Time Series Project

K Vineet PATNAIK

Table of contents

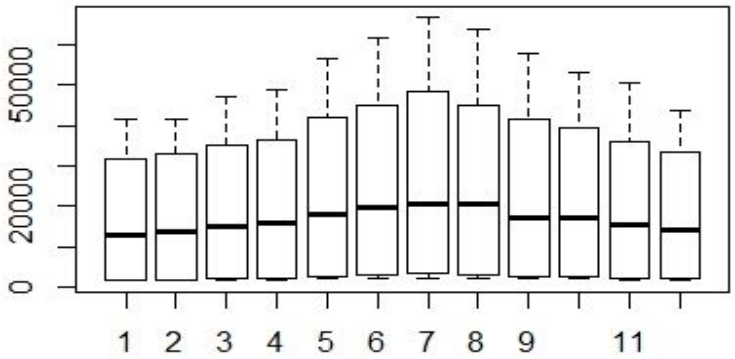
EDA

There are different libraries used in the following project which are mentioned in the code

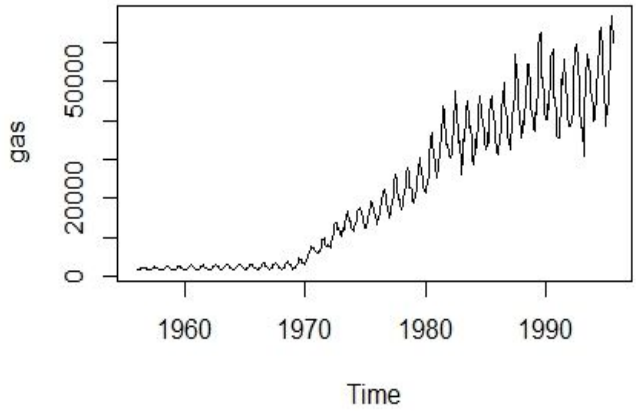
CODE

```
install.package(forecast)  # we will use the GAS dataset from this package

library(forecast)
library(quantmod)
library(tseries)
library(xts)

str(gas)   #Time-Series [1:476] from 1956 to 1996
summary(gas)
 #  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1646    2675   16788   21415   38629   66600
```

```
boxplot(gas~cycle(gas))
sum(is.na(gas)) #0
cycle(gas)
```




```
plot.ts(gas)
```




```
aus_gas <-ts(gas, frequency=12, start=c(1956,1))
```

trend tends to be constant for around 10 yrs till 70's and then starts increasing and again becomes constant over time from 1990 to remaining years but an increasing curve

plot(TScomp)

the random graph shows a lot of irregularity in the data which has increased over the recent years i.e various factors may have have affected the time series curve and it shows about the trend

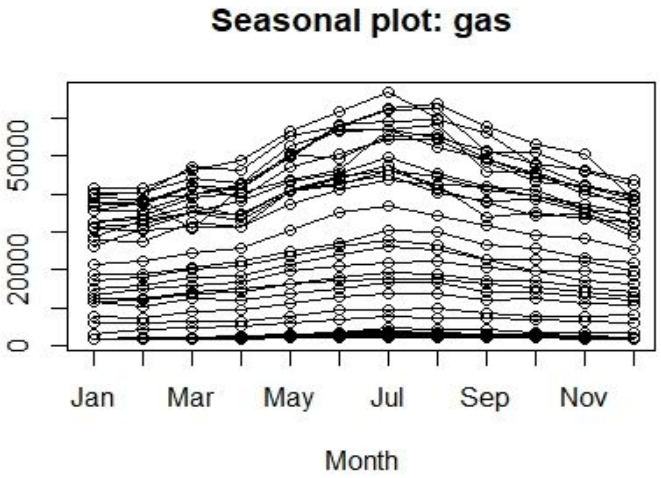
There is a multiplicative seasonality as the graph is getting wider and will increase or decrease along the trend

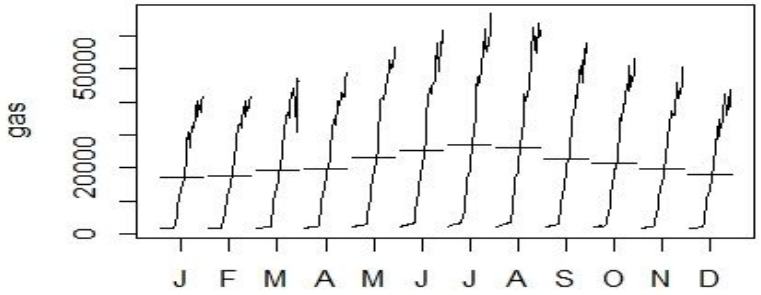
monthplot(gas)

the seasonilty peak is clearly at the centre of the year(june, july,aug) while year ends (jan and dec) are least ,inc till mid of the year and decreasing again to the end based on boxplot

boxplot(gas~cycle(gas))

seasonplot(gas)





```
periodicity(gas) # Monthly periodicity from Jan 1956 to Aug 1995
```

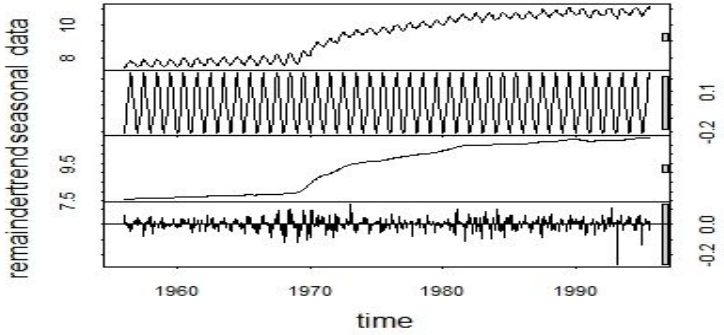
To Find out Stationarity of the dataset

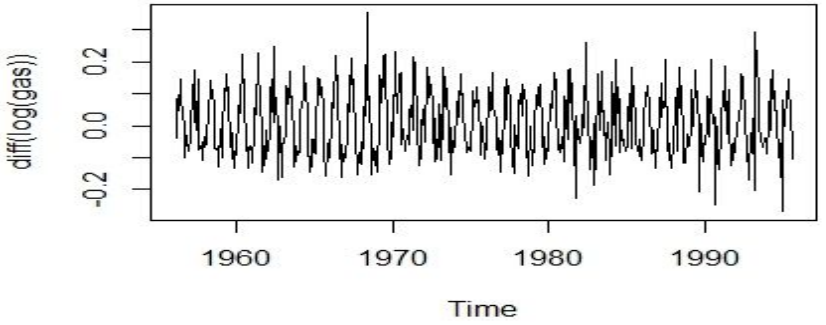
If we look at the actual plot or graph of the GAS time series dataset we can clearly understand the mean is varying across the time period and also has a variance factor involved, so we can say that there is no stationarity in the dataset.To make it a stationary we have to convert to make the mean data constant across the whole graph or decompose the time series which reduces the remainder part or external factor

```
# Decomposing the time series
TScomp = decompose(aus_gas)
decomp = stl(log(gas), s.window="periodic")
```

plot(decomp)
decomp$time.series

	seasonal	trend	remainder
Jan 1956	-4083.7994	3000.295	2792.50473
Feb 1956	-3715.9678	2792.712	2569.25583
Mar 1956	-2165.1863	2585.129	1374.05703
Apr 1956	-1809.3076	2461.267	1226.04094
May 1956	1713.8461	2337.404	-1878.25008
Jun 1956	3591.8664	2261.258	-3532.12392
Jul 1956	5312.5120	2185.111	-5029.62308
Aug 1956	4256.0658	2100.722	-3940.78794
Sep 1956	1528.2903	2016.333	-1360.62357
Oct 1956	163.9636	2115.969	-158.93237
Nov 1956	-1471.7225	2215.604	1218.11825
Dec 1956	-3320.5600	2333.018	2812.54224
Jan 1957	-4083.7994	2450.431	3384.36823
Feb 1957	-3715.9678	2379.171	3024.79674
Mar 1957	-2165.1863	2307.911	1777.27535
Apr 1957	-1809.3076	2127.690	1622.61780
May 1957	1713.8461	1947.469	-1350.31467
Jun 1957	3591.8664	1848.064	-3160.93037
Jul 1957	5312.5120	1748.659	-4423.17138
Aug 1957	4256.0658	1813.580	-3621.64562
Sep 1957	1528.2903	1878.500	-1127.79062
Oct 1957	163.9636	2062.935	-63.89864
Nov 1957	-1471.7225	2247.370	1165.35277
Dec 1957	-3320.5600	2363.266	2835.29385

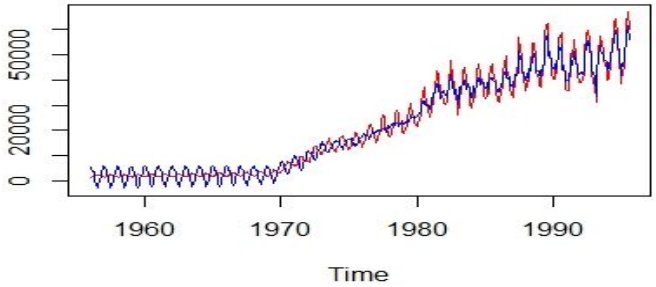




plot(diff(log(gas))) ## making a constant(mean) graph

```
library(tseries)
adf.test(gas,alternative = c("stationary"))
#Dickey-Fuller = -2.7131, Lag order = 7, p-value = 0.2764
#alternative hypothesis: stationary
```
Since we cannot reject the null hypothesis,so this is non stationary
```
kpss.test(gas)                  # KPSS test for Level Stationarity
                                # Null Hypo: Data is Stationary
                                # Alternate Hypo: Data is not stationary
```
KPSS Level = 7.7018, Truncation lag parameter = 5, p-value = 0.01
```
# Deseasonalisation

series_names = c('deseasoned','actual')
deseason_gas = seasadj(decomp)
deseason_gas
plot(deseason_gas)
```

```
ts.plot(gas,deseason_gas, col=c("red", "blue"))

kpss.test(deseason_gas)
data:  deseason_gas
```



KPSS Level = 7.7751, Truncation lag parameter = 5, p-value = 0.01

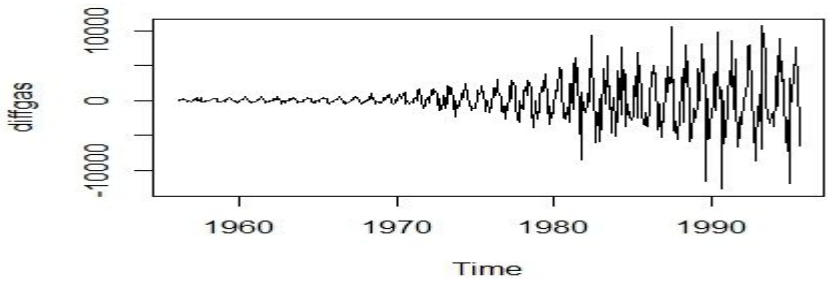
```
# KPSS test for Level Stationarity
# Null Hypo: Data is Stationary
# Alternate Hypo: Data is not stationary
```

This shows that even the deseasonalised data isn't stationary
Since the series are not stationary, let's do the differencing to get the stationary series.

ndiffs(gas) #number of differences required for time series x to be made stationary

diffgas = diff(gas,differences = 1)

plot(diffgas)



The series has it's trend removed

Now lets check with the adf test

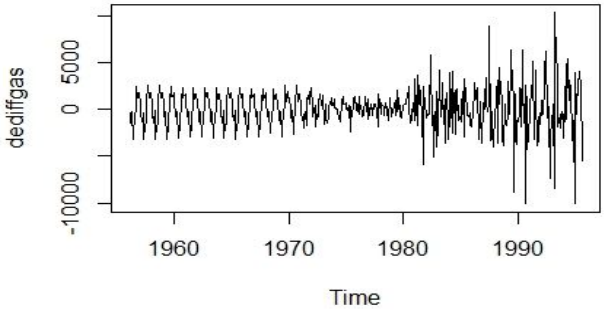
Dickey-Fuller = -19.321, Lag order = 7, p-value = 0.01

alternative hypothesis: stationary

The series is defineitly improved and also stationary, now lets make it deseasonalised

dediffgas = diff(deseason_gas,differences = 1)

plot(dediffgas) # deseasonalised



acf and pacf together show order of autoregression

Acf(gas)

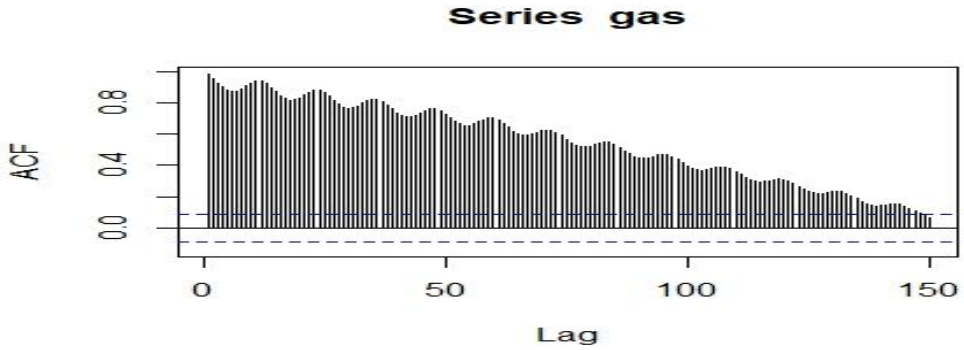
Acf(gas,lag = 150) ## all the values above blue line are significant around which is very slow decrease autocorrelation decreasing as lag increases

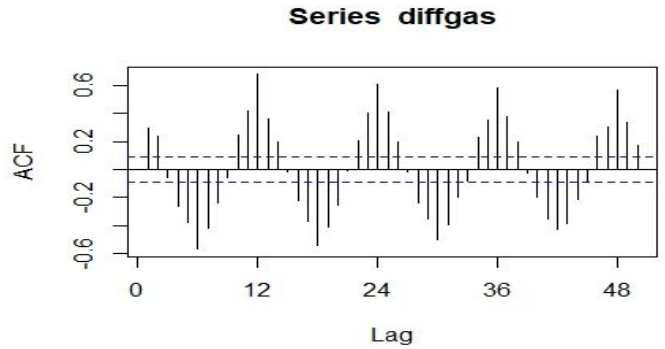
Acf(diffgas,lag=15)

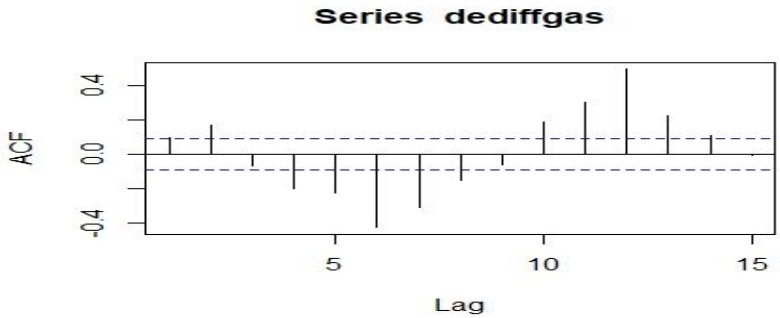
Acf(dediffgas,lag = 15)

Acf(diffgas,lag=15)

Acf(dediffgas,lag = 15)







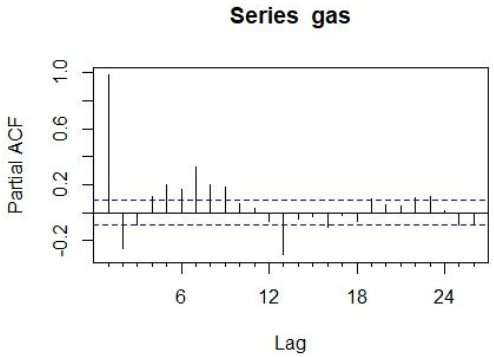
PACF it tells how many past values we have to include

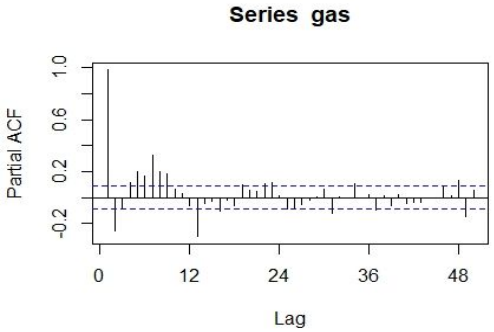
Pacf(gas)

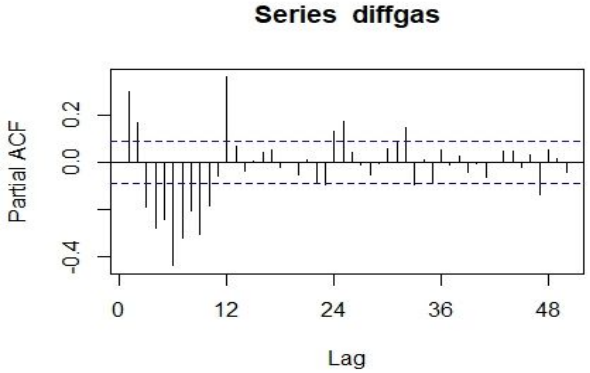
Pacf(gas,lag=15) # the lines above the blue line is said to be significant

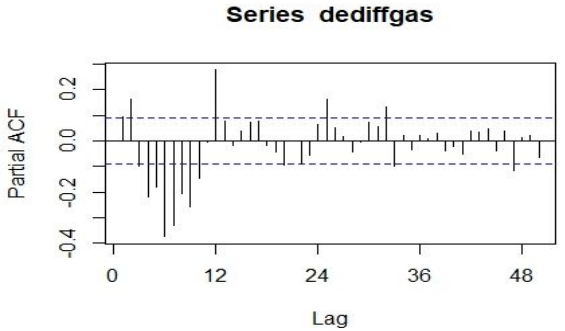
Pacf(diffgas,lag=15)

Pacf(dediffgas,lag=15)







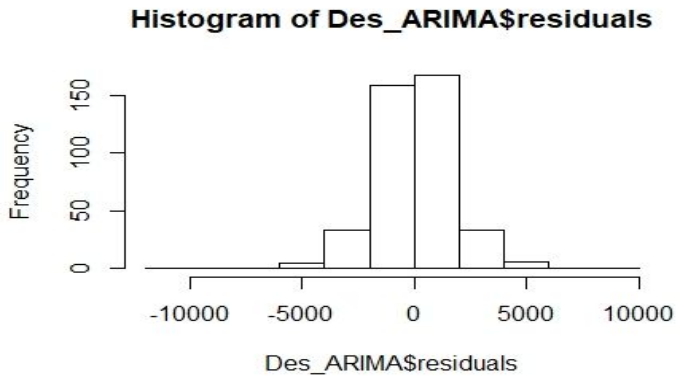


The values of (p,d,q) are (2,0,2)

To Determine the values of p and q

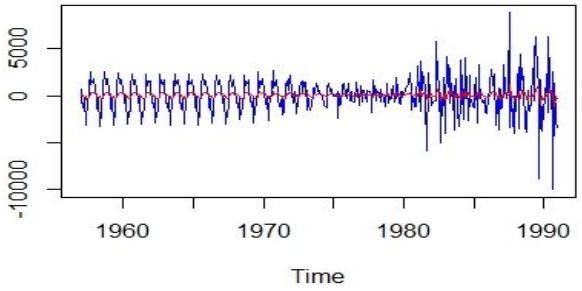
We need to train and test the data firstly

```
train1 = window(dediffgas,start = c(1957,1),end= c(1990,12), frequency=12)
test1 =  window(dediffgas,start = c(1991,1),end = c(1995,8),frequency = 12)
Des_ARIMA <- arima(train1, order=c(1,0,1))
Des_ARIMA
hist(Des_ARIMA$residuals)
```

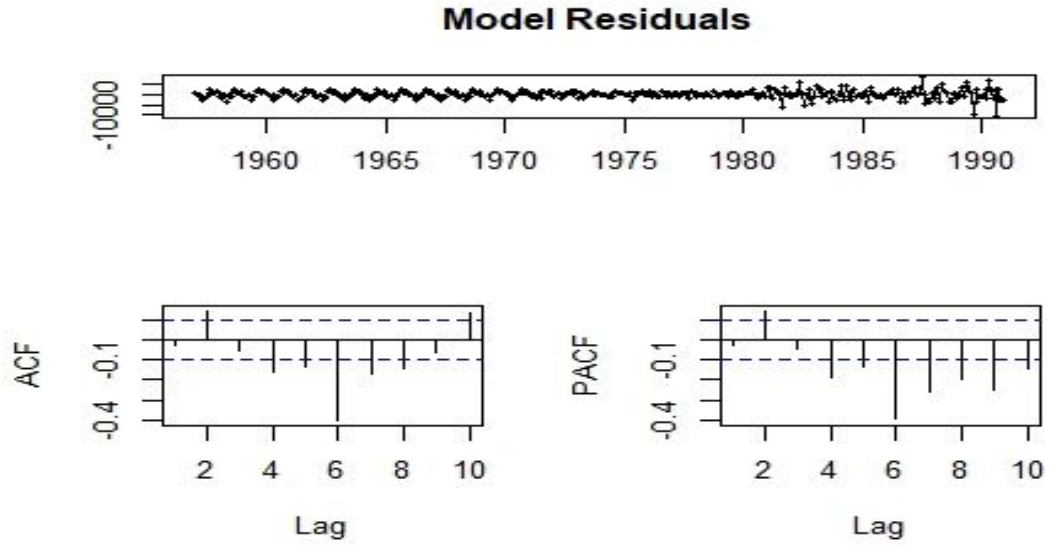



```
Coefficients:
         ar1      ma1     intercept
      0.3962  -0.2724     82.3883
s.e.  0.1370   0.1365    111.2596
  sigma^2 estimated as 3482716:
 log likelihood = -3651.85,  aic = 7311.71
```

```
PA_ARIMA<- cbind(train1,fitted(Des_ARIMA))
ts.plot(PA_ARIMA,col=c("blue", "red"))
tsdisplay(residuals(Des_ARIMA), lag.max=10
, main='Model Residuals')
```



The values are given for deseasonalised (p,d,q)=(0,1,0)



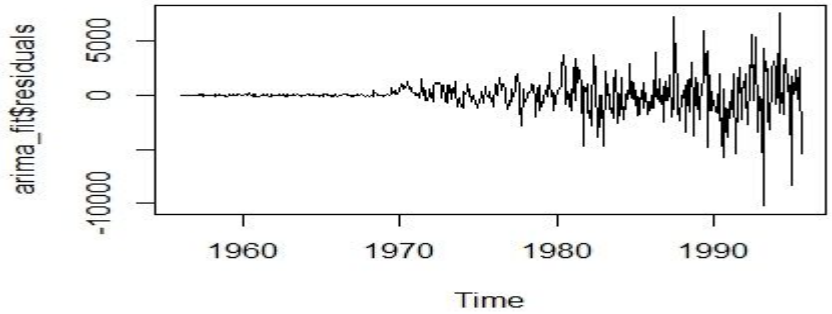
ARIMA Forecasting model and Accuracy(AUTO)

```
arima_fit = auto.arima(gas,seasonal = TRUE)
Arima_fit
 ARIMA(0,0,2) with zero mean is given for deseasonalised
Coefficients:
         ma1     ma2
      0.2576  0.3087
s.e.  0.0450  0.0424
sigma^2 estimated as 6834834:  log likelihood=-4410.78
AIC=8827.56   AICc=8827.61   BIC=8840.05
plot(arima_fit$residuals)
plot(arima_fit$x,col = "purple")
lines(arima_fit$fitted,col="red",main="Actual vs Forecast ")
tsdisplay(residuals(arima_fit),lag.max=45, main='Auto ARIMA Model Residual
Box.test(arima_fit$residuals, type="Ljung-Box")
```

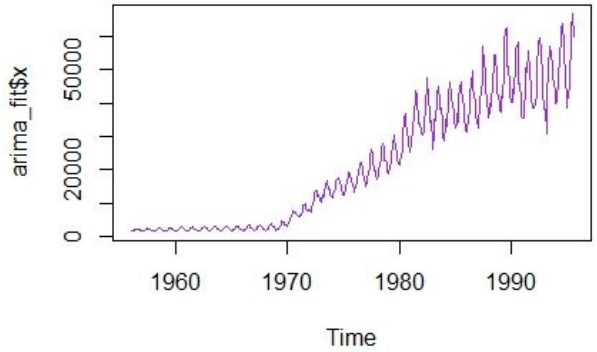
Now to check with the seasonality as TRUE

```
arima_fit = auto.arima(gas,seasonal = TRUE)
arima_fit
plot(arima_fit$residuals)
```

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

Coefficients:
         ar1     ar2      ma1     sma1
      0.3756  0.1457  -0.8620  -0.6216
s.e.  0.0780  0.0621   0.0571   0.0376

sigma^2 estimated as 2587081:  log likelihood=-4076.58
AIC=8163.16   AICc=8163.29   BIC=8183.85
```

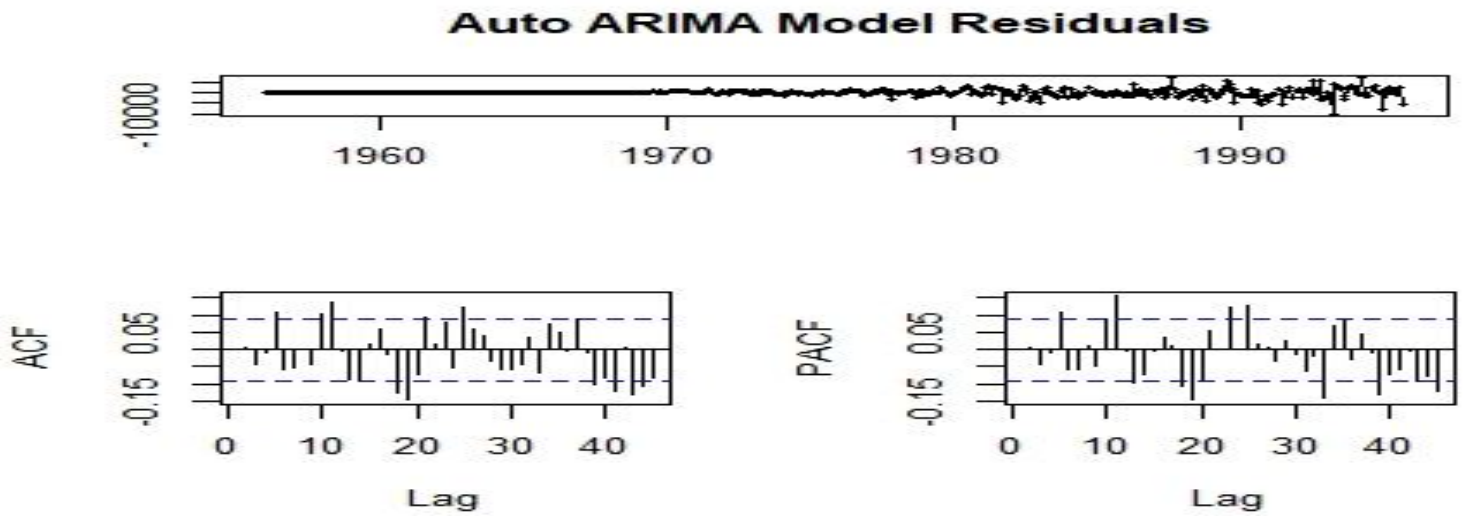



```
plot(arima_fit$x,col = "purple")
lines(arima_fit$fitted,col="red",
main="Actual vs Forecast ")
```



The values taken in the seasonalised data will be used for forecasting .the values are (2,1,1),(0,1,1)

```
tsdisplay(residuals(arima_fit),lag.max=45, main='Auto ARIMA Model Residuals')
Box.test(arima_fit$residuals, type="Ljung-Box") #p-value:0.9604
```



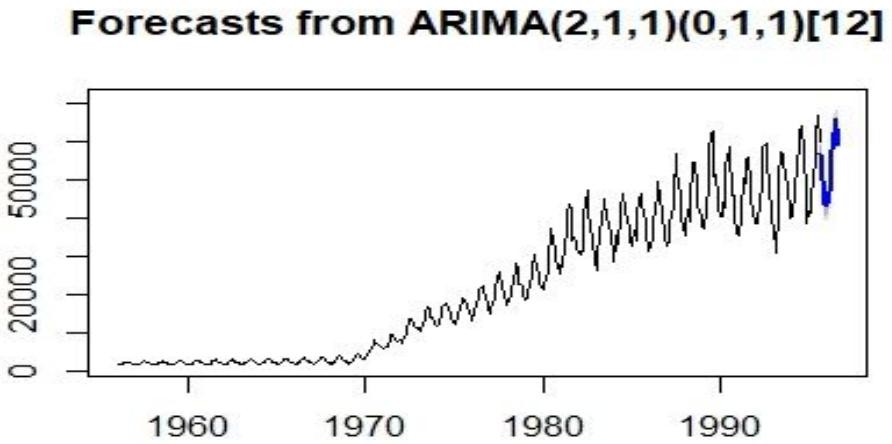
autocorrelations can be rejected or no autocorrelation

```
MAPE_ARIMA <- mean(abs(arima_fit$fitted-arima_fit$x)/arima_fit$x)
MAPE_ARIMA
MAPE(arima_fit$fitted,arima_fit$x)[1] 0.03900233
```

This shows that the time series has very less error rate of just around3.9%

ACCURACY

```
arima_fct = forecast(arima_fit,
h=13)
plot(arima_fct)
arima_check = cbind(test
,arima_fct$mean)
ts.plot(arima_check,
col = c("blue","red"))
```



The plot above shows the forecasted monthly gas production with the blue line with 80 percent and 95 percent confidence intervals in dark and light blue

```
> accuracy(arima_fct)
                    ME     RMSE      MAE       MPE     MAPE      MASE         ACF1
Training set 27.72266 1579.457 893.4504 0.272275 3.900233 0.4789312 0.002271673
>
```

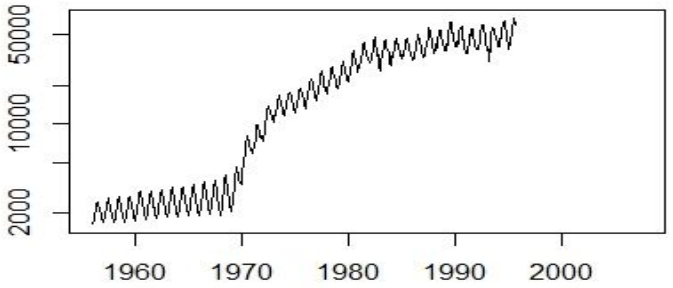
Manual ARIMA model

```
fit = arima(gas,c(1,1,0))
fit
```

```
      ar1
   0.2994
s.e.  0.0440
```

```
pred = predict(fit, n.ahead = 12*12)
pred1 = 2.718^pred$pred
pred1   # this will predict the units of gas production from sept 1995 to aug
          2007
ts.plot(gas,pred1,log = "y",lty = c(1,3))
```

ts.plot(gas,pred1,log = "y",lty = c(1,3))




```
> gasD <- log(gas)
> Model1 <-arima(gasD, order = c(0,1,0), season=list(order = c(2,0,2), period=12))
> Model1

Call:
arima(x = gasD, order = c(0, 1, 0), seasonal = list(order = c(2, 0, 2), period = 12))

Coefficients:
         sar1     sar2     sma1    sma2
      1.1739  -0.1746  -0.9518  0.0076
s.e.  0.2909   0.2903   0.2984  0.2665

sigma^2 estimated as 0.00269:  log likelihood = 714.05,  aic = -1418.1
> plot(forecast(Model1, h = 12), xlab="Time", ylab="Log Monthly Gas Production",
+      main = "Australian Monthly Gas Production")
> accuracy(Model1)
                      ME       RMSE        MAE         MPE      MAPE      MASE        ACF1
Training set 0.0007690667 0.05180848 0.03788103 0.01154637 0.4064119 0.4558894 -0.2742453
```

The error rate is less so the method used and the values taken are used in prediction for the next 12 periods is pretty good and here is the graph predicted across 80 to 95 percent confidence levels which is shown by the blue line