Project 6 – Australian Gas Production

Assignment Report

- By Samrat Mallik

Table of Contents

1.Project Objective

- 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

2.Assumptions

- Normally distributed.
- Seasonality & Trend may be present
- Linear relationship
- Multivariate normality
- No or little multicollinearity
- No auto-correlation
- Homoscedasticity

3.Exploratory Data Analysis
3.1.Environment Setup and Data Import
3.1.1.Installing Necessary Packages and Invoking Libraries

```
library(data.table) # For converting data into time series format

library(ggplot2) # To plot various plots

## Registered S3 methods overwritten by 'ggplot2':
##   method         from
##   [.quosures     rlang
##   c.quosures     rlang
##   print.quosures rlang

library(fpp2) # For examining seasonality graphically

## Warning: package 'fpp2' was built under R version 3.6.2

## Loading required package: forecast

## Warning: package 'forecast' was built under R version 3.6.2

## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

## Registered S3 methods overwritten by 'forecast':
##   method             from
##   fitted.fracdiff    fracdiff
##   residuals.fracdiff fracdiff

## Loading required package: fma

## Warning: package 'fma' was built under R version 3.6.2

## Loading required package: expsmooth

## Warning: package 'expsmooth' was built under R version 3.6.2

library(forecast) # For various functions related to Time series
library(stats) # For applying tests like acf, Ljung-Box Tests
library(tseries)# For applying Dickey Fuller test

## Warning: package 'tseries' was built under R version 3.6.2

library(MLmetrics) # For applying various metrics related to ML

## Warning: package 'MLmetrics' was built under R version 3.6.2
```

```
##
## Attaching package: 'MLmetrics'

## The following object is masked from 'package:base':
##
##     Recall

data = gas

str(data)

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

start(data)

## [1] 1956    1

end(data)

## [1] 1995    8

frequency(data)

## [1] 12

cycle(data)

##      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
## 1958   1   2   3   4   5   6   7   8   9  10  11  12
## 1959   1   2   3   4   5   6   7   8   9  10  11  12
## 1960   1   2   3   4   5   6   7   8   9  10  11  12
## 1961   1   2   3   4   5   6   7   8   9  10  11  12
## 1962   1   2   3   4   5   6   7   8   9  10  11  12
## 1963   1   2   3   4   5   6   7   8   9  10  11  12
## 1964   1   2   3   4   5   6   7   8   9  10  11  12
## 1965   1   2   3   4   5   6   7   8   9  10  11  12
## 1966   1   2   3   4   5   6   7   8   9  10  11  12
## 1967   1   2   3   4   5   6   7   8   9  10  11  12
## 1968   1   2   3   4   5   6   7   8   9  10  11  12
## 1969   1   2   3   4   5   6   7   8   9  10  11  12
## 1970   1   2   3   4   5   6   7   8   9  10  11  12
## 1971   1   2   3   4   5   6   7   8   9  10  11  12
## 1972   1   2   3   4   5   6   7   8   9  10  11  12
## 1973   1   2   3   4   5   6   7   8   9  10  11  12
## 1974   1   2   3   4   5   6   7   8   9  10  11  12
## 1975   1   2   3   4   5   6   7   8   9  10  11  12
## 1976   1   2   3   4   5   6   7   8   9  10  11  12
## 1977   1   2   3   4   5   6   7   8   9  10  11  12
## 1978   1   2   3   4   5   6   7   8   9  10  11  12
## 1979   1   2   3   4   5   6   7   8   9  10  11  12
```

```
## 1980   1   2   3   4   5   6   7   8   9  10  11  12
## 1981   1   2   3   4   5   6   7   8   9  10  11  12
## 1982   1   2   3   4   5   6   7   8   9  10  11  12
## 1983   1   2   3   4   5   6   7   8   9  10  11  12
## 1984   1   2   3   4   5   6   7   8   9  10  11  12
## 1985   1   2   3   4   5   6   7   8   9  10  11  12
## 1986   1   2   3   4   5   6   7   8   9  10  11  12
## 1987   1   2   3   4   5   6   7   8   9  10  11  12
## 1988   1   2   3   4   5   6   7   8   9  10  11  12
## 1989   1   2   3   4   5   6   7   8   9  10  11  12
## 1990   1   2   3   4   5   6   7   8   9  10  11  12
## 1991   1   2   3   4   5   6   7   8   9  10  11  12
## 1992   1   2   3   4   5   6   7   8   9  10  11  12
## 1993   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
```

3.1.2. Importing the Data

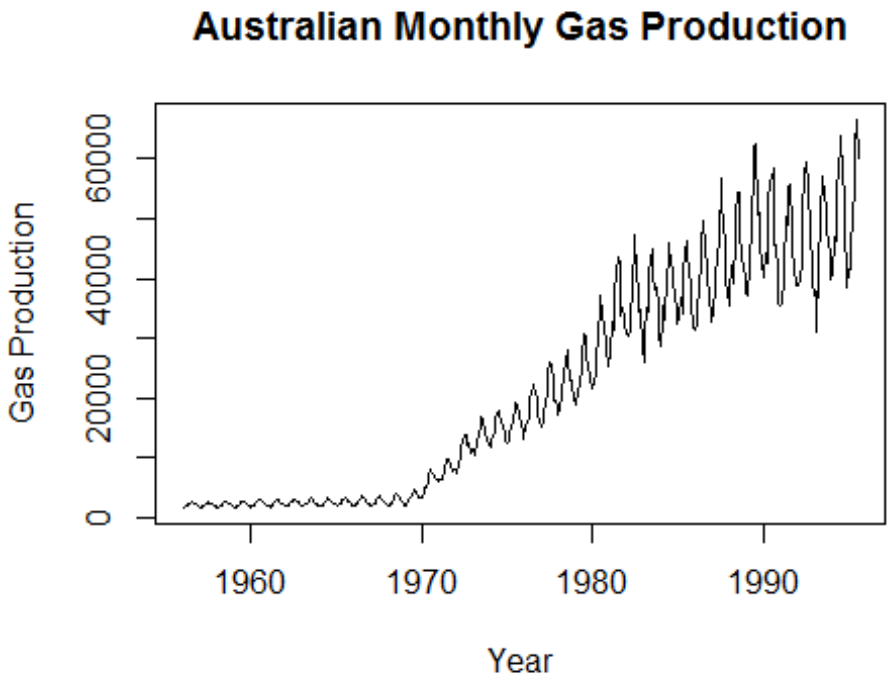
Reading the data as a time-series object

```
ts.data = ts(gas, start = c(1956,1), end = c(1995,8), frequency = 12)

summary(ts.data)

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

str(ts.data)

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

3.2. Descriptive Analysis

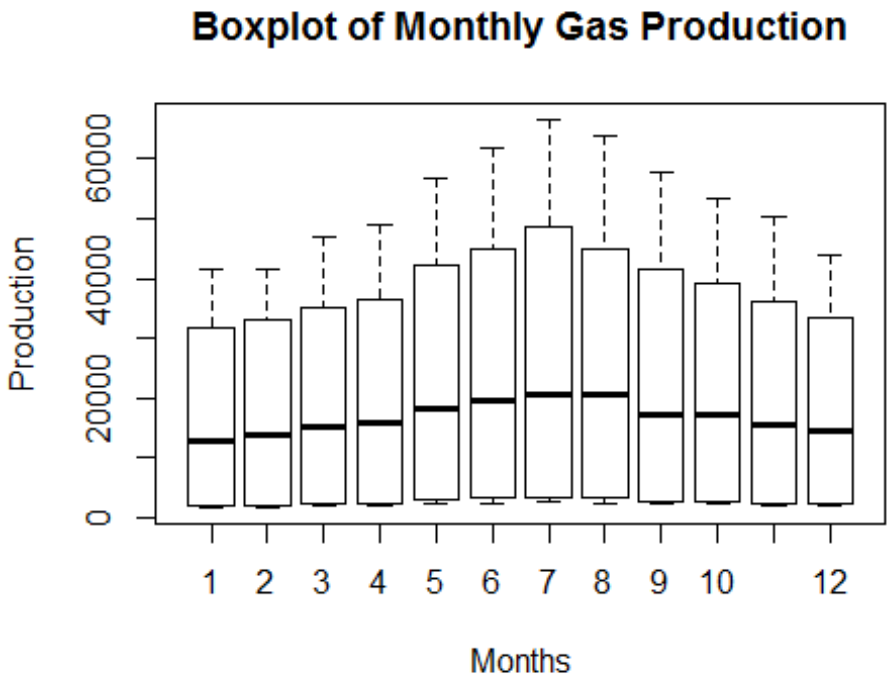
The Gas Production values range from 1646 to 66600 We see that the data spans from January 1956 to August 1995 and it is a monthly time-series data. The data follows a time order and there are no missing values Periodicity of the dataset is monthly.

```
ts.plot(ts.data, xlab = "Year", ylab = "Gas Production", main = "Australian M
onthly Gas Production")
```



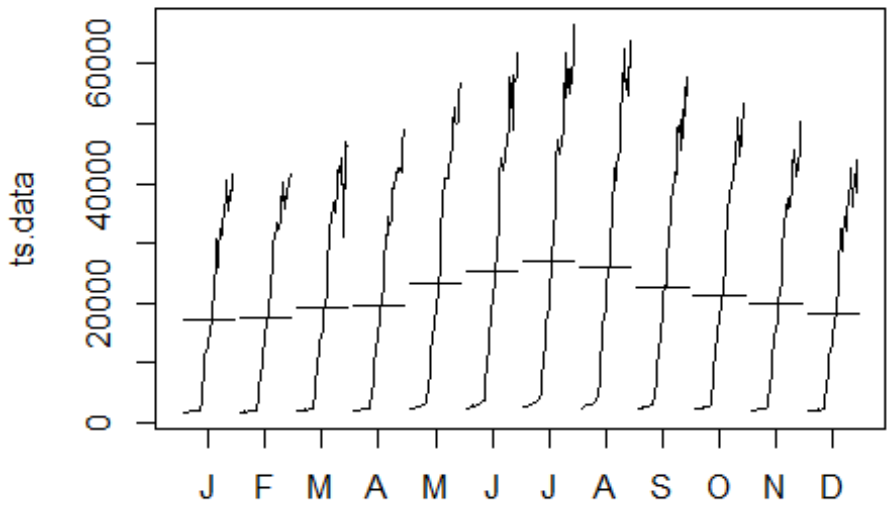
The gradual increase in production suggests the presence of trend component The stable intra-year fluctuations indidcate the presence of seasonal component in time-series data. The data clearly shows that production was stationary with stable seasonal fluctuations from 1956 to 1969. So, the overall series in non-stationary. There onwards production has increased greatly showing strong presence of trend component.

```
boxplot(ts.data ~ cycle(data), xlab = "Months", ylab = "Production", main = "
Boxplot of Monthly Gas Production")
```



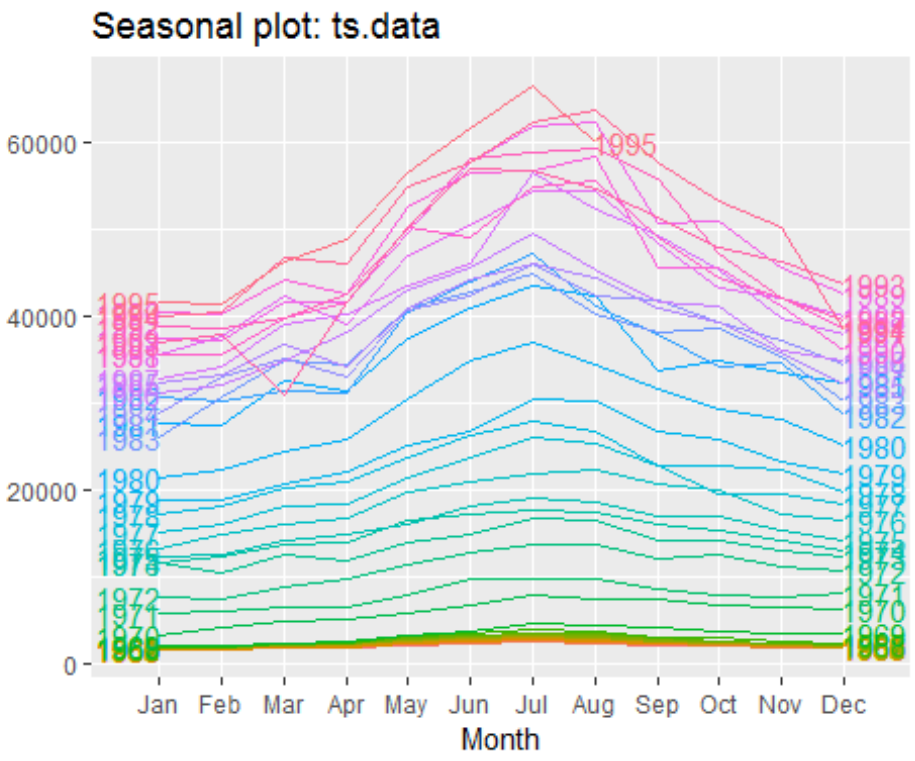
Production appears to be highest in the months of July and August and lowest in December-January

```
monthplot(ts.data)
```

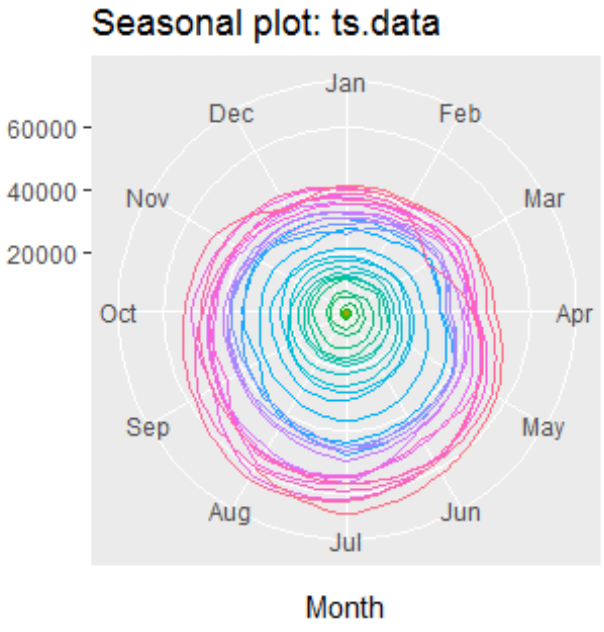


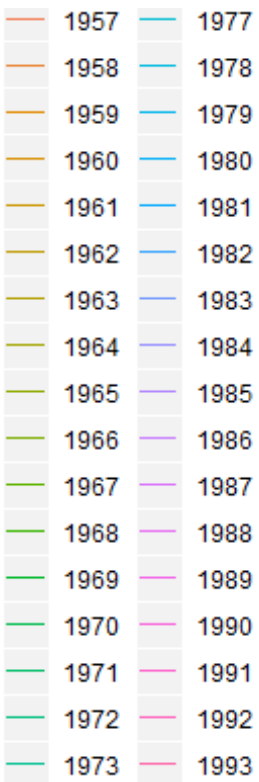
The vertical lines represting monthly production suggest highest production in July and also highest average production in July as represented by the horizontal lines.

```
ggseasonplot(ts.data, year.labels = TRUE, year.labels.left = TRUE)
```




```
ggseasonplot(ts.data, polar = TRUE)
```



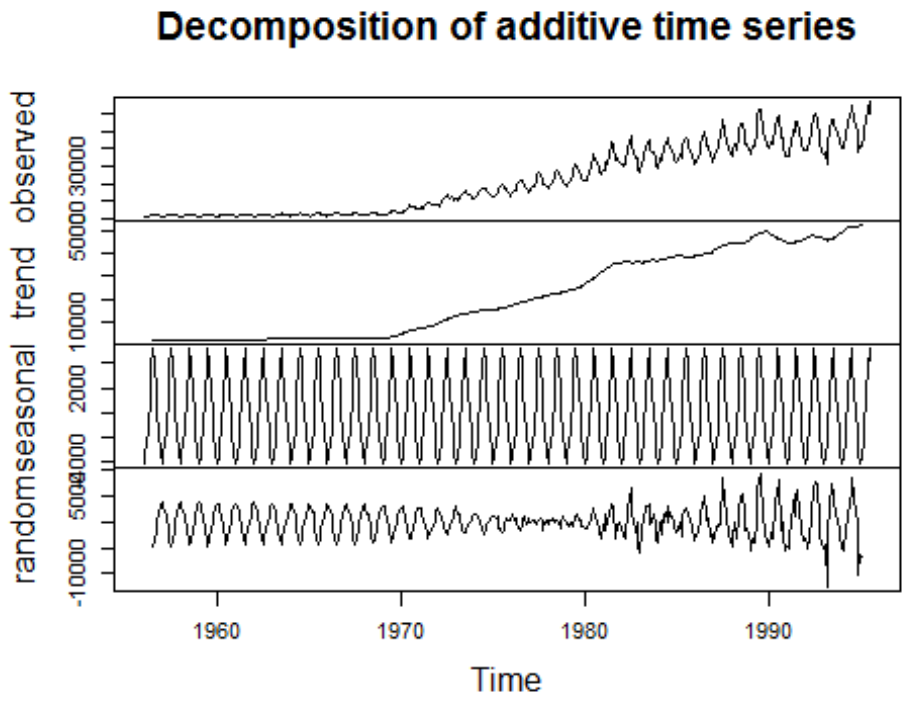


There appears to be a sudden dip in production in March 1993 and a few other instances but the general seasonality suggests highest productoin in July or August Production remained nearly constant till 1969 thereafter increased every year. The above plots show that Production has seasonal fluctuations along with a trend. Thus there is evidence of multiplicative seasonality.

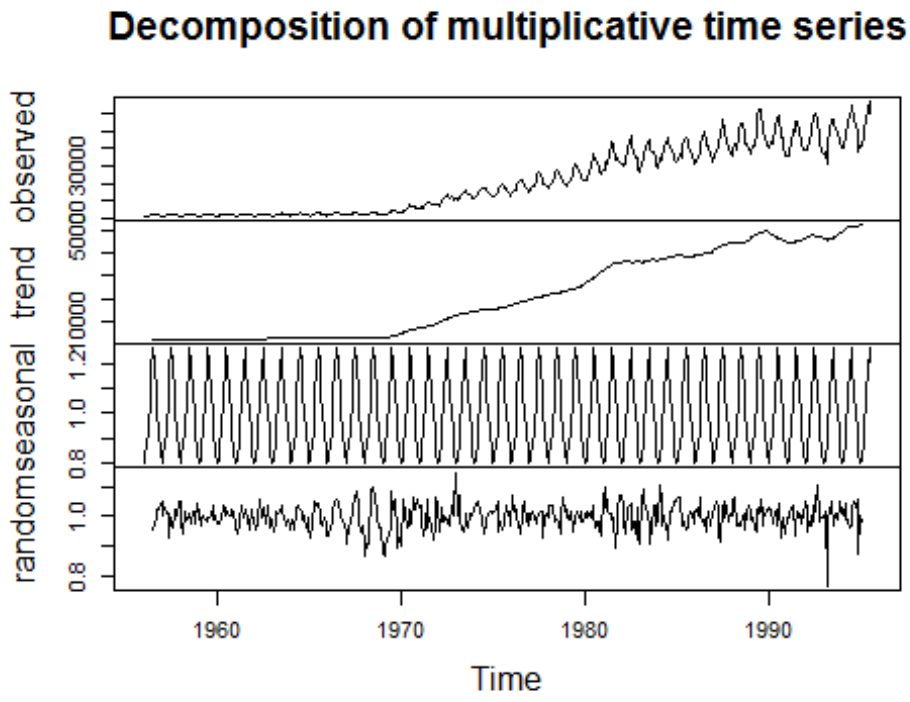
4. Statistical Analysis

4.1. Decomposing the data

```
decomp.data.add = decompose(ts.data, type = "additive")
plot(decomp.data.add)
```




```
decomp.data.multi = decompose(ts.data, type = "multiplicative")
plot(decomp.data.multi)
```



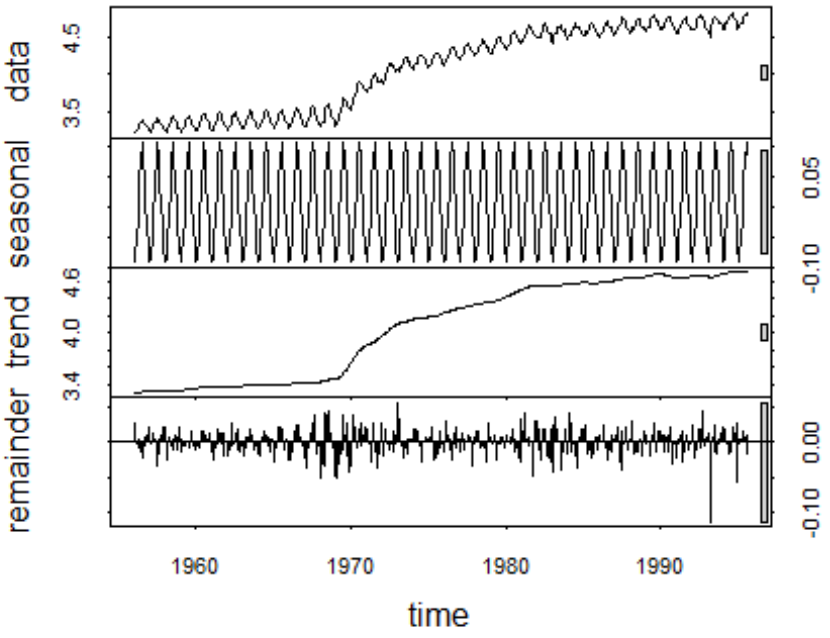
No major changes between the multiplicative and additive models except for the random component.

```
seasonal.indc = round(t(decomp.data.multi$figure),2)
seasonal.indc
```

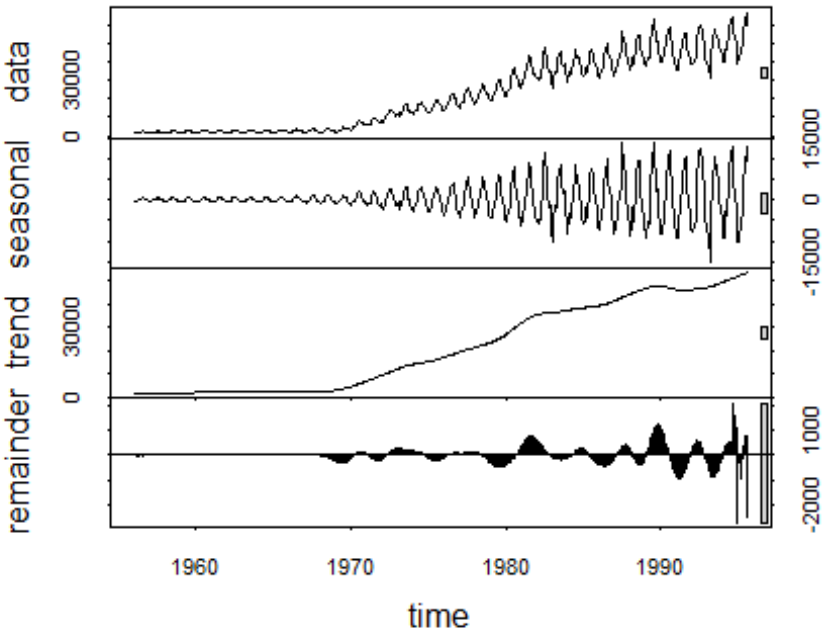
```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
## [1,]  0.8 0.81 0.89 0.91 1.09 1.17 1.26 1.21 1.07  1.01  0.92  0.85
```

By observing he seasonal indices we identify the highest production month as July(bearing the highest value of the seasonal component) and lowest production month as January having the lowest seasonality component.

```
stl.data = stl(log(ts.data), s.window = "p")
plot(stl.data)
```



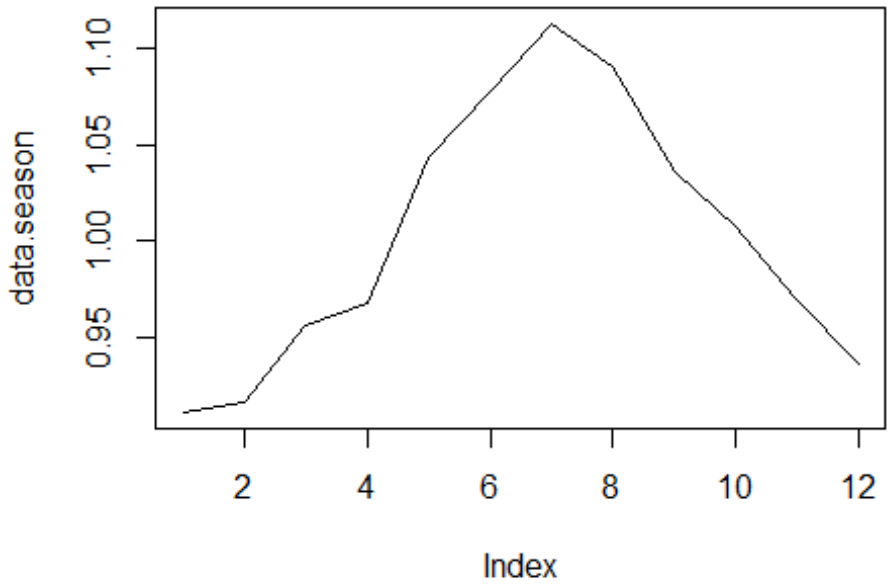

```r
stl.data.3 = stl(ts.data, s.window = 3)
plot(stl.data.3)
```



```
stl.data$time.series[1:12,1]
```

```
##  [1] -0.092826741 -0.087502978 -0.045034554 -0.032304497  0.042273593
##  [6]  0.075306754  0.106786566  0.087073322  0.035760109  0.007539753
## [11] -0.030839467 -0.066231860
```

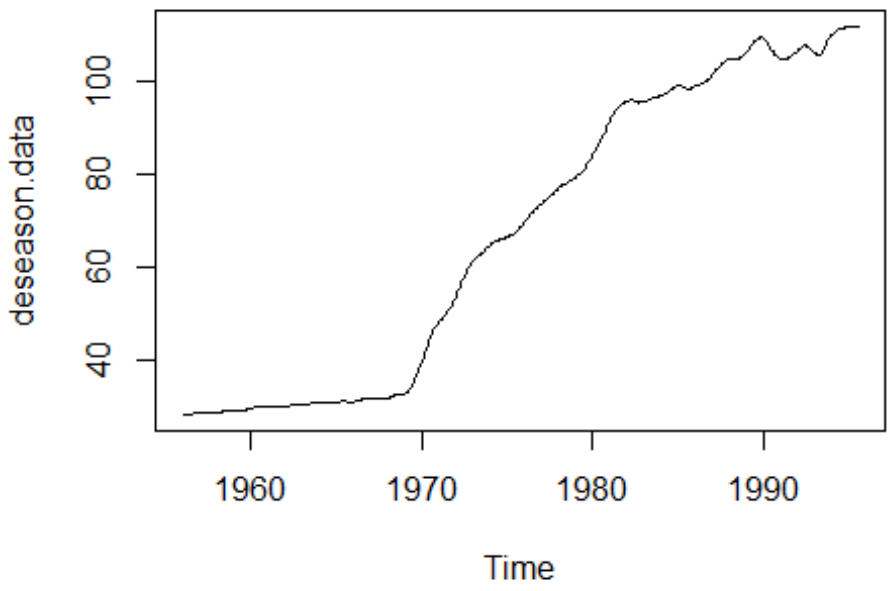
```
data.season = exp(stl.data$time.series[1:12,1])
plot(data.season, type="l")
```



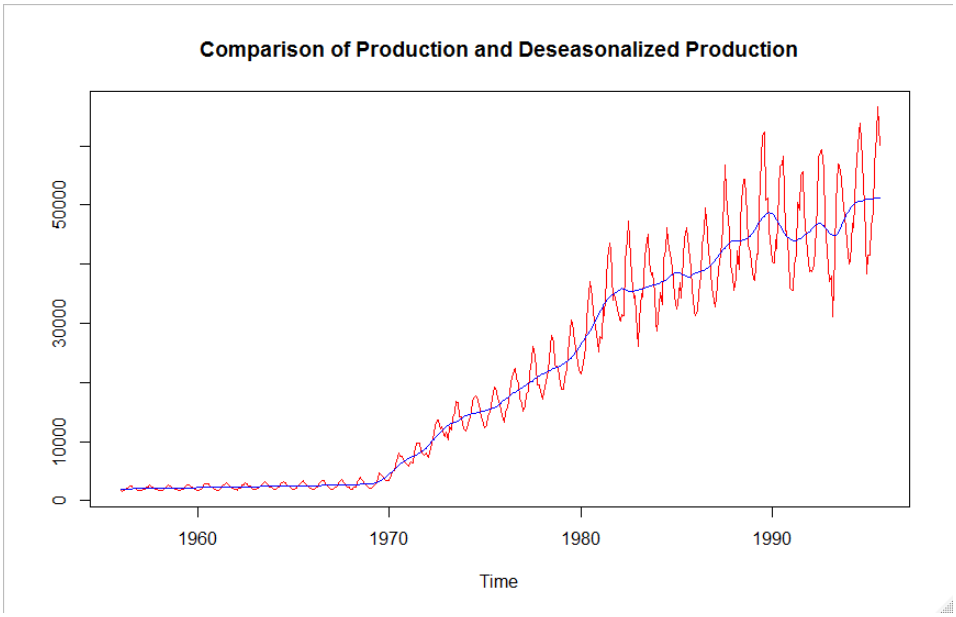
We observe constant seasonality in the data Trend is highly significant as indicated by the small grey bars on the right and increases steadily from 1970.

Deseasonalize the data

```
deseason.data = exp(stl.data$time.series[,2]) + exp(stl.data$time.series[,3])
ts.plot(deseason.data)
```



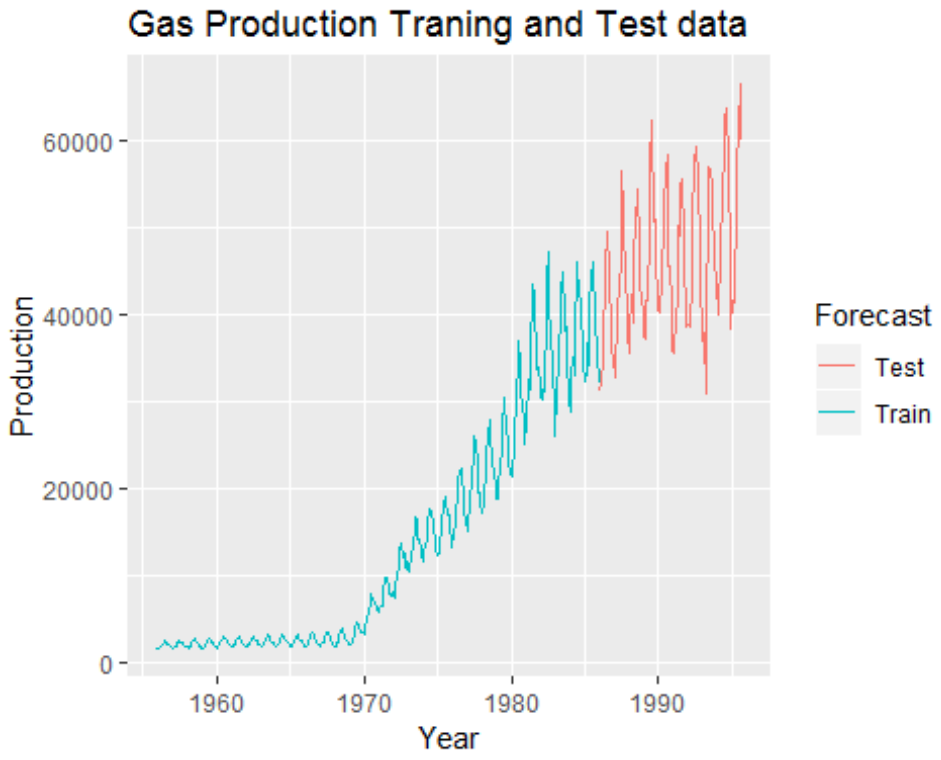

```
ts.plot(ts.data, deseason.data, col=c("red", "blue"), main="Comparison of Pro
duction and Deseasonalized Production")
```



We observe that seasonality has been removed from the time-series data.

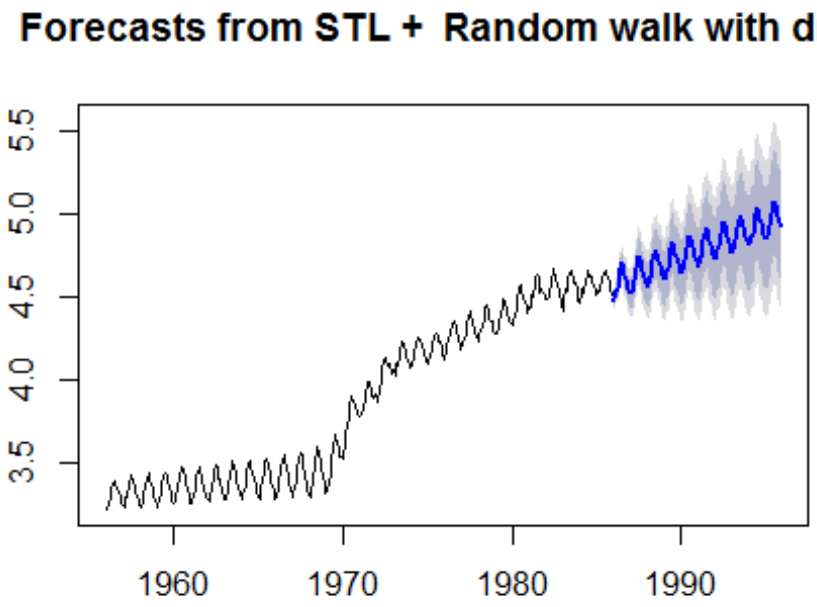
Splitiing the time-series data in train and test data

```
ts.train = window(ts.data, start = c(1956,1), end = c(1985,12), frequency = 1
2)

ts.test = window(ts.data, start = c(1986,1), frequency = 12)

autoplot(ts.train, series="Train") +
  autolayer(ts.test, series="Test") +
  ggtitle("Gas Production Traning and Test data") +
  xlab("Year") + ylab("Production") +
  guides(colour=guide_legend(title="Forecast"))
```

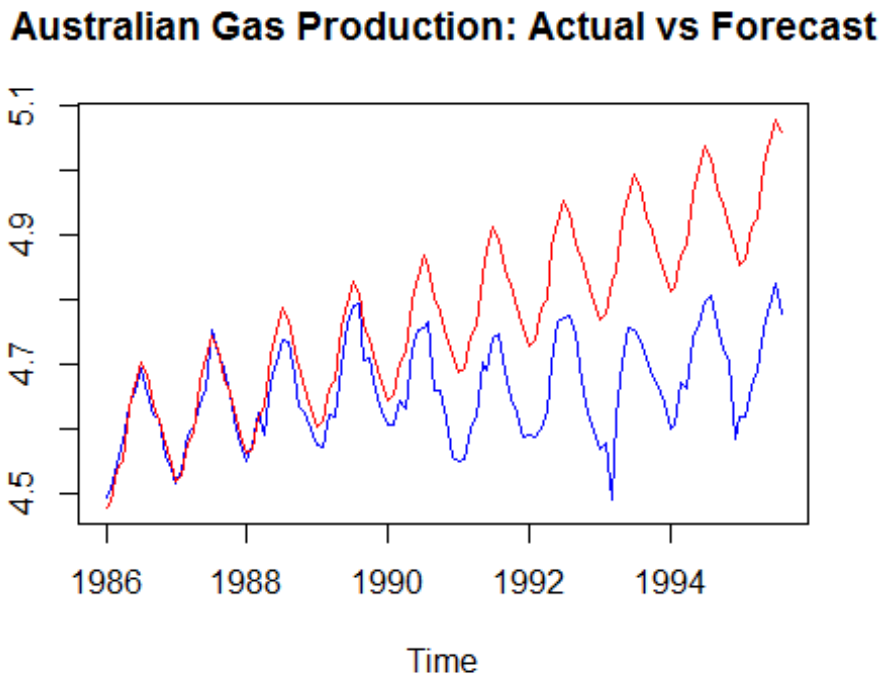


4.2. Random Walk with Drift

```
ts.decomp.train = stl(log10(ts.train), s.window = "p")
ts.train.stl = forecast(ts.decomp.train, method = "rwdrift", h = 120)
plot(ts.train.stl)
```




```
vec1 = cbind(log10(ts.test),as.data.frame(forecast(ts.decomp.train, method =
"rwdrift", h = 116))[,1])

vec = cbind(log10(ts.test),ts.train.stl$mean)

ts.plot(vec1, col = c("blue", "red"), main = "Australian Gas Production: Actu
al vs Forecast")
```



We notice that the forecasted data matches closely with the actual data from about 1986 to 1988 suggesting that the model can forecast accurately for 2 about years.

```
RMSE = round(sqrt(sum(((vec1[,1]-vec1[,2])^2)/length(vec1[,1]))),4)
RMSE

## [1] 0.1483

MAPE = round(mean(abs(vec1[,1]-vec1[,2])/vec1[,1]),4)
MAPE

## [1] 0.0252
```

Box-Ljung test: H0: Residuals are Independent Ha: Residuals are not Independent

```
Box.test(ts.train.stl$residuals, type="Ljung-Box")

##
##  Box-Ljung test
##
## data:  ts.train.stl$residuals
## X-squared = 11.156, df = 1, p-value = 0.0008378
```

Accuracy measures give us a Root Mean Square Error value of 0.1483 and a Mean Absolute Percentage Error value of 0.0252. Also the Ljung-Box test gives us a very small p-value of 0.0008378 whcih signifies that the residuals are not independant. This suggests that Random Walk with drift can give us a strong model which can forecast accurately for about 2 years the Australian Gas Production based on the given dataset.

4.3. ARIMA

Auto Regressive Induced Moving Average(ARIMA) requires a stationary time series so we check the data for stationarity using ADF test.

Augmented Dicky Fuller Test

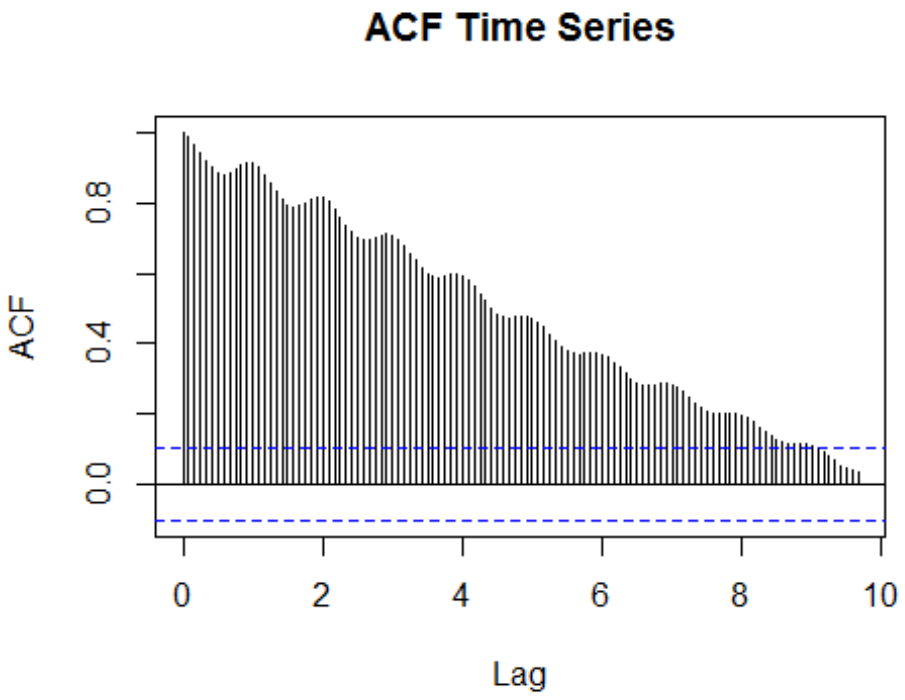
H0: Time series is non-stationary

H1: Time series is stationary

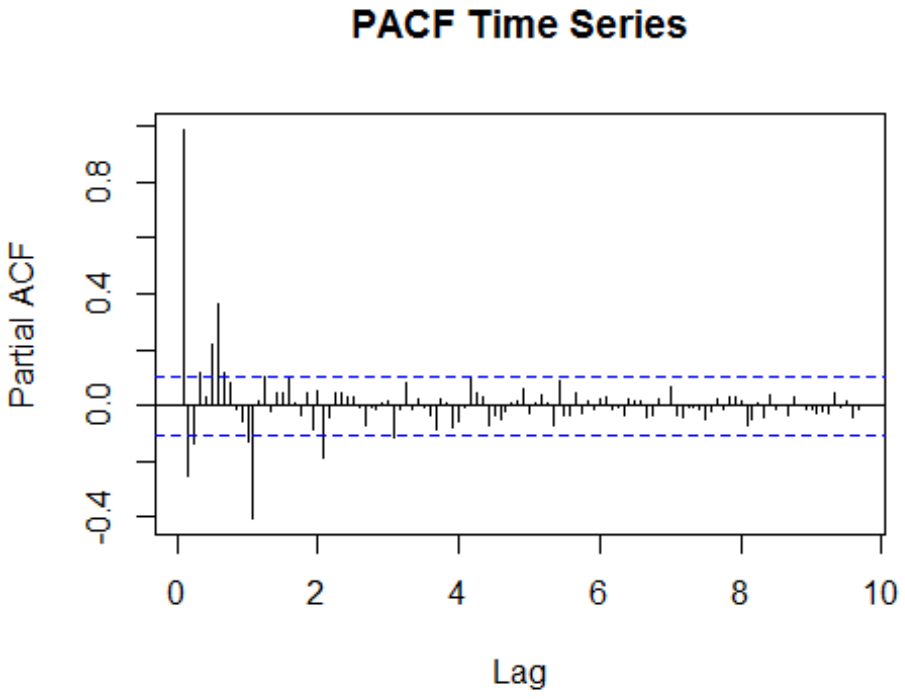
```
adf.test(ts.train)

##
##  Augmented Dickey-Fuller Test
##
## data:  ts.train
## Dickey-Fuller = -1.7308, Lag order = 7, p-value = 0.6905
## alternative hypothesis: stationary

# We check for stationarity after log transfor

ts.train.log = log10(ts.train)

adf.test(ts.train.log)

##
##  Augmented Dickey-Fuller Test
##
## data:  ts.train.log
## Dickey-Fuller = -1.7232, Lag order = 7, p-value = 0.6937
## alternative hypothesis: stationary
```

We observe that in both the cases the P-value after conducting Augmented Dicky Fuller test is much higher than 5% which leads us to conclude that the series in non-stationary.

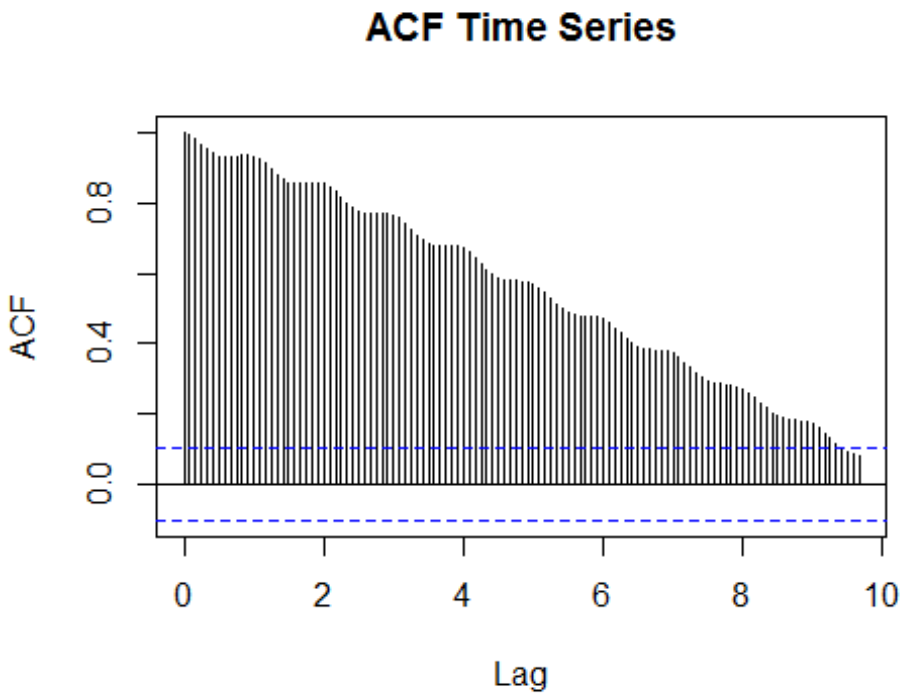
```
acf(ts.train, lag = 116, main = "ACF Time Series")
```



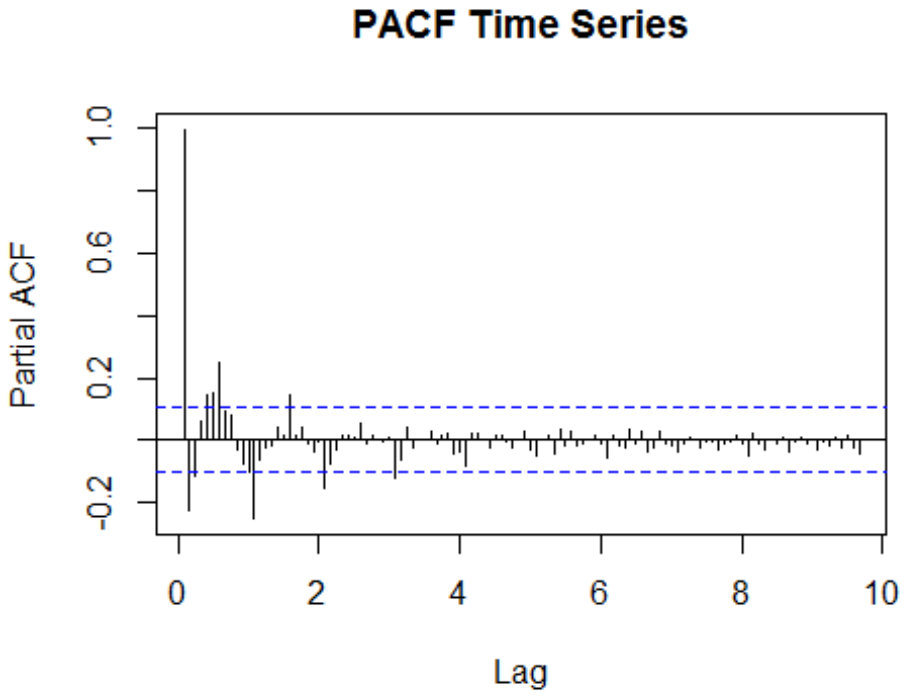

```r
pacf(ts.train, lag = 116, main = "PACF Time Series")
```




```r
acf(ts.train.log, lag = 116, main = "ACF Time Series")
```

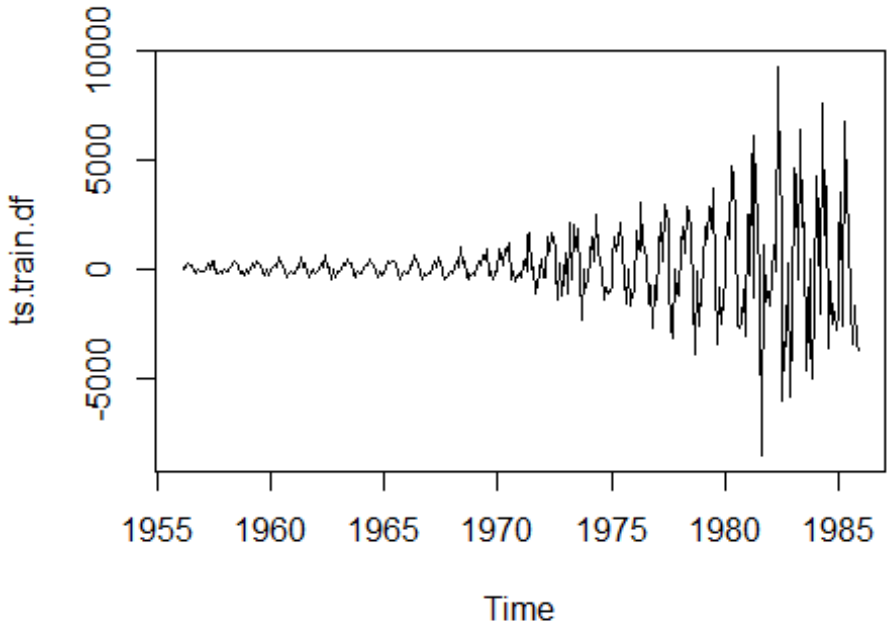



```r
pacf(ts.train.log, lag = 116, main = "PACF Time Series")
```

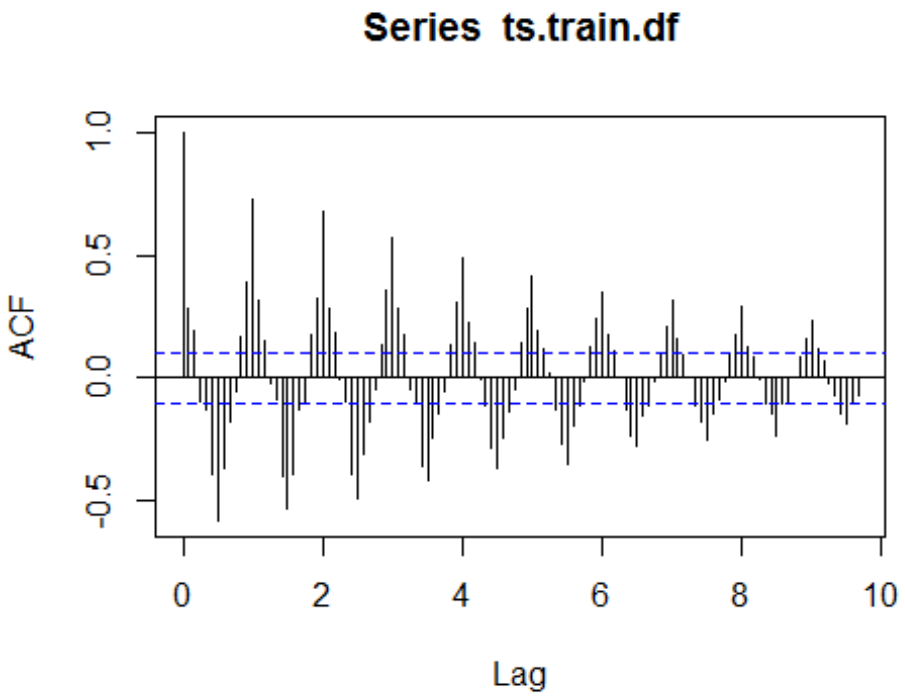


We try differencing the data to make the series stationary.

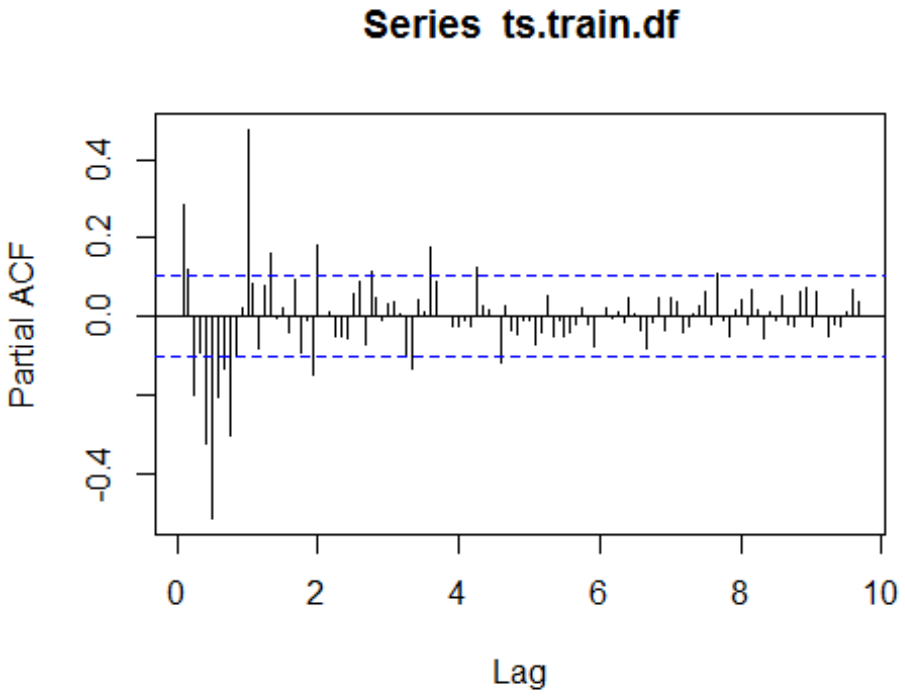
```
ts.train.df = diff(ts.train)
plot(ts.train.df)
```




```
acf(ts.train.df, lag = 116)
```




```
pacf(ts.train.df, lag = 116)
```




```
adf.test(ts.train.df)
```

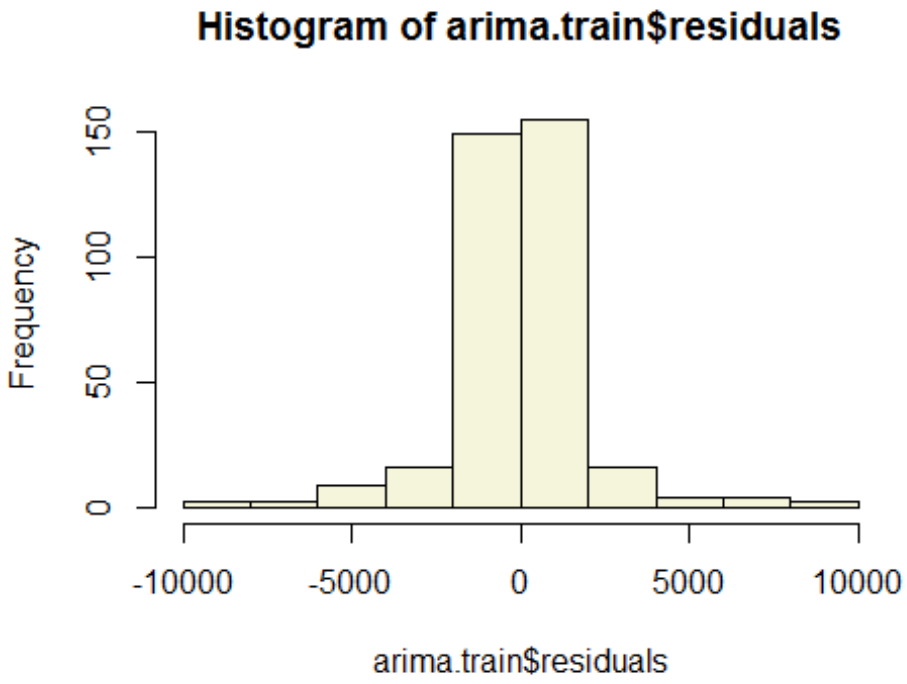
```
## Warning in adf.test(ts.train.df): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  ts.train.df
## Dickey-Fuller = -15.399, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

Autocorrelations are significant upto a very high lag value in the ACF plot. But after differencing the values change and we take q = 0. The Partial Autocorrelations seem to suggest that about 1 past observation are significant hence, we take the p value for ARIMA as 1.

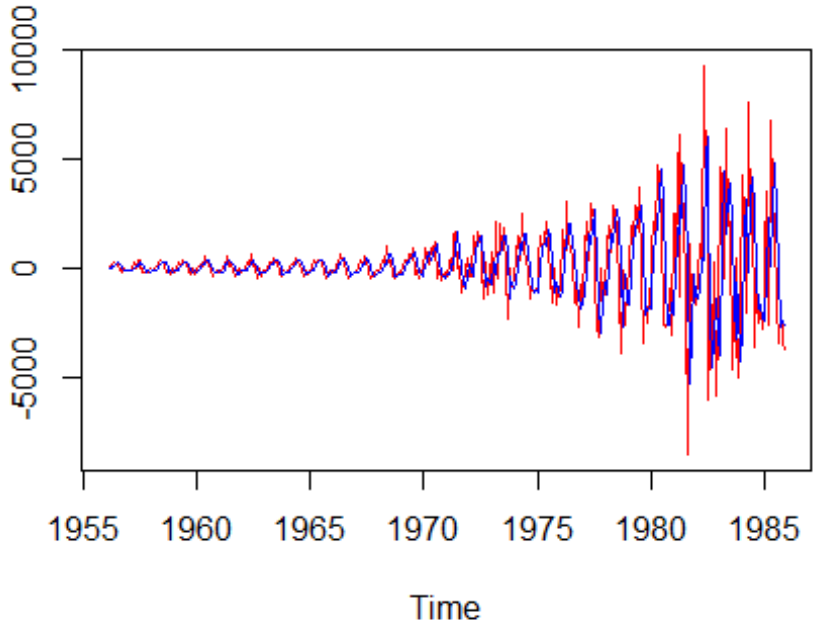
```
arima.train = Arima(ts.train.df,c(1,1,0))
arima.train
```

```
## Series: ts.train.df
## ARIMA(1,1,0)
##
## Coefficients:
##           ar1
##       -0.4360
## s.e.   0.0475
##
## sigma^2 estimated as 3700269:  log likelihood=-3214.77
## AIC=6433.53   AICc=6433.57   BIC=6441.29
```

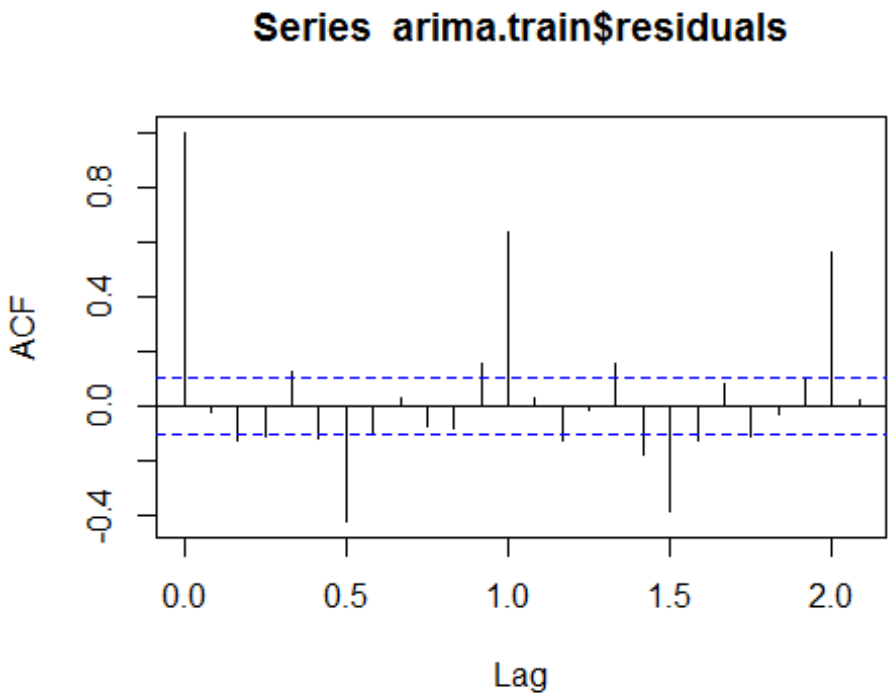
```
hist(arima.train$residuals, col = "beige")
```




```
arima.fit = fitted(arima.train)

ts.plot(ts.train.df, arima.fit, col = c("red","blue"))
```



```
acf(arima.train$residuals)
```




```
Box.test(arima.train$residuals, type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  arima.train$residuals
## X-squared = 0.2655, df = 1, p-value = 0.6064
```

We may infer from the plots that the residuals are normally distributed. The Ljung-Box test gives us a p-value of 60% whcih signifies that the residuals are independant. We try auto arima to try and get a better model.

```
autoarima.train = auto.arima(ts.train.log, seasonal = TRUE)
autoarima.train
```

```
## Series: ts.train.log
## ARIMA(0,1,1)(0,1,2)[12]
##
## Coefficients:
##           ma1     sma1     sma2
##       -0.2331  -0.6987  -0.1143
## s.e.   0.0529   0.0537   0.0537
##
## sigma^2 estimated as 0.0004108:  log likelihood=856.15
## AIC=-1704.3   AICc=-1704.18   BIC=-1688.9
```

According to the auto arima result we should use the ARIMA(1,0,1)(0,1,2)[12] model for this time series data. The lower AIC and BIC values indicate that the auto.arima model is better in terms of goodness of fit.

```
tsf = Arima(ts.train.log, order = c(0,1,1), seasonal = c(0,1,2), method ="ML"
)
tsf
```

```
## Series: ts.train.log
## ARIMA(0,1,1)(0,1,2)[12]
##
## Coefficients:
##           ma1     sma1     sma2
##       -0.2331  -0.6986  -0.1143
## s.e.   0.0529   0.0537   0.0537
##
## sigma^2 estimated as 0.0004108:  log likelihood=856.15
## AIC=-1704.3   AICc=-1704.18   BIC=-1688.9
```

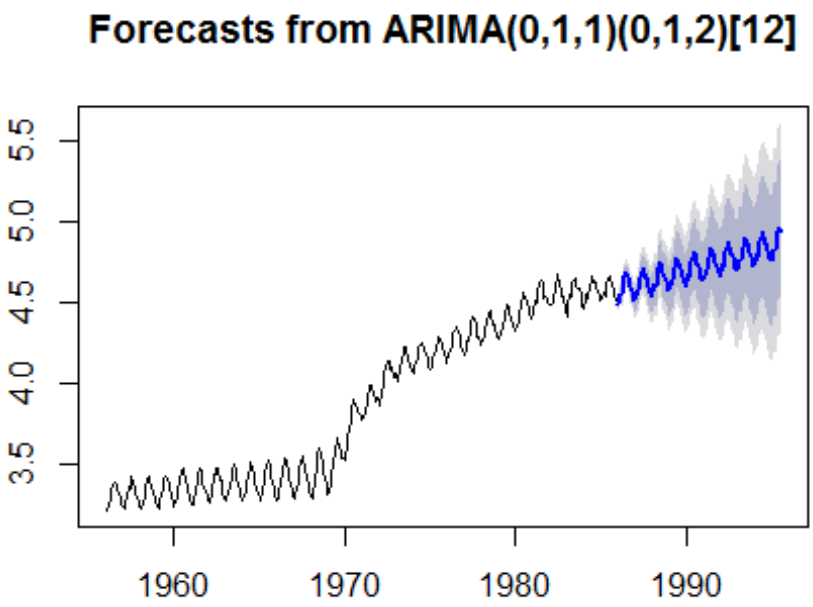
By using a seasonal ARIMA model with parameters obtained from auto.arima we get a better model fit.

```
Box.test(tsf$residuals, type="Ljung-Box")
```

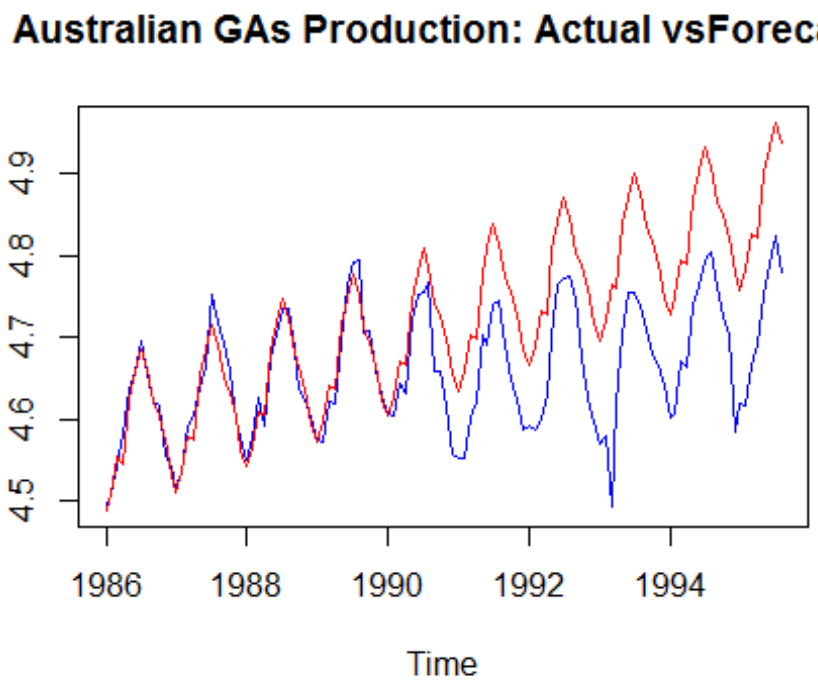
```
##
##  Box-Ljung test
##
## data:  tsf$residuals
## X-squared = 0.018126, df = 1, p-value = 0.8929
```

The Ljung-Box test gives us a p-value of 89% which signifies that the residuals are stationary.

```
ProdForecast = forecast(tsf, h = 116)
plot(ProdForecast)
```




```
vec2 = cbind(log10(ts.test),as.data.frame(forecast(tsf, h = 116))[,1])

ts.plot(vec2, col = c("blue", "red"), main = "Australian GAs Production: Actu
al vsForecast")
```



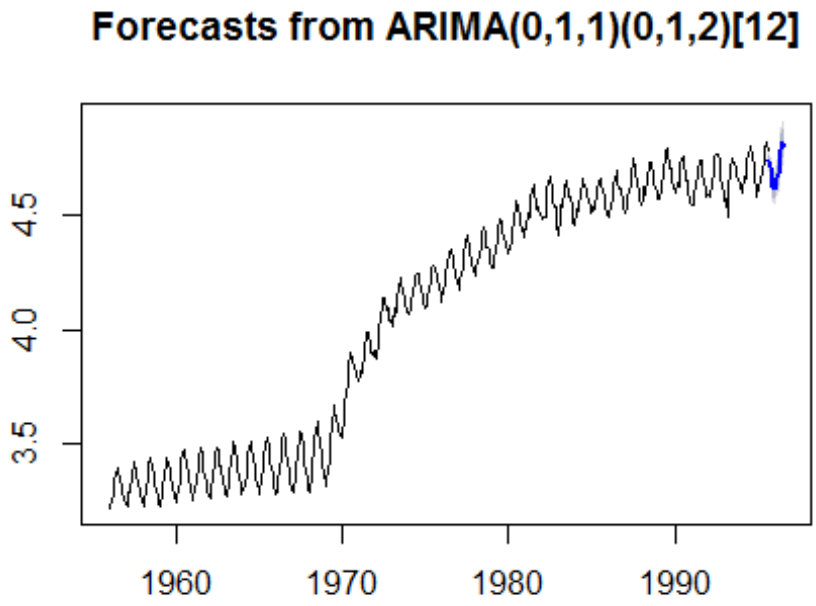

```
RMSE1 = round(sqrt(sum(((vec2[,1]-vec2[,2])^2)/length(vec2[,1]))),4)
RMSE1

## [1] 0.0895

MAPE1 = round(mean(abs(vec2[,1]-vec2[,2])/vec2[,1]),4)
MAPE1

## [1] 0.0148
```

The SARIMA model gives us a Root Mean Square error value of 0.0877 and Mean Absolute Percentage Error value of 0.0145 which suggests that this is a better model than RWDrift.

We proceed to forecast for the next 12 months using the SARIMA model.

```
ts.final.arima = Arima(log10(ts.data), order = c(0,1,1), seasonal = c(0,1,2),
method = 'ML')
ts.final.arima

## Series: log10(ts.data)
## ARIMA(0,1,1)(0,1,2)[12]
##
## Coefficients:
##           ma1     sma1     sma2
##       -0.3274  -0.7407  -0.1540
## s.e.   0.0451   0.0477   0.0478
##
```

```
## sigma^2 estimated as 0.0004679:  log likelihood=1110.6
## AIC=-2213.19   AICc=-2213.11   BIC=-2196.64

ts.final.forecast = forecast(ts.final.arima , h = 12)
plot(ts.final.forecast)
```



5. Conclusion

We find that the Seasonal ARIMA model with parameters p = 0, d = 1, q = 1 and P = 0, D = 1, Q = 2 and m = 12 gives us an accurate forecast with a MAPE value of 0.0145 which indicates very high accuracy. Using this model we have made a forecast for the period of the next 12 months which has been plotted in the diagram above.