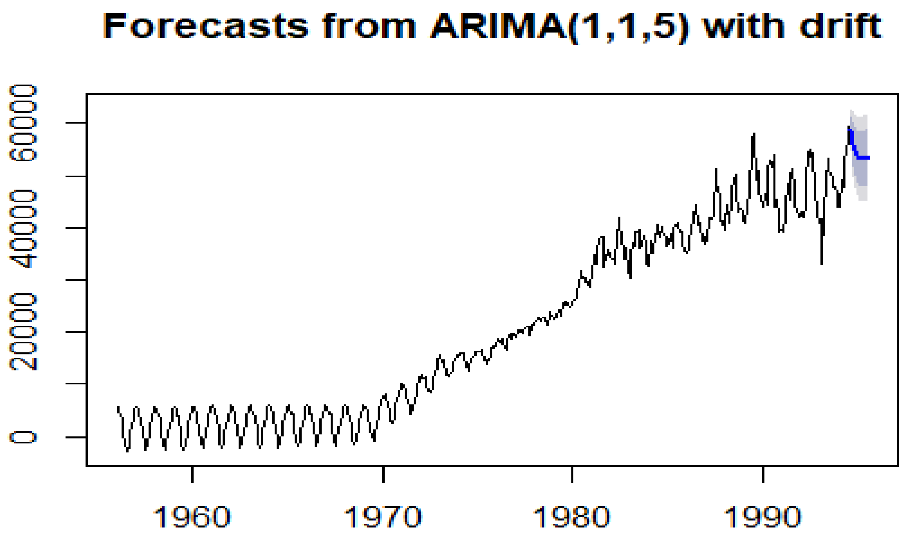
We are predicting for the next 12 months. Hence h is now 12 time points.

```r
#Without Trend
fcast <- forecast(gasARIMA, h=12)
fcast1 <- forecast(fit, h=12)
plot(fcast)
```



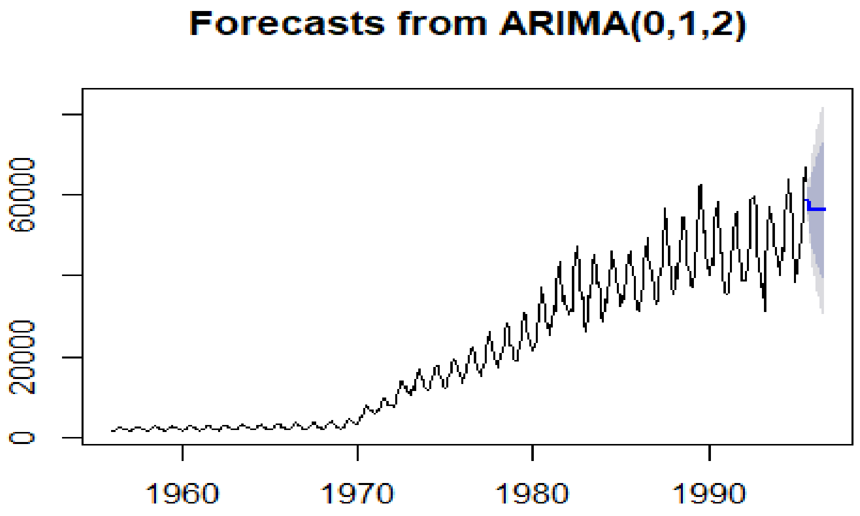
The Arima model without seasonality, forecasts a downward slope and then becoming a horizontal line.

```r
plot(fcast1)
```



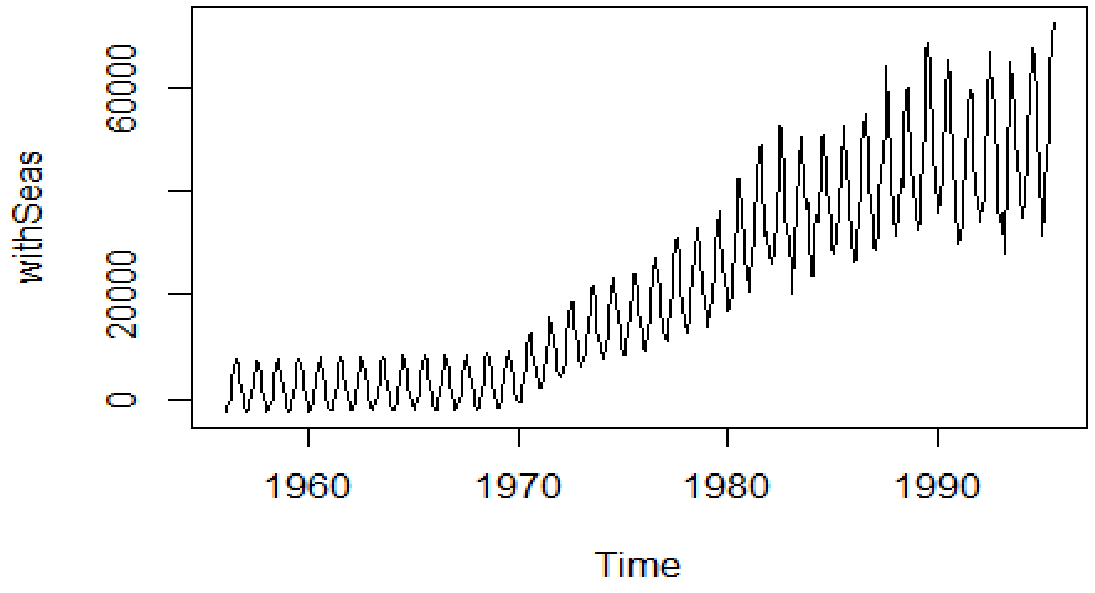
The Auto Arima model without seasonality, forecasts a downward slope and then becoming a horizontal line.

```r
# With Trend
fit1<-auto.arima(gas_ts, seasonal=FALSE)
fcast2=forecast(fit1, h=12)
plot(fcast2)
```



The Auto Arima model without seasonality and trend, forecasts a downward slope and then becoming a horizontal line.

```
withSeas = fcast2$fitted + decompose(gas_ts)$seasonal
plot(withSeas)
```



The model is built with seasonality and trend, forecast a similar graph as the past period.

Accuracy of the forecast

```
f7=forecast(gasARIMA)
accuracy(f7, gasTStest)

##                      ME     RMSE      MAE        MPE     MAPE      MASE
## Training set   257.8937 1948.563 1396.125 -16.726377 54.04196 0.7649493
## Test set     -1922.1809 6141.801 5293.899  -4.931969 10.68154 2.9005734
##                    ACF1 Theil's U
## Training set -0.0113344        NA
## Test set      0.7282091  1.500084

f8=forecast(fit)
accuracy(f8, gasTStest)

##                       ME     RMSE      MAE        MPE     MAPE     MASE
## Training set   -17.39226 1911.752 1317.462 -23.399443 58.84725 0.721849
## Test set     -2631.14412 6320.408 5431.086  -6.294951 11.05075 2.975739
##                      ACF1 Theil's U
## Training set 0.01878295        NA
## Test set     0.72567705  1.555193
```

- **Mean Absolute Percentage Error (MAPE) –** The Arima model gave a value of 10.68%, implies the model is about 89.32% accurate in predicting the next 12 observations. While auto Arima gave, 11.05%, implying an accuracy of 88.95%.
- **Root Mean Squared Error (RMSE)** - Lower values of RMSE indicate a better fit. The Arima model gave a value of 6141.801 while auto Arima gave ,6320.408.

Conclusion

The Australian Monthly Gas Production dataset was spread over a period from 1956 – 1995. It was a monthly series dataset. The time series was univariate. It had both seasonality and trend. When these components were removed, the series became stationary.

For Arima to be run on a time series, it must be univariate and stationary. Ones this was achieved, we divided the dataset into train and test. Two models were built, ARIMA and Auto Arima. We have done a forecast for the next 12 months for both the models.

The Accuracy of both the forecasted models was checked. Arima gave an accuracy of 88.95% while Auto Arima gave, 88.95%. The RMSE returned by Arima model was lower compared to the Auto Arima model.

Based on both these parameters, we can say that Arima Model has returned a better forecast than the Auto Arima model. Hence, we can go for the Arima Model forecast.

Recommendation

We can also build models and forecast the data using, Simple Exponential Smoothing Model and Simple Moving Average for better accuracy.