Australian Monthly Gas Production – Time Series Forecasting

Submission Date: March 8, 2020

Author: Ayush Jain
Mentor: Deepak Gupta

Table of Content

Table of Contents

1 Project Objective

This is to understand the Monthly gas Production of Australian Gas Plant. This is an In-built dataset in R within the package Forecast.

This dataset is of the Class “Time Series” with 476 values and 1 variable

Using the given data, we are expected to forecast the Production of the of the Gas Plant for next year. To do so, we will first split the data into Train and Test and perform different Modelling techniques on train and evaluate those by forecasting on test set (Unseen data).

The data given is a monthly data from January 1956 to August 1995.

We are required to perform the below tasks on the dataset:

- Read the data as a time series object in R and 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 initial forecast for next 20 periods. Check the same using the various metrics, after finalising the model, develop a final forecast for the 12 time periods. Use both manual and auto.arima.
- Report the accuracy of the model.

We will be performing ARIMA and Auto ARIMA Model on the dataset.

2 Exploratory Data Analysis – Step by step approach

Exploratory Data Analysis is one of the important phases in the data Analysis in understanding the significance and accuracy of the data. It usually consists of setting up the environment to work in R, loading the data and checking the validity of data loaded.

A Typical Data exploration activity consists of the following steps:

- Environment Set up and Data Import.
    - Install Necessary Package in R.
    - Reading the Dataset in R environment.
    - Performing Basic EDA Steps.
    - Performing Outlier and Null Value check.
    - Performing Univariate Analysis.
    - Preparing the Data for Model Building.

- Variable Identification.

We shall follow these steps in exploring the provided dataset.

2.1 Environment Set up and Data Import

2.1.1 Deploying necessary Packages in R.

In this section, we will install and invoke the necessary Packages and Libraries that are going to be the part of our work throughout the project. Having all the packages at the same places increases code readability and Understandability.

```
#Installing required packages

install.packages("forecast")
install.packages("tseries")
install.packages("dygraphs")
install.packages("TTR")
install.packages("xts")

library(forecast)
library(tseries)
library(dygraphs)
library(TTR)
library(xts)
library(TSstudio)
```

2.1.2 Reading Dataset in R Environment.

The given dataset is an inbuild dataset in R under forecast package. Hence, we will assign the gas dataset to the local R environment.

```
#Reading the Dataset

USGas<- forecast::gas
```

2.1.3 Performing basic EDA Steps.

This section of the report checks for the basic steps to ensure that the data is imported properly and also checks the Structure of the dataset and Summary to have the basic understanding of the Data.

```
> class(USGas)
[1] "ts"
> str(USGas)
 Time-Series [1:476] from 1956 to 1996: 1709 1646 1794 1878 2173
 ...
> summary(USGas)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   1646    2675   16788   21415   38629   66600 
```

Checking the Periodicity of the data.

```
#Checking the periodicity of the Gas Data.

periodicity(USGas)
```

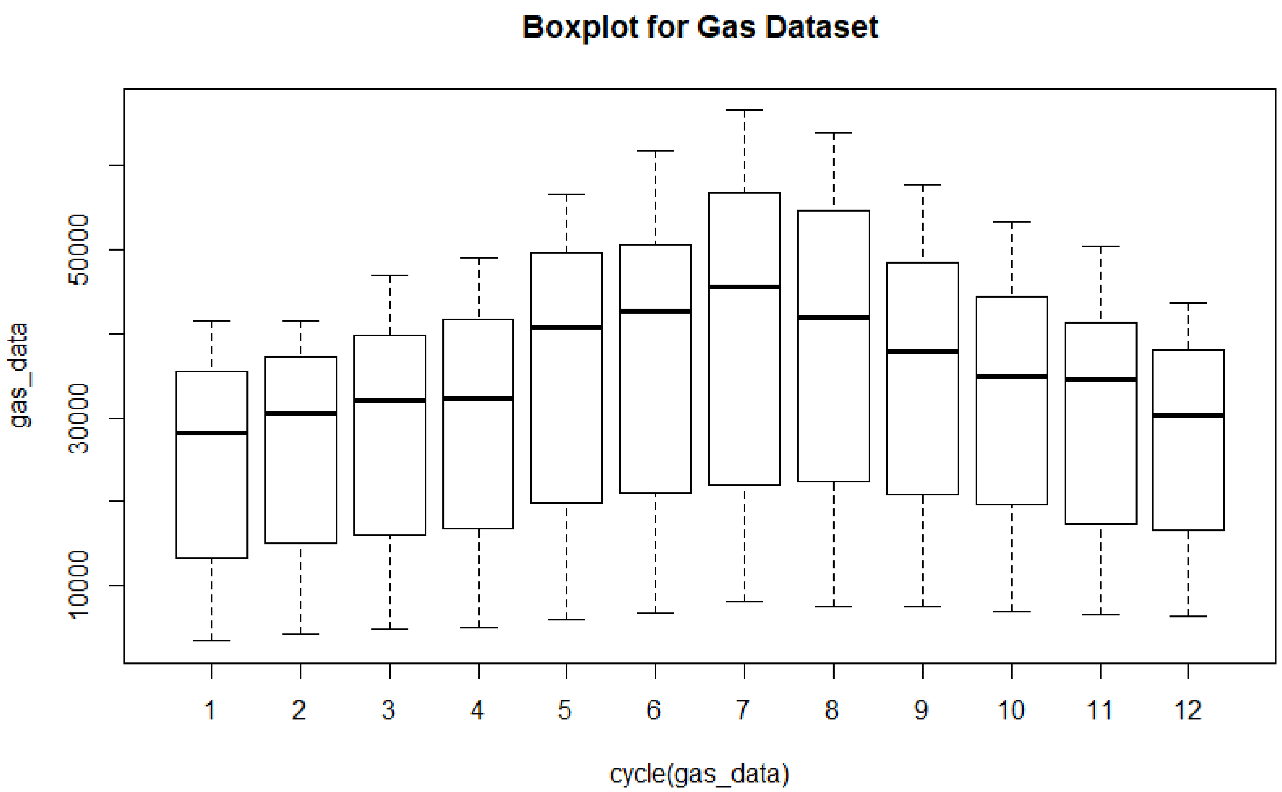
```
> periodicity(USGas)
Monthly periodicity from Jan 1956 to Aug 1995
```

The data is the Monthly Data.

2.1.4 Checking for Outliers and Null Values in the Dataset.

Outliers: An outlier is an observation that lies an abnormal distance from other values in a random sample from a population. Examination of the data for unusual observations that are far removed from the mass of data. These points are often referred to as outliers.

```
# Plotting the box Plot to check the Outliers.

boxplot(gas_data~cycle(gas_data), main = "Boxplot for Gas Dataset")
```



From the above Box Plot, we see that their exist no Outliers in the above dataset.

Null Values: These are the missing values in the dataset that needs to be treated to get the proper accuracy of the Model.

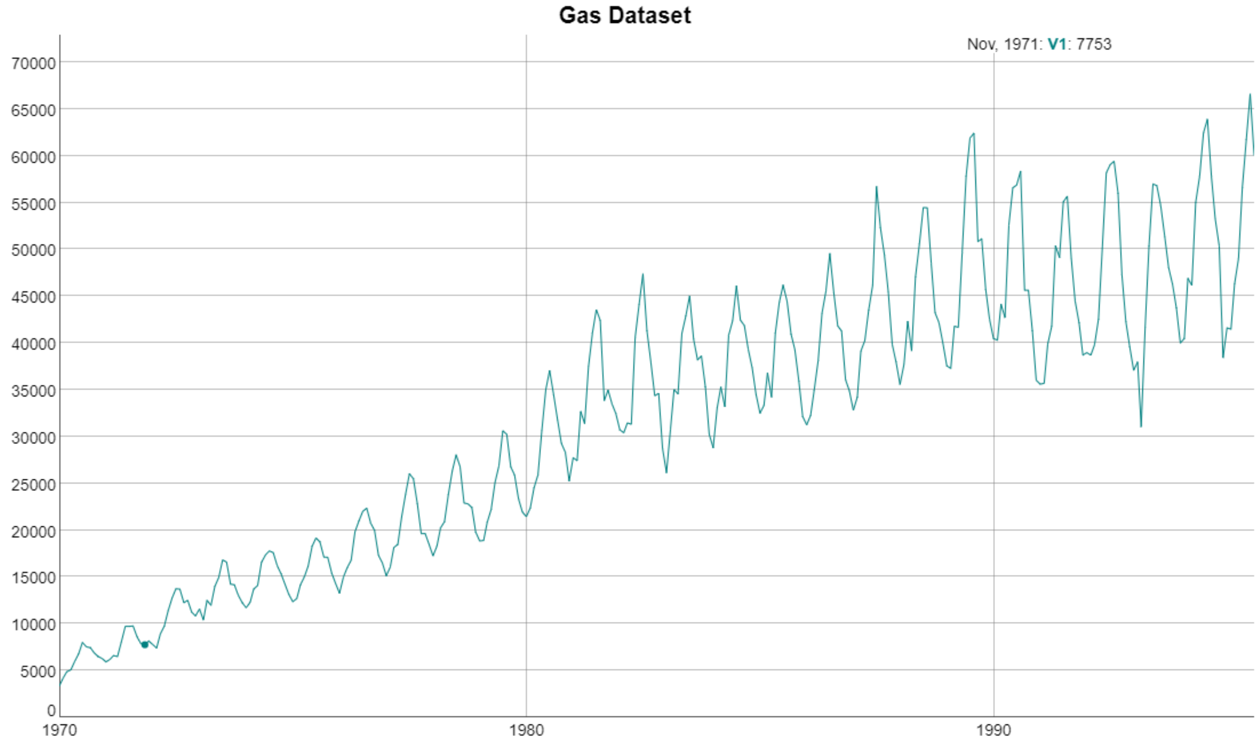
```
> sum(is.na(USGas))
[1] 0
```

There exist no Null values in the Series.

2.1.5 Performing Univariate Analysis on independent variables.

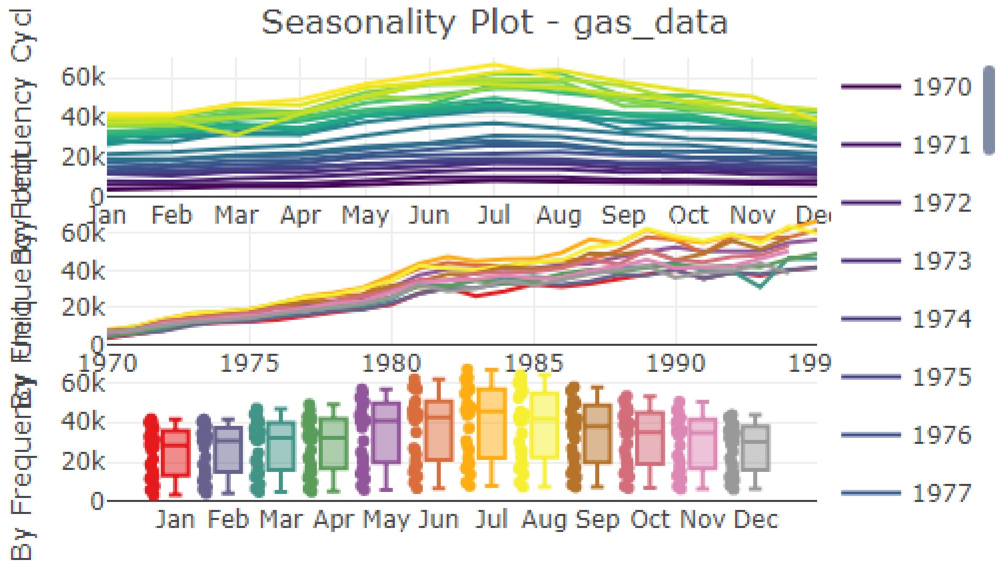
Univariate analysis is perhaps the simplest form of analysis. Like other forms of statistics, it can be inferential or descriptive. The key fact is that only one variable is involved.

- **Plot on Original Dataset.**



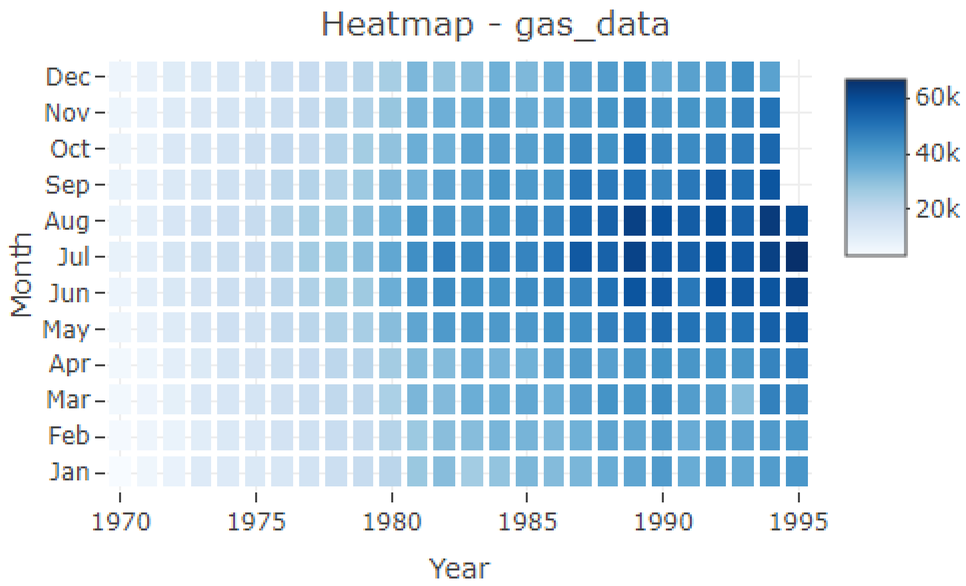
From the above plot we can see that there exists the trend in the data, making it a non-Stationary series. We will also be checking if there exists any Seasonality.

- **Plotting the Seasonal Pot on Original Dataset.**



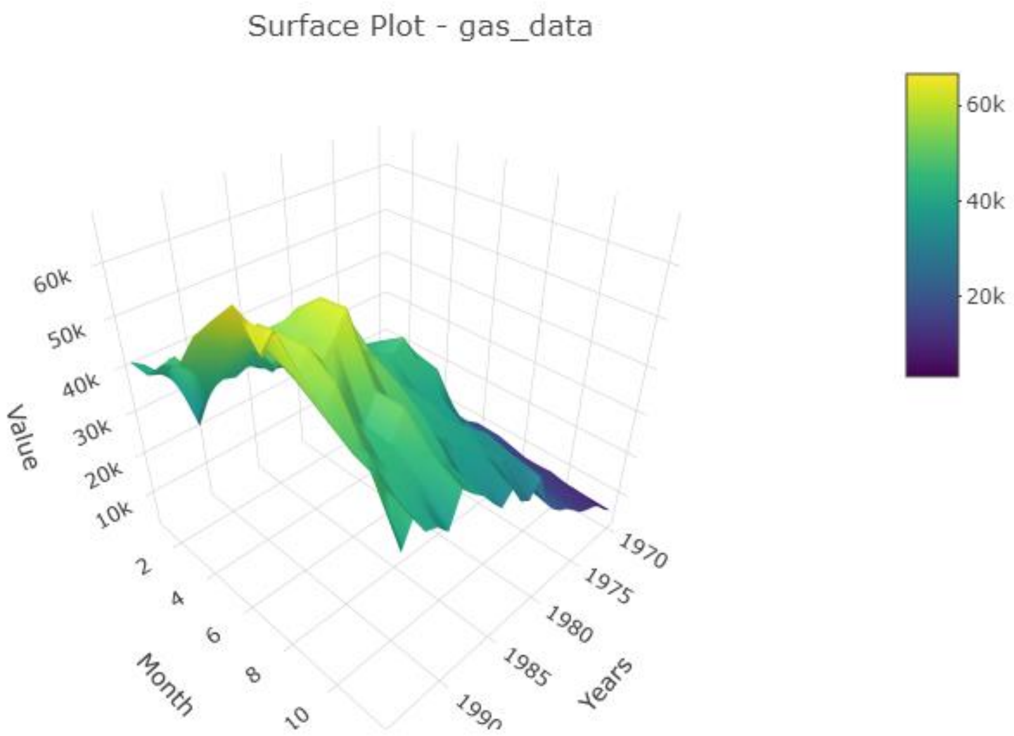
As we look at the above plot, we found that the Structure of the pattern is distributed across, hence there is no Seasonality present in the data.

- **Plotting the Heat Map on the Original Dataset.**



From the above Heat Map, it is evident that the data doesn't have seasonality present as we do not see any repeated values across year for any of the Month.

- **Plotting the Surface Map on the Original Dataset.**



2.1.6 Preparing the Data for Model Building.

In this part of the report we will be preparing the data for Modelling purpose.

As we have been asked, we subset the data from January 1970. We will further be using this data during our further functioning.

Once the data is handy, we check if the Series is Stationary.

"A stationary time series is one whose statistical properties such as mean, variance, autocorrelation, etc. are all constant over time."

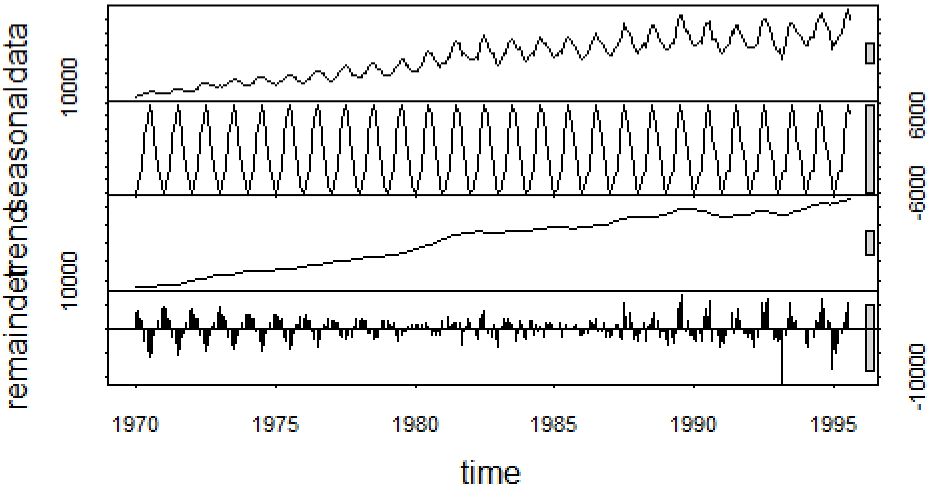
To check if the Series is Stationary we use the below two approaches.

- Plotting Decomposition. (Visual approach)
- Performing Augmented Dickey Fuller Test (ADF Test).

Approach 1:

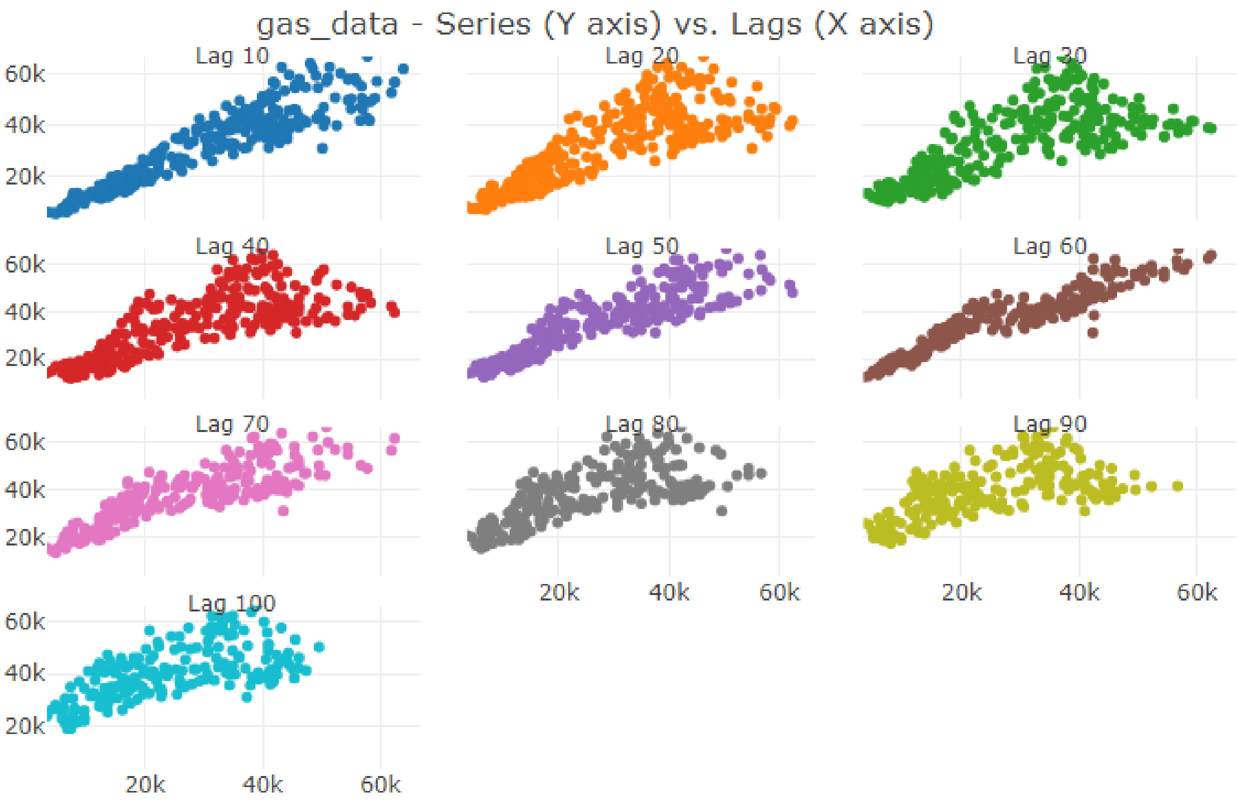
In this we plot the decomposition graph to check if there exist any Trend or Seasonality in the series.

```
#Plotting the decompositions of data.
gas.ts <- stl(gas_data, s.window = "periodic")
plot(gas.ts)
```



From the Plot, we see that there is **no Sign of Seasonality but there exist the Trend** in the data making the Series Non- Stationary.

```
ts_lags(gas_data, lags = c(10,20,30,40,50,60,70,80,90,100))
```



Approach 2:

In this approach, we perform the ADF Test on the data, which helps us Identify if the data is Stationary. We perform the Hypothesis testing of the data.

Null Hypothesis; H0 = Series is Non- Stationary

Alternate Hypothesis; H1 = Series is Stationary

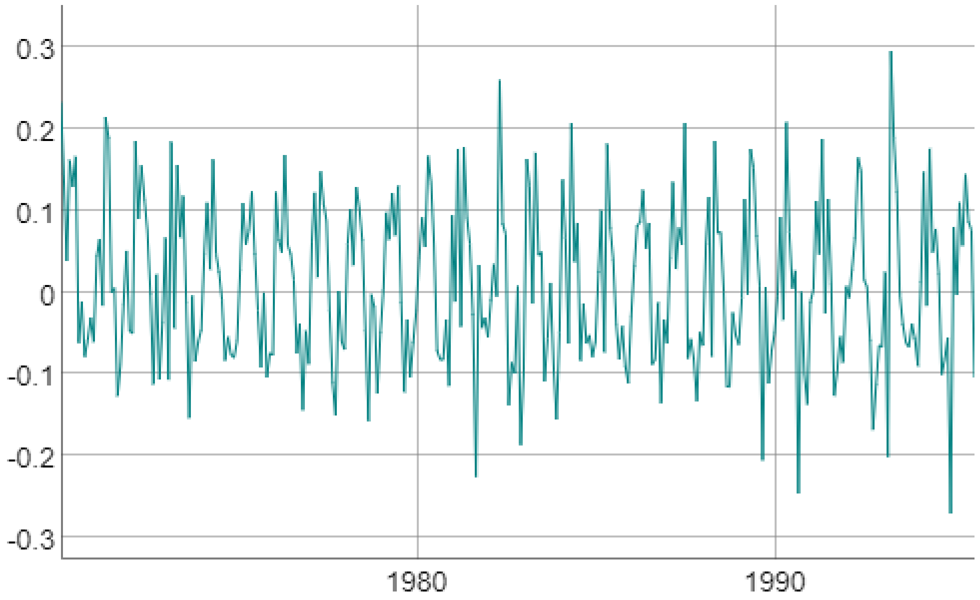
```
#Checking the Stationarity of the data.

adf.test(gas_data,alternative = "stationary", k=12)

        Augmented Dickey-Fuller Test

data:  gas_data
Dickey-Fuller = -1.3538, Lag order = 12, p-value =
0.8488
alternative hypothesis: stationary
```

From the above test we get the p-value of 0.84, which is greater than alpha (0.05). Hence we can’t reject the Null Hypothesis, which means that the series is Non-Stationary.

Stationarizing the series.

To make the Series Stationary, we remove the trend from the data. This can be done by differencing the data.

```
#As we see that the data have trend, we difference the data.

diff.gas.data<- diff(log(gas_data))

dygraph(diff.gas.data)
```



From the above plots, we see that the is no Trend available.

Performing ADF Test.

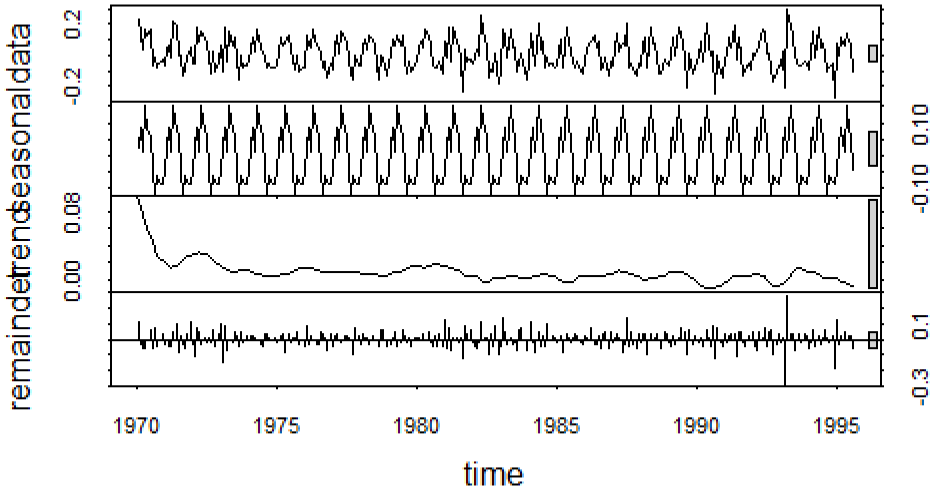
```
#Performing the ADF Test on differenced data

adf.test(diff.gas.data,k=12)

        Augmented Dickey-Fuller Test

data:  diff.gas.data
Dickey-Fuller = -5.4776, Lag order = 12, p-value =
0.01
alternative hypothesis: stationary
```

As we see that the p-value came down to 0.01 which is less then alpha, hence we accept the Alternate Hypothesis which is **"Series is Stationary"**

From the test, we have achieved the p-value of 0.01 which is lesser then alpha (0.05) and hence we Reject the Null Hypothesis and accept the Alternate Hypothesis i.e. Series is Stationary.

Plotting the Decomposition.

We plot the decomposition to visually check if there exist any Trend or Seasonality in the data.

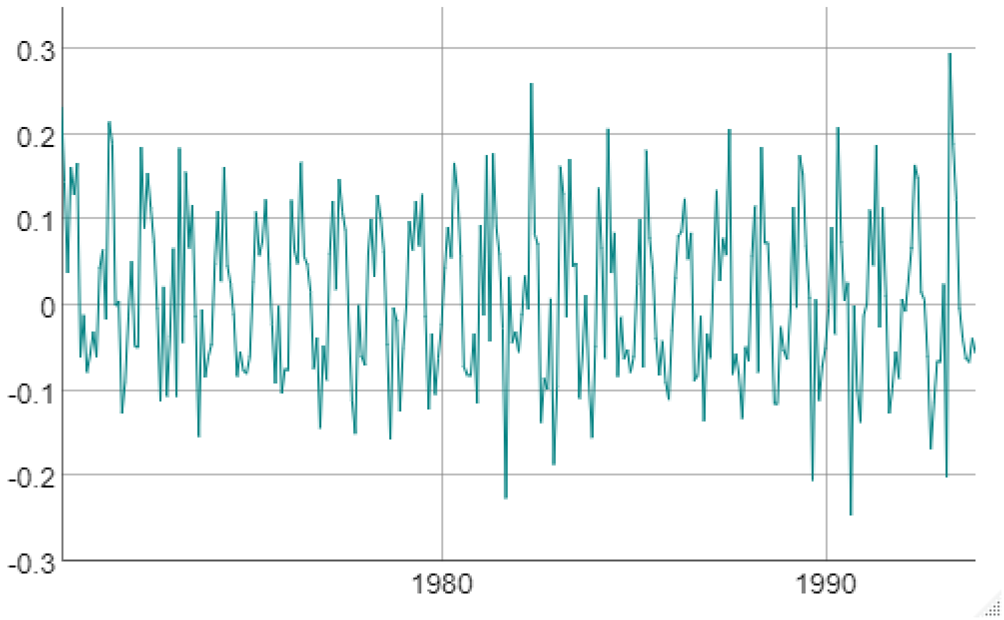


We see that there exist no Trend in the differenced data hence the Series is Stationary.

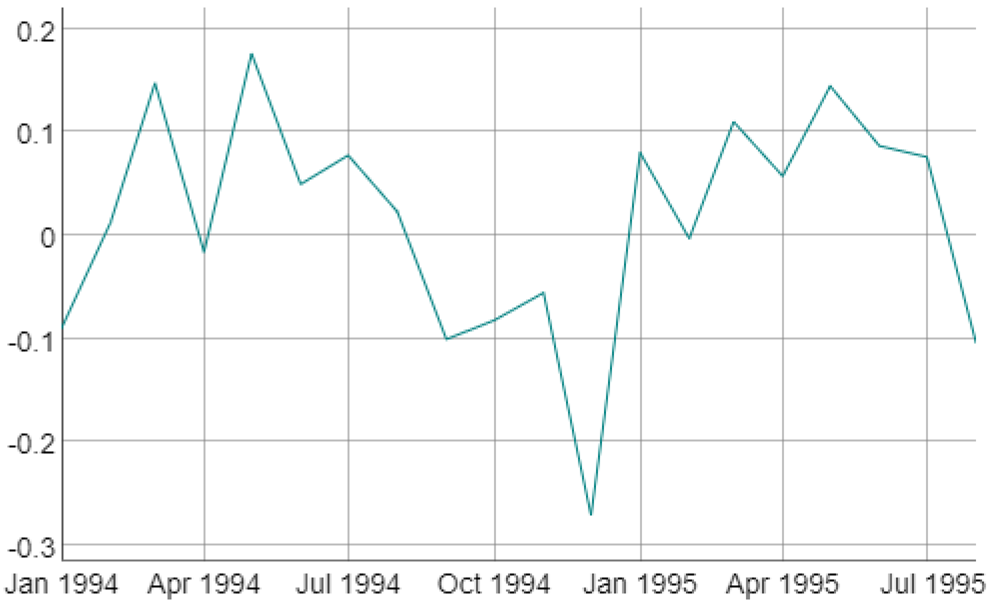
Now when the Series is Stationary, we can build the models on this data. Before we build the model, we split the series to Train and Test. This is done using "window" function.

We divide the Series into two parts where data from 1970 to 1993 is considered as Train and the rest is considered as Test.

```r
#Splitting the data into train and test.

train <- window(diff.gas.data, end =c(1993,12), frequency =12)
dygraph(train)
```




```r
test <- window(diff.gas.data,start = c(1994,1), frequency=12)
dygraph(test)
```



Now, the data is ready for Model building.

2.2 Variable Identification

This section holds the Methods that are used during the Analysis of the problem. Below are the Functions that we have used for the Analysis.

- **library()**

  `library` and `load` and attach add-on packages

- **install.packages()**

  Download and install packages from CRAN-like repositories or from local files.

- **class()**

  R possesses a simple generic function mechanism which can be used for an object-oriented style of programming. Method dispatch takes place based on the class of the first argument to the generic function.

- **str()**

  Compactly display the internal Structure of an R object.

- **summary()**

  Summary is a generic function used to produce result summaries of the results of various model fitting functions.

- **is.null()**

  NULL is often returned by expressions and functions whose value is undefined. is.null returns TRUE if its argument's value is NULL and FALSE otherwise.

- **boxplot()**

It is plotting technique, which is used to identify if there any outliners are present in the data.

- **periodicity()**

  Estimate the periodicity of a time-series-like object by calculating the median time between observations in days.

- **window()**

  `window` is a generic function which extracts the subset of the object `x` observed between the times `start` and `end`. If a frequency is specified, the series is then re-sampled at the new frequency.

- **dygraph()**

  R interface to interactive time series plotting using the dygraphs JavaScript library.

- **ts_seasonal()**

  Visualize time series object by it periodicity, currently support time series with daily, monthly and quarterly frequency

- **ts_heatmap()**

  Heatmap plot for time series object by it periodicity

- **ts_surface()**

  3D surface plot for time series object by it periodicity

- **stl()**

  Decompose a time series into seasonal, trend and irregular components using `loess`, acronym STL.

- **adf.test()**

  Computes the Augmented Dickey-Fuller test for the null that `x` has a unit root.

- **diff()**

  Returns suitably lagged and iterated differences.

- **arima()**

  Fit an ARIMA model to a univariate time series.

- **box.test()**

  Compute the Box–Pierce or Ljung–Box test statistic for examining the null hypothesis of independence in a given time series. These are sometimes known as ‘portmanteau’ tests.

- **test_forecast()**

  Visualize the fitted values of the training set and the forecast values of the testing set against the actual values of the series

- **auto.arima()**

  Returns best ARIMA model according to either AIC, AICc or BIC value. The function conducts a search over possible model within the order constraints provided.

3 Conclusion

Once the Data is ready and divided into the Train and Test, we will further perform ARIMA and AUTO.ARIMA on the Train data and evaluating it on the Test data.

3.1 ARIMA Model

In statistics and econometrics, and in particular in time series analysis, an **autoregressive integrated moving average (ARIMA)** model is a generalization of an autoregressive moving average (ARMA) model. Both of these models are fitted to time series data either to better understand the data or to predict future points in the series (forecasting). ARIMA models are applied in some cases where data show evidence of non-stationarity, where an initial differencing step (corresponding to the "integrated" part of the model) can be applied one or more times to eliminate the non-stationarity.

3.1.1 <u>Building Manual ARIMA Model on Train Data</u>

To perform Manual ARIMA, we are required to identify the below parameters.

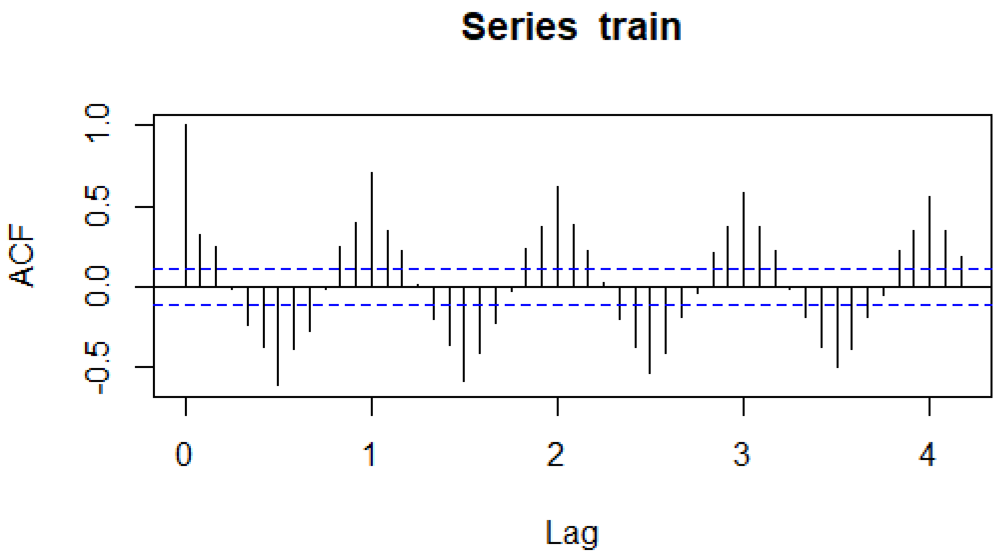
p: Auto Regressive Order, achieved using pacf

d: Degree of Differencing, Number of times differenced

q: Moving Average order, achieved using acf plot

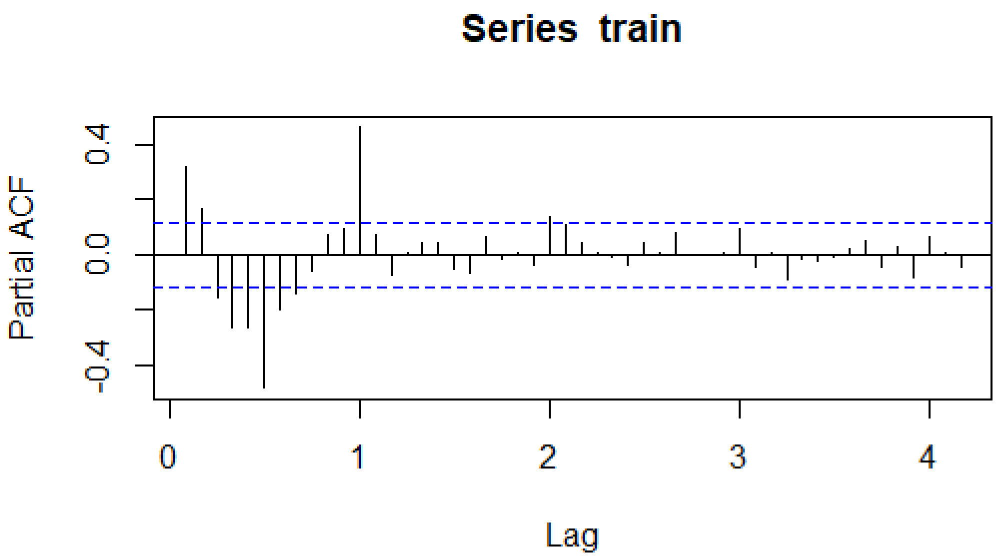
<u>Plotting ACF</u>

```
# Performing ACF test to identify q.
acf(train, lag.max = 50)
```



Plotting PACF

```
# Performing PACF to identify p.
pacf(train, lag.max = 50)
```



From the above plots we found the value of q=3, p=2 and d=1.

As we have received the values of p,d,q we will build the model on the same.

```
# Performing Manual arima on calculated p and q.
gasArima <- arima(exp(train), order = c(2,1,3))
summary(gasArima)
```

```
Call:
arima(x = exp(train), order = c(2, 1, 3))

Coefficients:
         ar1      ar2      ma1     ma2      ma3
      1.7242  -0.9860  -2.8038  2.7604  -0.9526
s.e.  0.0113   0.0109   0.0326  0.0633   0.0320

sigma^2 estimated as 0.00462:  log likelihood = 357.06,  aic = -702.11

Training set error measures:
                       ME       RMSE         MAE       MPE
Training set -0.009200633 0.06785293 0.05194797 -1.320767
                 MAPE      MASE       ACF1
Training set 5.198202 0.5756159 -0.2574102
```

From the Model above, we achieved the AIC value of -702.11 which is too low. We will further perform various performance models on this to test the efficiency of the Model.

Performing Ljung Box test.

This test is performed to check if the Residuals are Independent. The Hypothesis for this test will be:

H0: Residuals are independent

Ha: Residuals are not independent

```
# Perforing Ljung Test to check if the residuals are independent.

Box.test(exp(gasArima$residuals))


        Box-Pierce test

data:  exp(gasArima$residuals)
X-squared = 1.388, df = 1, p-value = 0.2387
```

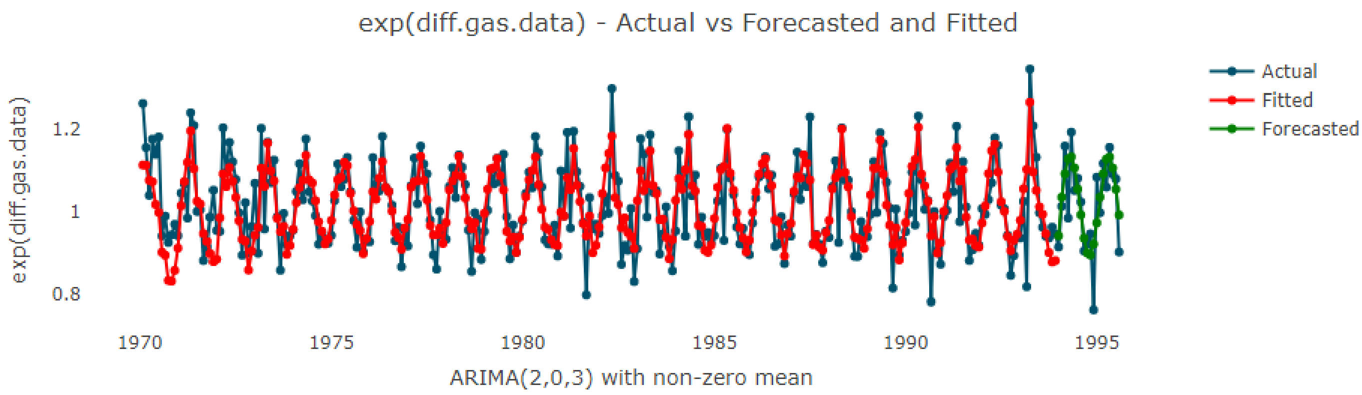
The p-value for the test is 0.2 which is larger than alpha (0.05), hence we Accept the Null Hypothesis i.e. Residuals are Independent.

Performing the accuracy on Test Data

```
# Performing accuracy on Test data.
forecast_arima <- forecast(gasArima, h=20)

accuracy(forecast_arima, exp(test))

                        ME       RMSE        MAE        MPE    MAPE      MASE        ACF1 Theil's U
Training set  0.0006422578 0.06111115 0.04616250 -0.2717222 4.61010 0.8790939  0.05126293        NA
Test set     -0.0102156701 0.06779089 0.05440361 -1.4820666 5.57312 1.0360332 -0.68492823 0.4822607
```

From the above accuracy test, we got the Mean Absolute Percentage Error for **Train as 4.6%** and for **Test as 5.5%** which can be considered.

Plotting the Actual vs Fitted vs Forecast



The forecasted values are quite similar to that of the Actual values for the test data with a bit of variation of approximately 5%.

We will now be performing the AUTO ARIMA and compare the performance of both the Models.

3.2 AUTO ARIMA Model

Auto Arima is the similar to Arima except that it Automatically defines the value of p, d, q for the model building.

3.2.1 Building Auto Arima Model

```
#Performing Auto Arima

gasAuto <- auto.arima(exp(train),seasonal = TRUE)
summary(gasAuto)
```

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

Coefficients:
         ar1      ma1    sar1    sar2     sma1
      0.1168  -0.5452  0.1913  0.0349  -0.8869
s.e.  0.1305   0.0998  0.0945  0.0841   0.0749

sigma^2 estimated as 0.003012:  log likelihood=403.54
AIC=-795.07   AICc=-794.76   BIC=-773.37

Training set error measures:
                     ME       RMSE        MAE       MPE     MAPE
Training set -0.0112713 0.05323167 0.03857387 -1.336179 3.872699
                  MASE        ACF1
Training set 0.7345801 -0.02972483
```

From Auto Arima, we achieved the AIC value of -795.07 and the Mean Absolute Percentage Error of 3.8%

Performing Ljung -Box Test.

This test is performed to check if the Residuals are Independent. The Hypothesis for this test will be:

H0: Residuals are independent

Ha: Residuals are not independent

```
        Box-Pierce test

data:  gasAuto$residuals
X-squared = 0.25358, df = 1, p-value = 0.6146
```

p value calculated is 0.6, which is greater than alpha (0.05) hence we accept the Null Hypothesis i.e. Residuals are Independent.

Predicting the accuracy on Test data.

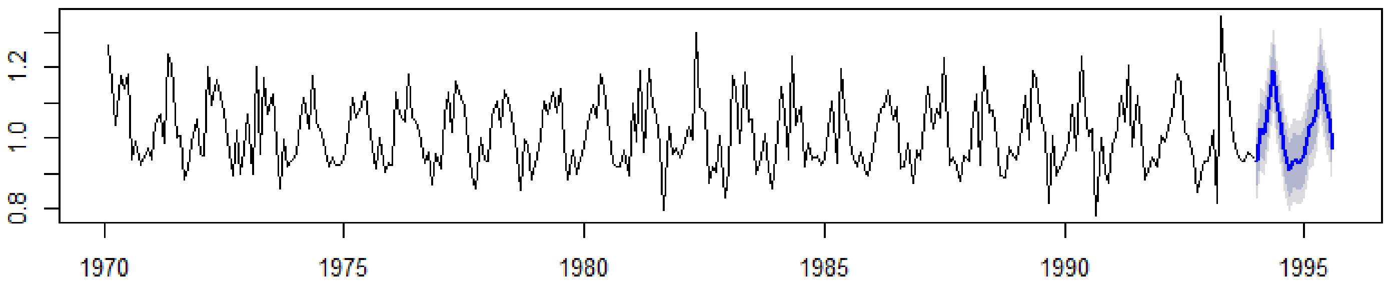
```
#Predicting the accuracy on test.

forecast_auto <- forecast(gasAuto, h=20)
accuracy(forecast_auto, exp(test))
```

```
                       ME       RMSE        MAE       MPE     MAPE      MASE        ACF1 Theil's U
Training set -0.011271298 0.05323167 0.03857387 -1.336179 3.872699 0.7345801 -0.02972483        NA
Test set     -0.005177004 0.07248664 0.05187003 -1.013439 5.261249 0.9877851 -0.52404102 0.5281561
```

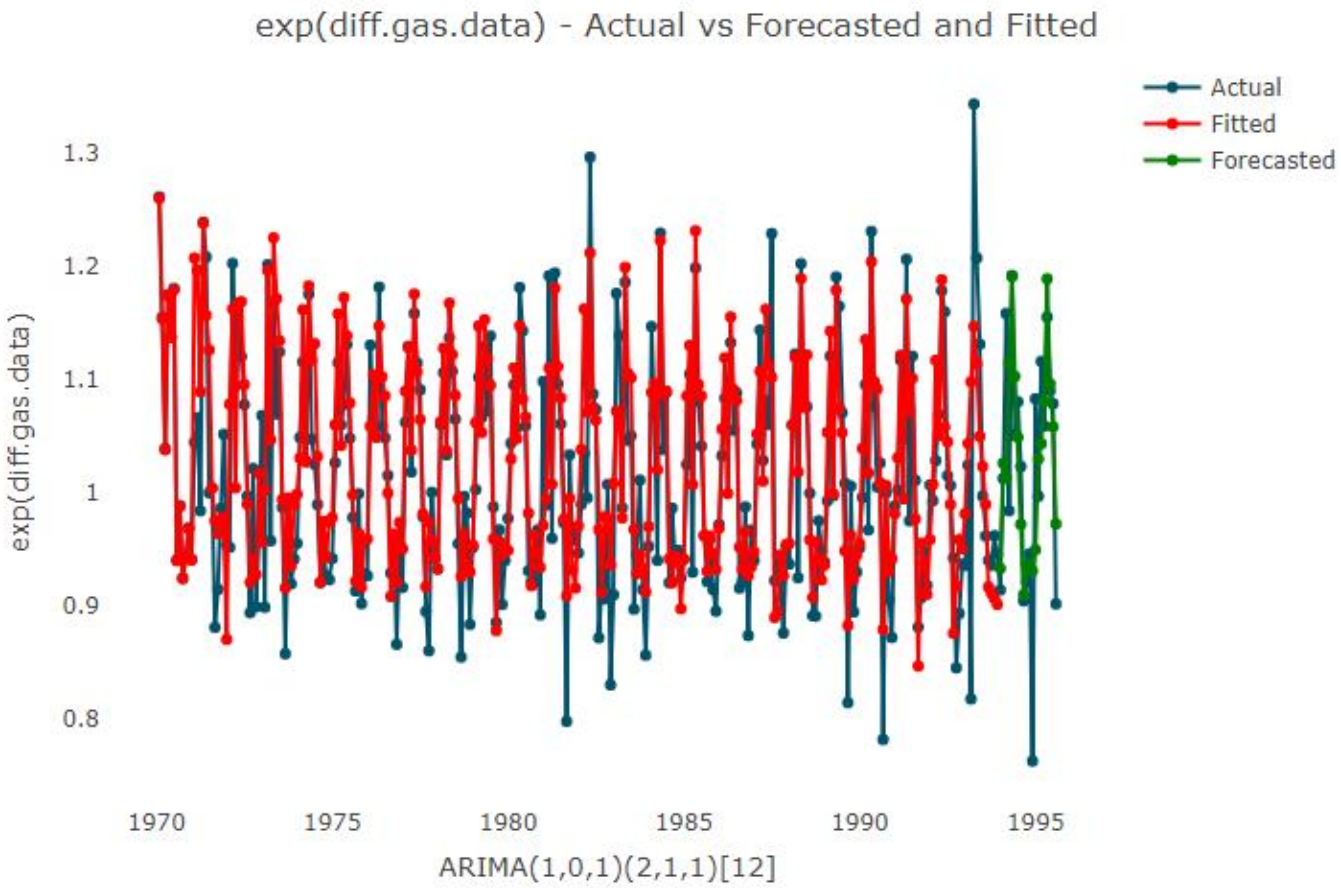
From the above accuracy test, we got the Mean Absolute Percentage Error for **Train as 3.6%** and for **Test as 5.2%** which can be considered.

Plotting the forecast

Forecasts from ARIMA(1,0,1)(2,1,1)[12]



Plotting the Actual vs Fitted vs Forecast



The forecasted values are for fitted to the Actual values for the test data with an error rate of approximately 5%.

Conclusion

Now when we have both the Models created, we compare both the Models and choose the best Model to for future forecast if next 12 Months i.e. Sep 1995 to Sep 1996.

	AIC Value	MAPE_Train	MAPE_Test
ARIMA	-763.01	4.6	5.5
AUTO ARIMA	-795.07	3.8	5.2

As we know that AIC is lower the better, AIC value of Auto Arima is lower than the Arima Model. We see that the Mean Absolute Percentage Error(MAPE) is reduced in Auto Arima. Hence we consider that Auto ARIMA is a more suitable Model.

Forecasting the values for Next 12 Months

```
# Building the Model on the Original Dataset.

auto_original <- auto.arima(gas_data, seasonal = TRUE)
summary(auto_original)

Series: gas_data
ARIMA(1,0,1)(0,1,1)[12] with drift

Coefficients:
         ar1      ma1     sma1     drift
      0.9021  -0.4419  -0.5889  152.9030
s.e.  0.0344   0.0782   0.0508   23.1751

sigma^2 estimated as 3975987:  log likelihood=-2669.78
AIC=5349.55   AICc=5349.76   BIC=5368.01

Training set error measures:
                   ME     RMSE     MAE         MPE     MAPE
Training set 4.501012 1941.505 1314.97 -0.09114548 4.108932
                  MASE       ACF1
Training set 0.4779848 0.01710761
```

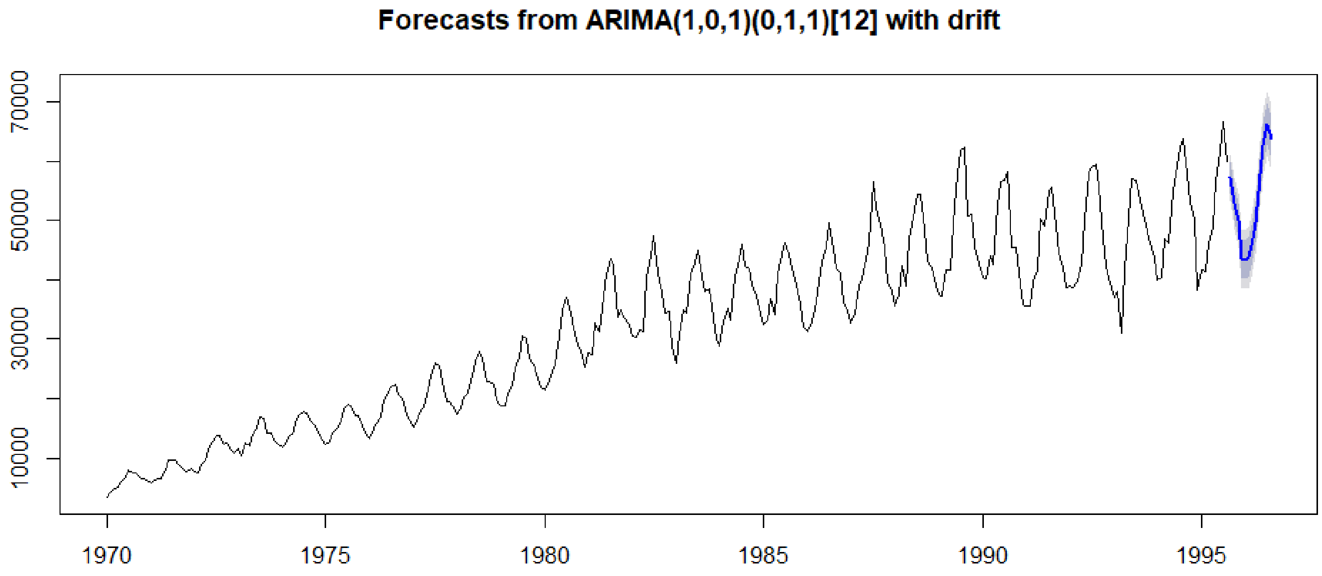
AIC= 5349.55

MAPE = 4.10

```
#Perfroming Ljung test

Box.test(auto_original$residuals)

        Box-Pierce test

data:  auto_original$residuals
X-squared = 0.090142, df = 1, p-value = 0.764
```

Since the p-value is 0.7 which is greater than alpha, we accept the Null hypothesis i.e. Residuals are Independent.

```r
#Forecasting for next 12 Months

next_forecast <- forecast(auto_original, h=12)
next_forecast

         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Sep 1995       57315.51 54760.11 59870.91 53407.36 61223.65
Oct 1995       52996.45 50183.48 55809.42 48694.38 57298.52
Nov 1995       50052.65 47046.31 53058.98 45454.85 54650.44
Dec 1995       43502.53 40347.57 46657.48 38677.43 48327.62
Jan 1996       43469.40 40198.47 46740.32 38466.95 48471.84
Feb 1996       43753.25 40390.90 47115.59 38610.99 48895.51
Mar 1996       46953.27 43518.33 50388.21 41699.98 52206.56
Apr 1996       49729.49 46236.58 53222.40 44387.55 55071.44
May 1996       57929.47 54390.09 61468.85 52516.45 63342.49
Jun 1996       62638.16 59061.40 66214.91 57167.99 68108.33
Jul 1996       66353.93 62747.05 69960.81 60837.69 71870.18
Aug 1996       63799.59 60168.38 67430.80 58246.13 69353.05

plot(next_forecast)
```



4 Appendix A – Source Code

```
#-------------- Project 6 -------------------#

#Installing required packages

install.packages("forecast")
install.packages("tseries")
install.packages("dygraphs")
install.packages("TTR")
install.packages("xts")

library(forecast)
library(tseries)
library(dygraphs)
library(TTR)
library(xts)
library(TSstudio)
#Reading the Dataset

USGas<- forecast::gas

#------------ Performing EDA -------------#

class(USGas)

str(USGas)

summary(USGas)

# Checking if the dataset have null values.

sum(is.na(USGas))

# Checking Frequency.
```

```
frequency(USGas)

#Checking the periodicity of the USGas.

periodicity(USGas)

#Plotting the on Original Dataset.
gas_data <- window(USGas, start = c(1970,1), frquency =12)

dygraph(gas_data, main = "Gas Dataset")

#Plotting the seasonal Plot on Original Dataset.

ts_seasonal(gas_data, type = "all")

# Plotting the Heat Map on Original Dataset.

ts_heatmap(gas_data)

# Plotting the Surface Map on Original Dataset.
ts_surface(gas_data)

# Plotting the box Plot to check the Outliers.

boxplot(gas_data~cycle(gas_data), main = "Boxplot for Gas Dataset")


#Plotting the decompositionS of data.
gas.ts <- stl(gas_data, s.window = "periodic")
plot(gas.ts)

ts_lags(gas_data, lags = c(10,20,30,40,50,60,70,80,90,100))

#Checking the Stationarity of the data.
```

```
adf.test(gas_data,alternative = "stationary", k=12)

#As we see that the data have trend, we difference the data.

diff.gas.data<- diff(log(gas_data))

dygraph(diff.gas.data)


#Performing the ADF Test on differenced data

adf.test(diff.gas.data,k=12)

#Plotting the decomposition of differenced data.

decom.ts<- (stl(diff.gas.data,s.window = "periodic"))

plot(decom.ts)

#Splitting the data into train and test.

train <- window(diff.gas.data, end =c(1993,12), frequency =12)
dygraph(train)


test <- window(diff.gas.data,start = c(1994,1), frequency=12)
dygraph(test)


# Performing ACF test to identify q.
acf(train, lag.max = 50)

# Performing PACF to identify p.
pacf(train, lag.max = 50)

# Performing Manual arima on calculated p and q.
```

```
gasArima <- arima(exp(train), order = c(2,0,3))
summary(gasArima)

# Perforing Ljung Test to check if the residuals are independent.

Box.test(exp(gasArima$residuals))

# Plotting Residuals
tsdisplay(residuals(gasArima), lag.max = 45, main = "Arima Residuals")



# Performing accuracy on Test data.
forecast_arima <- forecast(gasArima, h=20)

accuracy(forecast_arima, exp(test))

test_forecast(forecast.obj = forecast_arima,actual =exp(diff.gas.data),test = exp(test))



#Performing Auto Arima

gasAuto <- auto.arima(exp(train),seasonal = TRUE)
summary(gasAuto)



#Perfroming Ljung test

Box.test(gasAuto$residuals)

#Predicting the accuracy on test.

forecast_auto <- forecast(gasAuto, h=20)
accuracy(forecast_auto, exp(test))

plot(forecast_auto)
```

```
test_forecast(forecast.obj = forecast_auto,actual = exp(diff.gas.data),test = exp(test))

# Building the Model on the Original Dataset.

auto_original <- auto.arima(gas_data, seasonal = TRUE)
summary(auto_original)

#Perfroming Ljung test

Box.test(auto_original$residuals)

#Forecasting for next 12 Months

next_forecast <- forecast(auto_original, h=12)
next_forecast


plot(next_forecast)
```