Time Series Analysis Lab 1

Omkar Bhutra (omkbh878), Nadezhda Green (nadgr258)

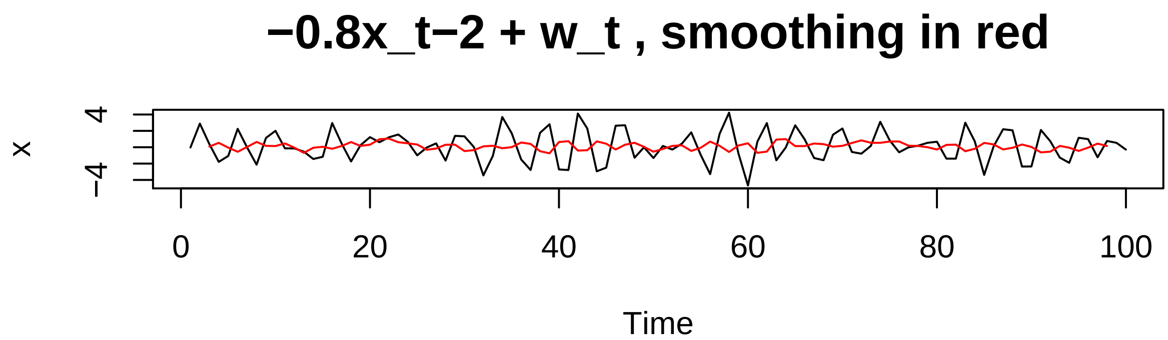
18 September 2019

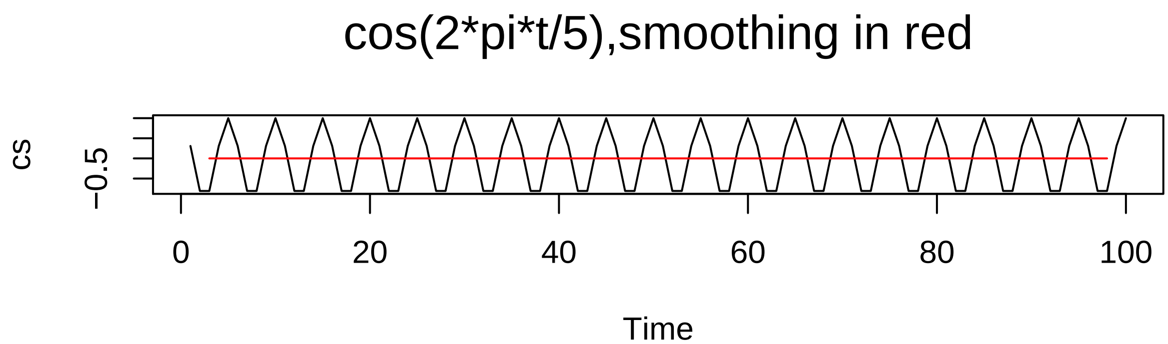
Assignment 1

a)

For the first time series, we can say that the filter has made the time series more smooth. Further, the series has less variation on the horizontal axis. We can see in the function of the filtering that it takes the avarage of neigbour observations in the past. Regarding the cosine time series, we can see that smoothing filter makes the time series completely horizontal. The reason for this is that the cosine series have the same pattern as a function of time and if we do not use an even number of lags in the smoothing filter we will always get a straight line.

```
set.seed(12345)
par(mfrow=c(2,1), cex.main=1.5)
#First Time Series
w = rnorm(110,0,1) #10 extra to avoid startup problems
x = stats::filter(w, filter = c(0,-0.8), method="recursive")[-(1:10)] # remove first 10
plot.ts(x, main = "-0.8x_t-2 + w_t , smoothing in red")
v1 = stats::filter(x,sides = 2, filter = rep(0.2,5))
lines(v1, type = "l", col = "red")
#Second Time Series
cs = cos(2*pi*1:100/5); w = rnorm(100,0,1)
v2 = stats::filter(cs,sides = 2, filter = rep(0.2,5))
plot.ts(cs, main = expression("cos(2*pi*t/5),smoothing in red"))
lines(v2, type = "l", col = "red")
```





The first time series is quite irregular with large fluctuations in values and looks choppy, the smoothing filter takes the average of 5 the previous points of the past and makes its smoother and also closer to the centerline. The seconds time series is a cosine function, the cosine wave has regular patterns and the smoothing filter completely reduces the series to 0 which is seen as the horizontal line. If odd number of lags are used in the smoothing filter this will be the result.

b)

Given the following equation with the autoregressive function on the left hand side and the moving avarage on the right hand side. In order to check if the series is causal and invertible we have to check the polynomial root of the series. This is is to see that if any of the resulting parameters are inside the unit circle then they are non causal and non invertible.

$$x_t - 4x_{t-1} + 2x_{t-2} + x_{t-5} = w_t + 3wt - 2 + w_{t-4} - 4w_{t-6}$$

.
Simplified:

$$(1 - 4B + 2B^2 + B^5)x_t = (1 + 3B^2 + B^4 - 4B^6)w_t$$

.

We have to check if all of the parameters satisfy the constraint:

$$\sqrt{(}R^2 + I^2)$$

where R is the real number and I is the imaginary number.

The process is not invertible and not causal.

```r
#Check Invertiblity
Invertible <- polyroot(c(1,0,3,0,1,0,4))
#Check causality
Causal <- polyroot(c(1,-4,2,0,0,1))
rootFun <- function(y){
  res <- c()
  for(i in 1:length(y)){
    res[i] <- sqrt(Im(y[i])^2 + Re(y[i])^2)
  }
  res <- as.numeric(res)
  if(any(res<1)){
    return("Non Causal/Non Invertible")
  } else {
    return("Causal/Invertible")
  }
}
root1 <- rootFun(Invertible)
root2 <- rootFun(Causal)
root1
```

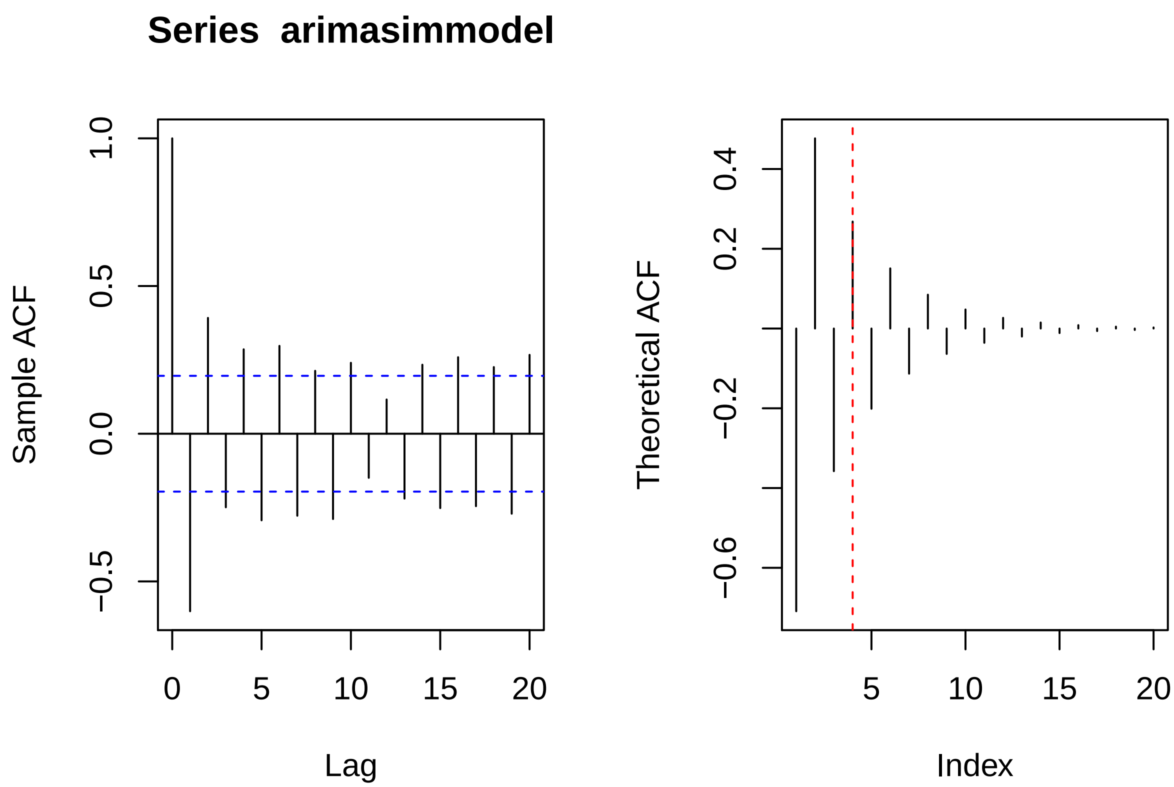
```
## [1] "Non Causal/Non Invertible"
```

```r
root2
```

```
## [1] "Non Causal/Non Invertible"
```

c)

```r
set.seed(54321)
#model
arimasimmodel<-arima.sim(model = list(ar = -3/4, ma = c(0,-1/9)), n = 100 )
#plotting
par(mfrow = c(1,2))
acftest<-acf(arimasimmodel,ylab="Sample ACF")
arimaacfobj<-ARMAacf(ar = c(-3/4), ma = c(0,-1/9), lag.max=20)[-1]
plot(arimaacfobj , type = "n",ylab="Theoretical ACF")
lines(arimaacfobj , type = "h")
abline(col = "red", v = 4, lty = 2)
```



Difference between the theoretical ACF and the sample ACF is that the autocorrelations in the theoretical ACF plot are given by a recursive filter when our $lag > 3$.

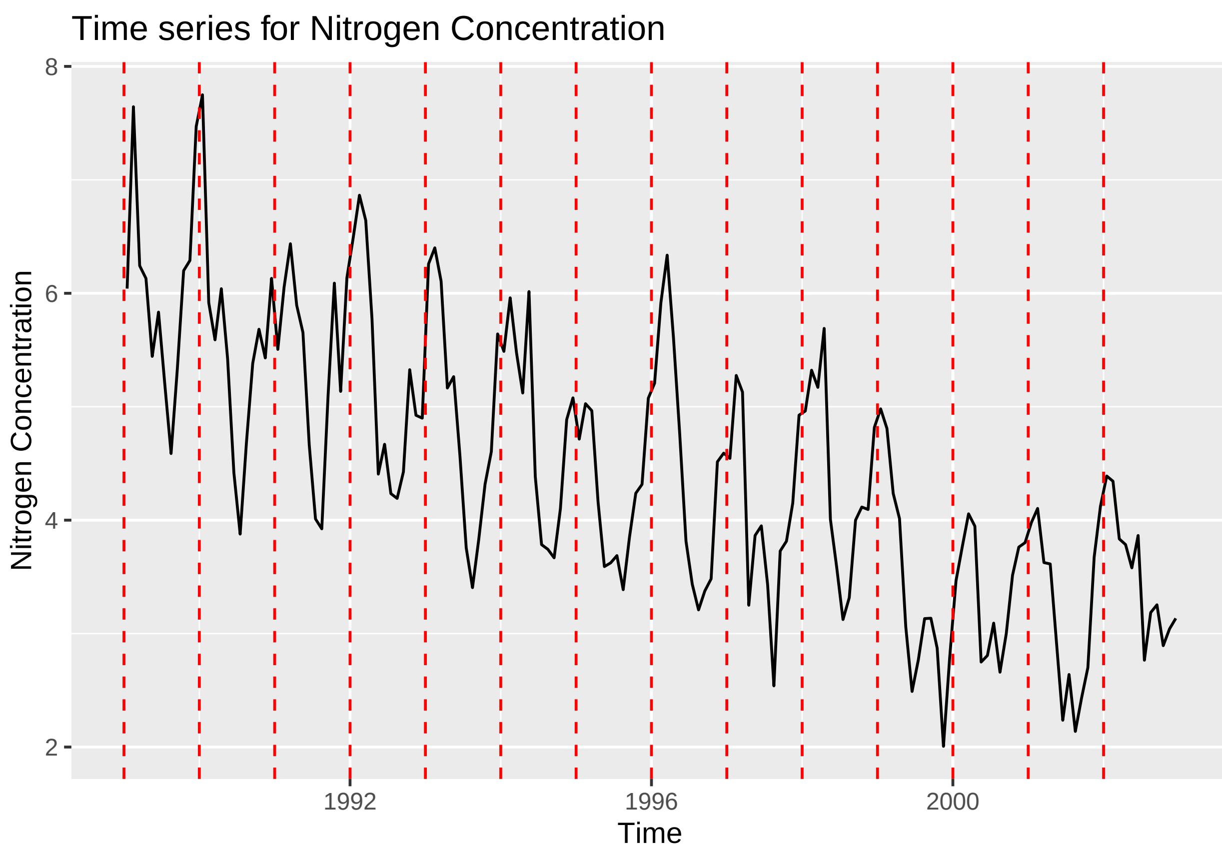
Assignment 2

a)

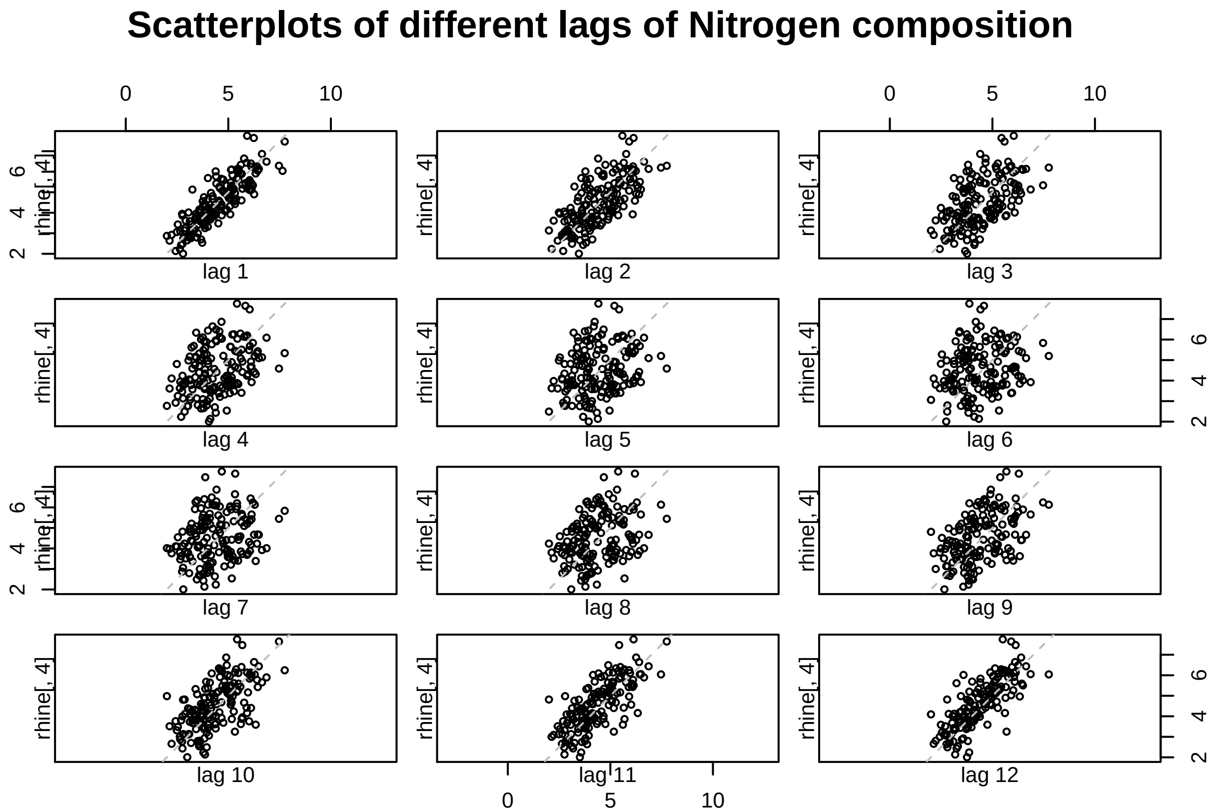
From the plot of the time series, it is observed that there is a general decreasing trend over the years observed and also a regular seasonality where the N conc. rises sharply towards the end of the year peaking at the beginning of the next year and the falls sharply. The variance does not seem to change over time. This is confirmed from the scatterplots where it is seen that there is a linear trend in the first and last months of the year.

```
rhine <- ts(read.csv2("C:/Users/Omkar/Documents/Rhine.csv"))
rhinedf <- as.data.frame( read.csv2("C:/Users/Omkar/Documents/Rhine.csv") )

ggplot(rhinedf, aes(x = rhinedf$Time))+geom_line(aes(y = rhinedf$TotN_conc))+geom_vline(aes(xintercept =
```




```r
lag.plot(x = rhine[,4], lag = 12,main = "Scatterplots of different lags of Nitrogen composition" )
```

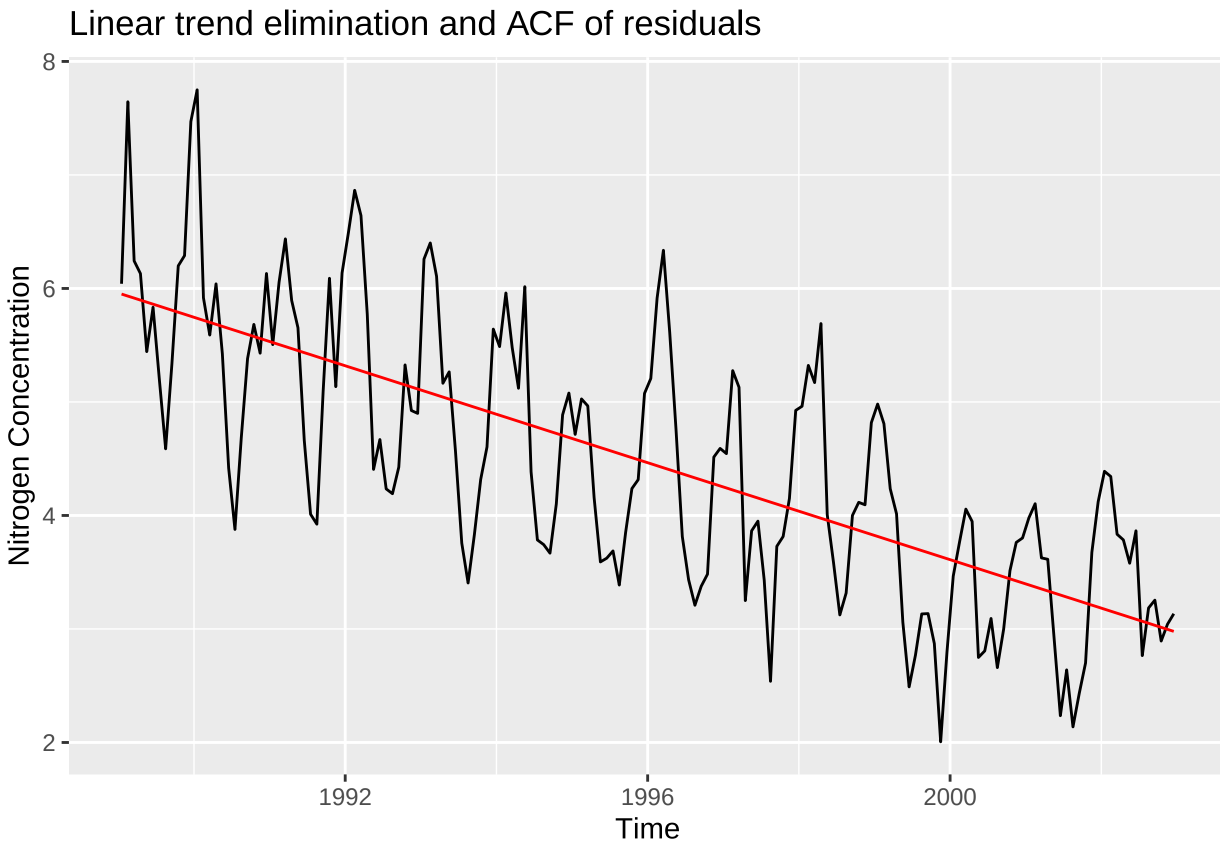


b)

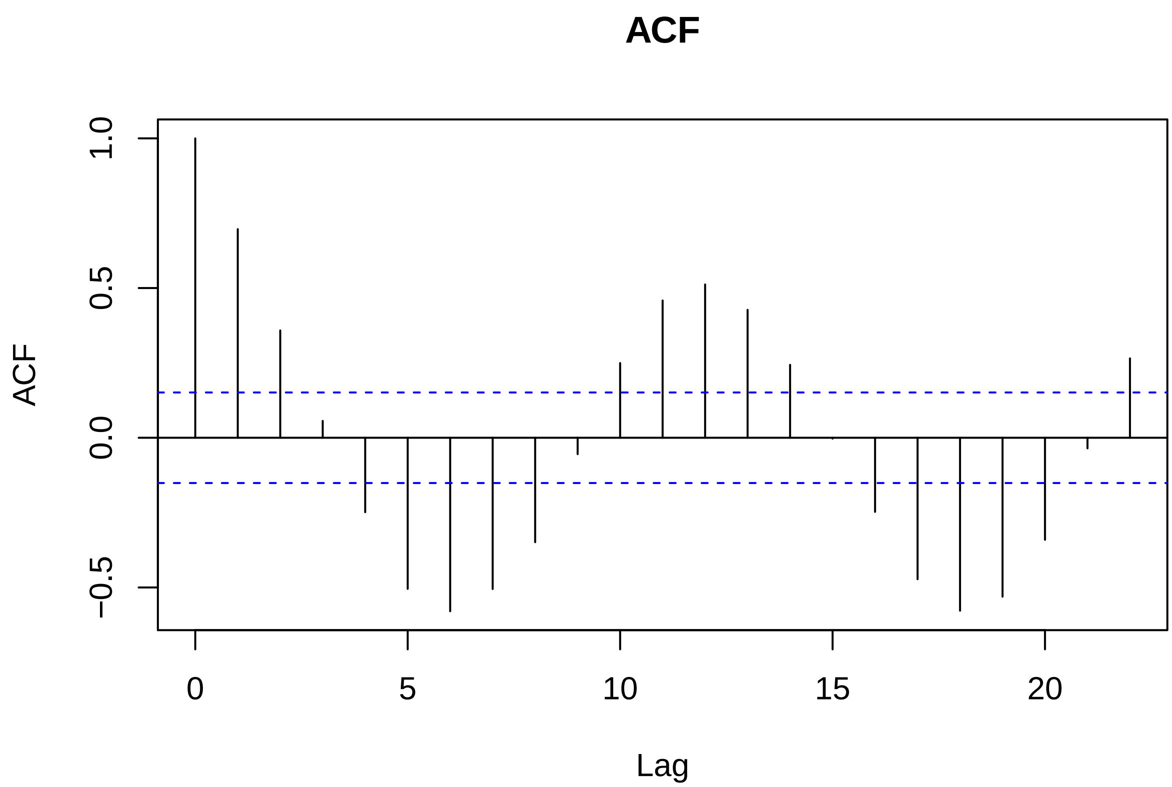
The fitted values in figure show that there is a decreasing trend in Nitrogen Concentration over time. For each month the slope has a significant trend of approximately -0.2 Nitrogen concentration units. On observation of the ACF we can see that the autocorrelation has a recurring pattern every year which is sign of seasonality. By looking at figure we can see the seasonality in the difference between the horizontal line at 0 and the data.

```
#Eliminating the trend by fitting a linear model
linear.model <- lm(TotN_conc ~ Time, data = rhinedf)

ggplot(rhinedf, aes(x = rhinedf$Time))+geom_line(aes(y = rhinedf$TotN_conc))+geom_line(aes(y =linear.mod
```

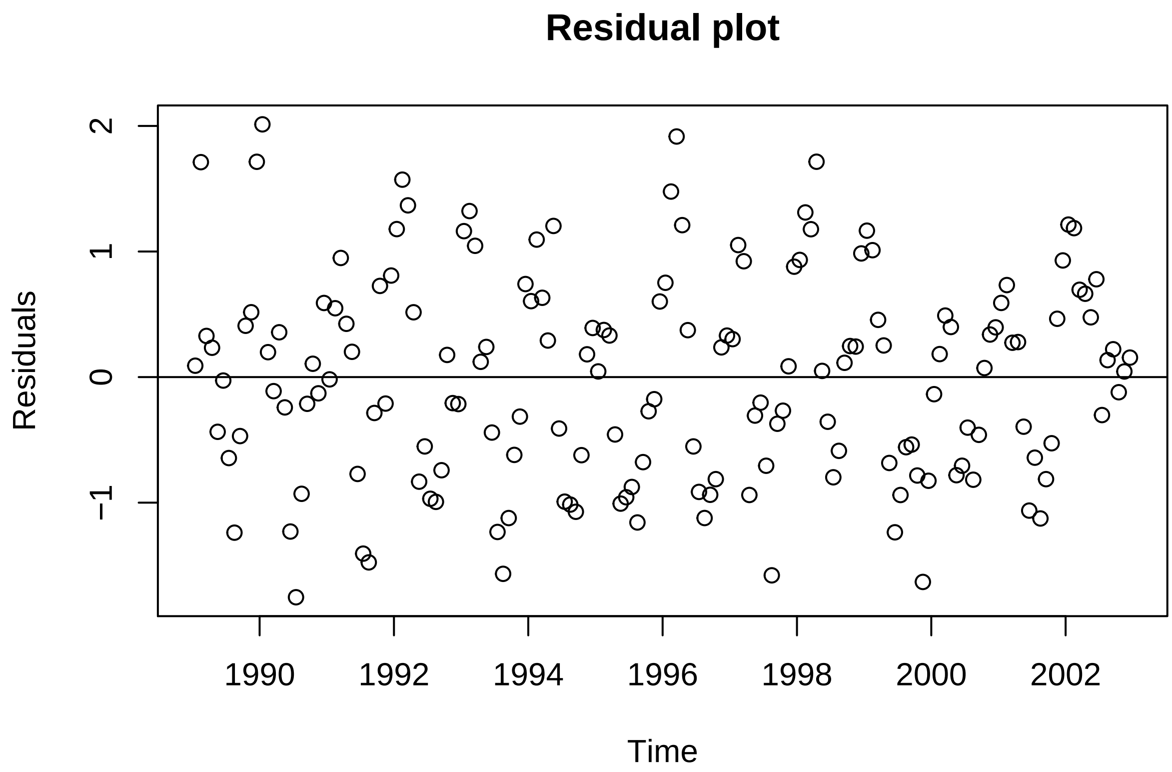



```
#Plot ACF
acf(linear.model$residuals, main = "ACF")
```




```r
#Plot Residuals
plot(linear.model$residuals, main = "Residual plot",
     x = linear.model$model$Time, ylab = "Residuals",
     xlab = "Time", type = "p")

abline(h = 0)
```



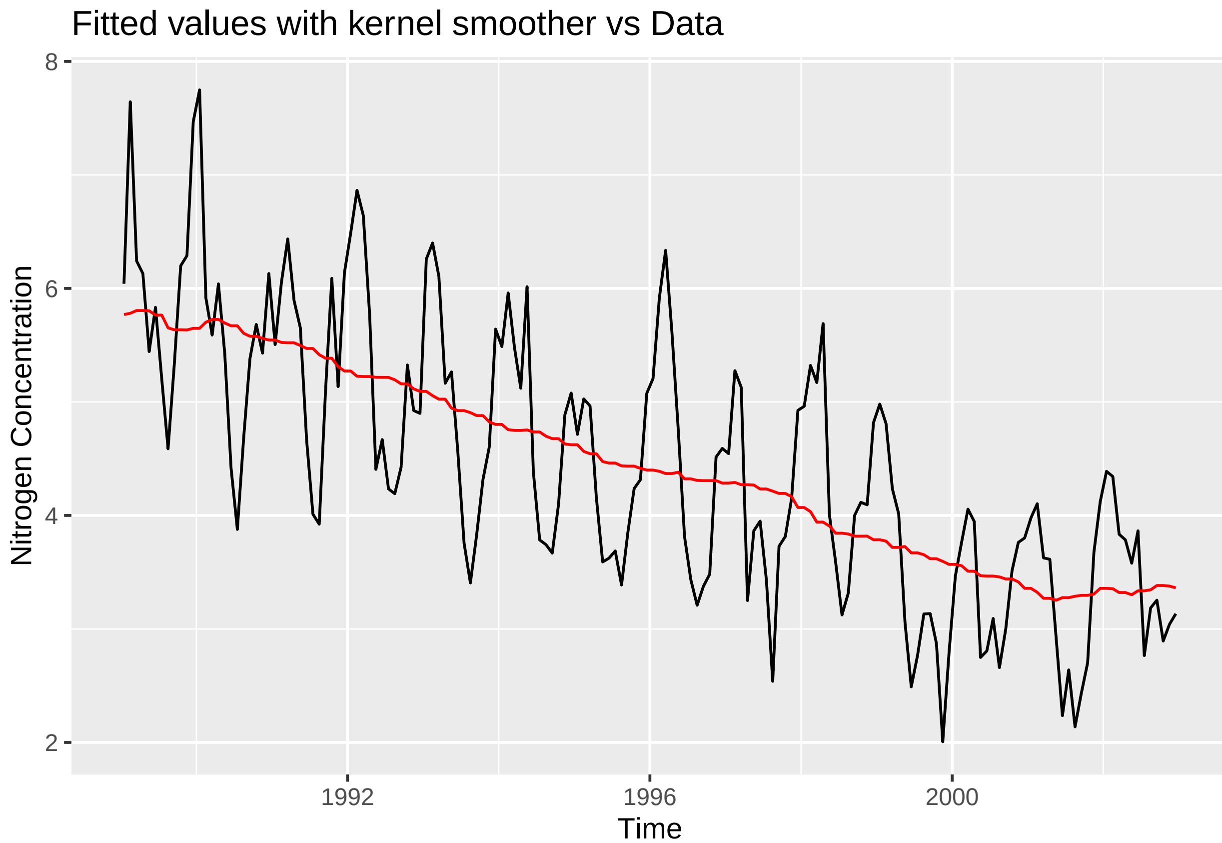

```
summaryModel <- summary(linear.model)
```

c)

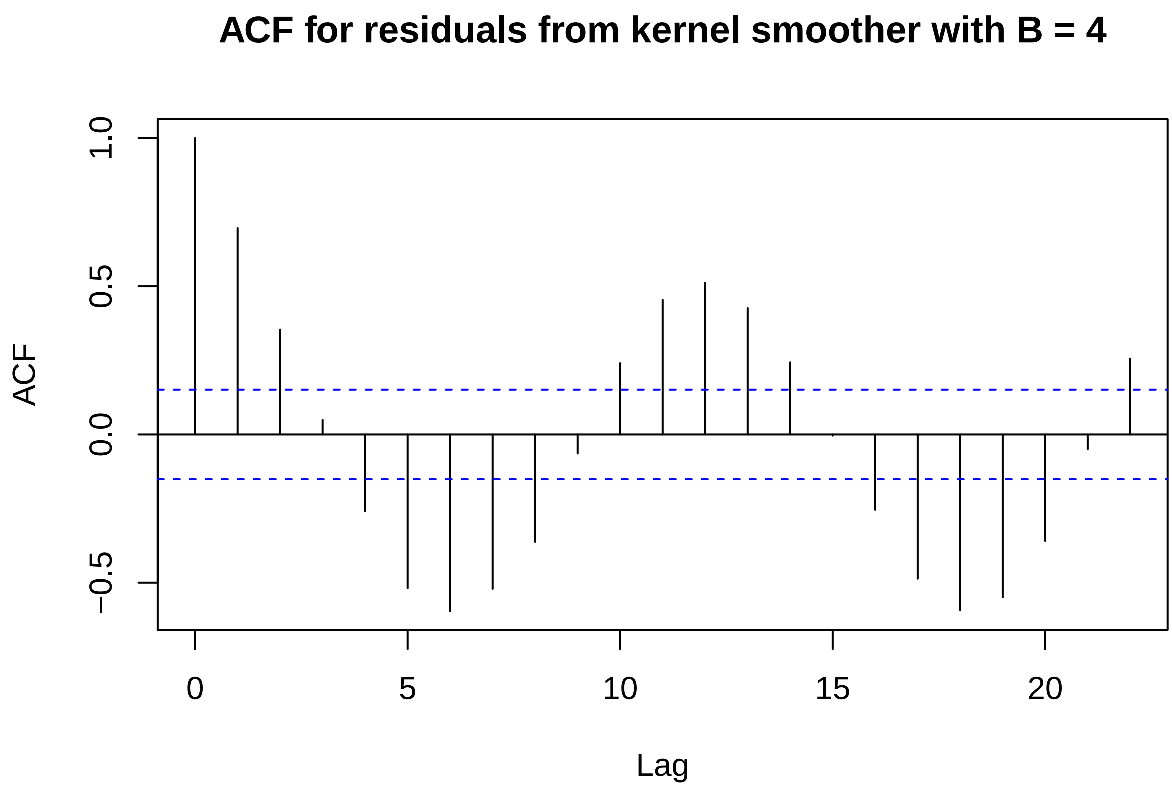
In Figure 1 we have compared the residual pattern from the linear model, kernel smoother with bandwith 4 and kernel smoother with bandwith 20. As we increase the bandwith the fitted line becomes underfitted and the residual pattern changes. The residuals does not seem to be stationary in the models since they seem to follow a pattern that is dependent on time which we can see in Figure 1, this can be interpreted as a seasonal effect.

```
#Eliminating the trend by fitting a kernel smoother
kernel.model <- ksmooth(x = rhine[, 3], y = rhine[, 4], bandwidth=4)
kernel.model20 <- ksmooth(x = rhine[, 3], y = rhine[, 4], bandwidth=20)
residuals.kernel <-  rhine[, 4] -  kernel.model$y
residuals.kernel20 <- rhine[, 4] -  kernel.model20$y

ggplot(rhinedf, aes(x = rhinedf$Time))+geom_line(aes(y = rhinedf$TotN_conc))+geom_line(aes(y =kernel.mo
```




```
#ACF plot for residuals of kernel smoother
acf(residuals.kernel, main = "ACF for residuals from kernel smoother with B = 4")
```




```
#Extract & plot residuals from kernel
plot(y = residuals.kernel, x = rhine[,3],type = "l",
     main = "Residuals from kernel smoother",
     xlab = "Time", ylab = "Residuals")
lines(linear.model$residuals,
     x = linear.model$model$Time, xlab = "Time",
     type = "l", col = "red")
lines(residuals.kernel20,
     x = linear.model$model$Time, xlab = "Time",
     type = "l", col = "green")
abline(h = 0)
legend("topright", col = c("red","black", "green"),
       legend = c("Linear Residuals", "Kernel Smoother B = 4", "Kernel Smoother B = 20"),
       lty = c(1,1,1), cex = 0.7)
```

d)

It is observed that the residual plot does not look dependent on time anymore and this looks like noise and it is visually confirmed that its stationary. From the ACF plot it is seen that the seasonality has disapeared since we now included the month in the model and this is not affected in the residuals as before.

```
linear.model.d <- lm(rhine[, 4] ~ rhine[, 3] + as.factor(rhine[, 2]))
par(mfrow = c(1,2))
acf(linear.model.d$residuals,
```

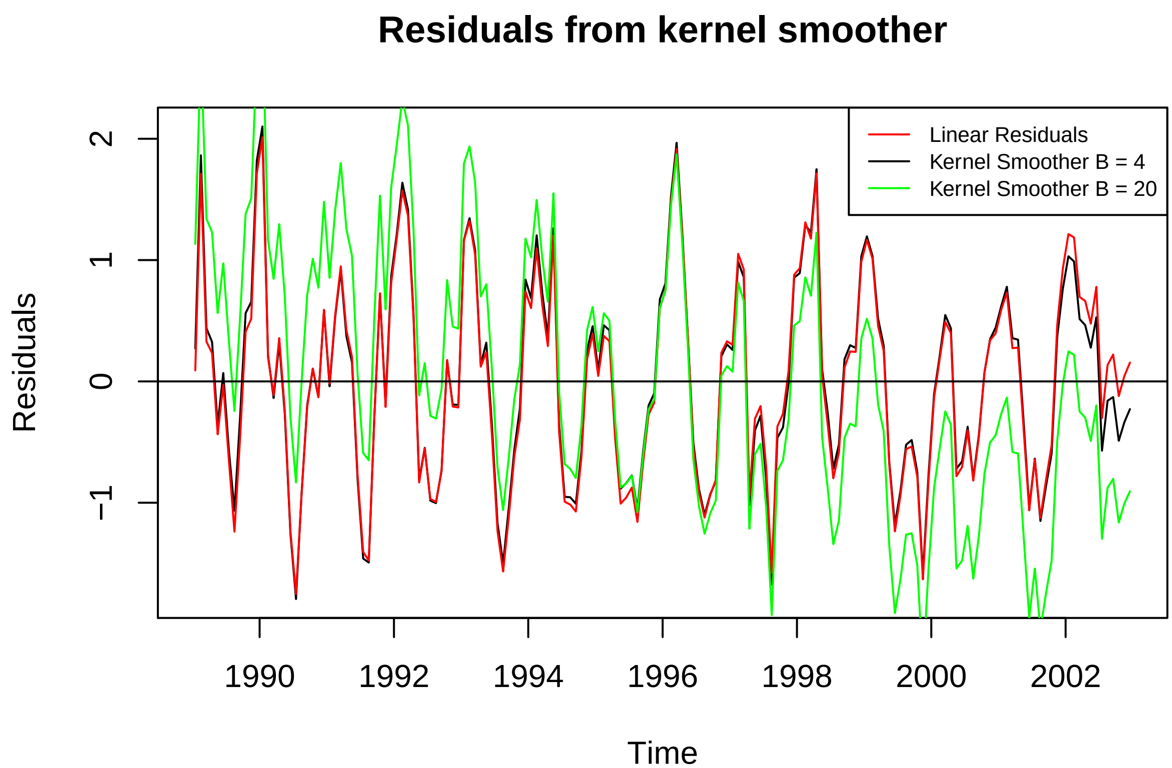
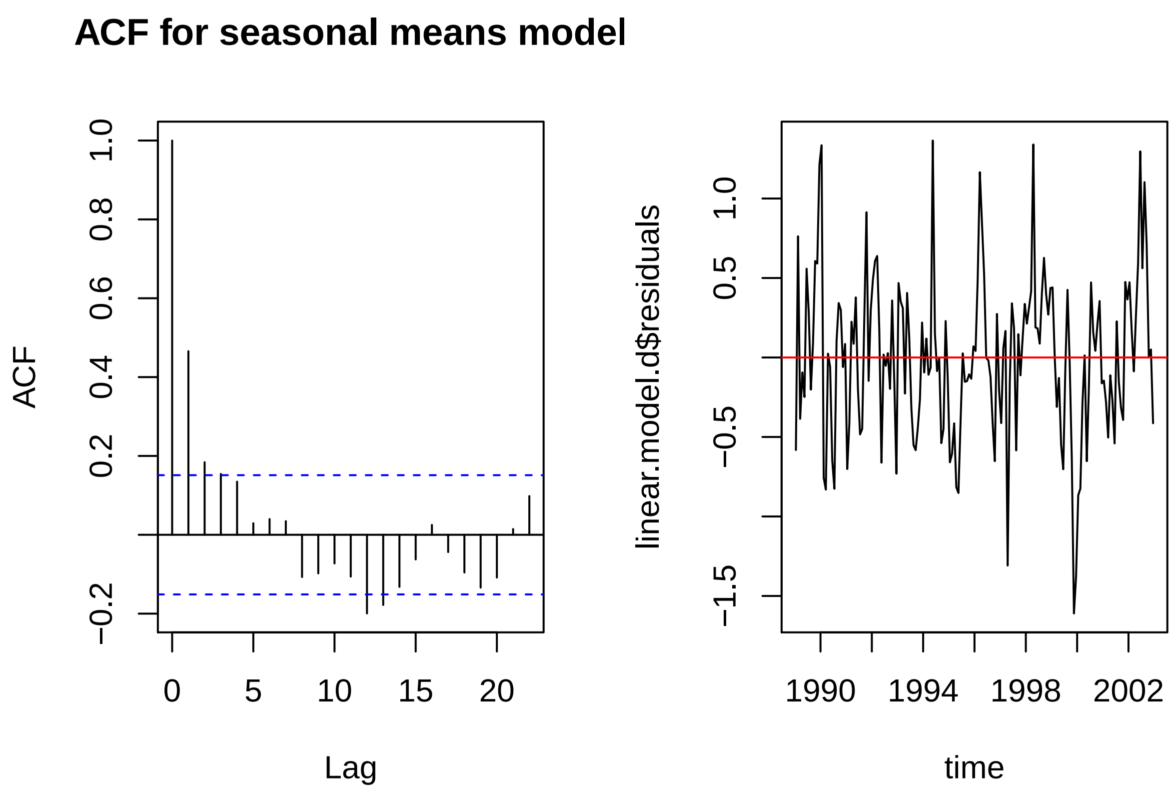


Figure 1: Comparing residuals for different kernel smoothers and linear model

```
    main = "ACF for seasonal means model")
plot(y = linear.model.d$residuals, x = rhine[,3],
      type = "l", xlab="time")
abline(h=0, col = "red")
```




```
par(mfrow = c(1,1))
```

e)

The previous model is the best one since we only consider two variables, time and month to estimate nitrogen concentration. The variable month is a factor and hence will not be removed even if it is significant. From the summary, but the best model given AIC is the same model as before.

```
#AIC used for stepwise feature selection
stepwise <- step(linear.model.d, direction = "backward")
```

```
## Start:  AIC=-202.02
## rhine[, 4] ~ rhine[, 3] + as.factor(rhine[, 2])
##
##                          Df Sum of Sq     RSS      AIC
## <none>                                 43.237 -202.023
## - as.factor(rhine[, 2]) 11    68.524 111.761  -64.477
## - rhine[, 3]             1   118.387 161.624   17.499
```

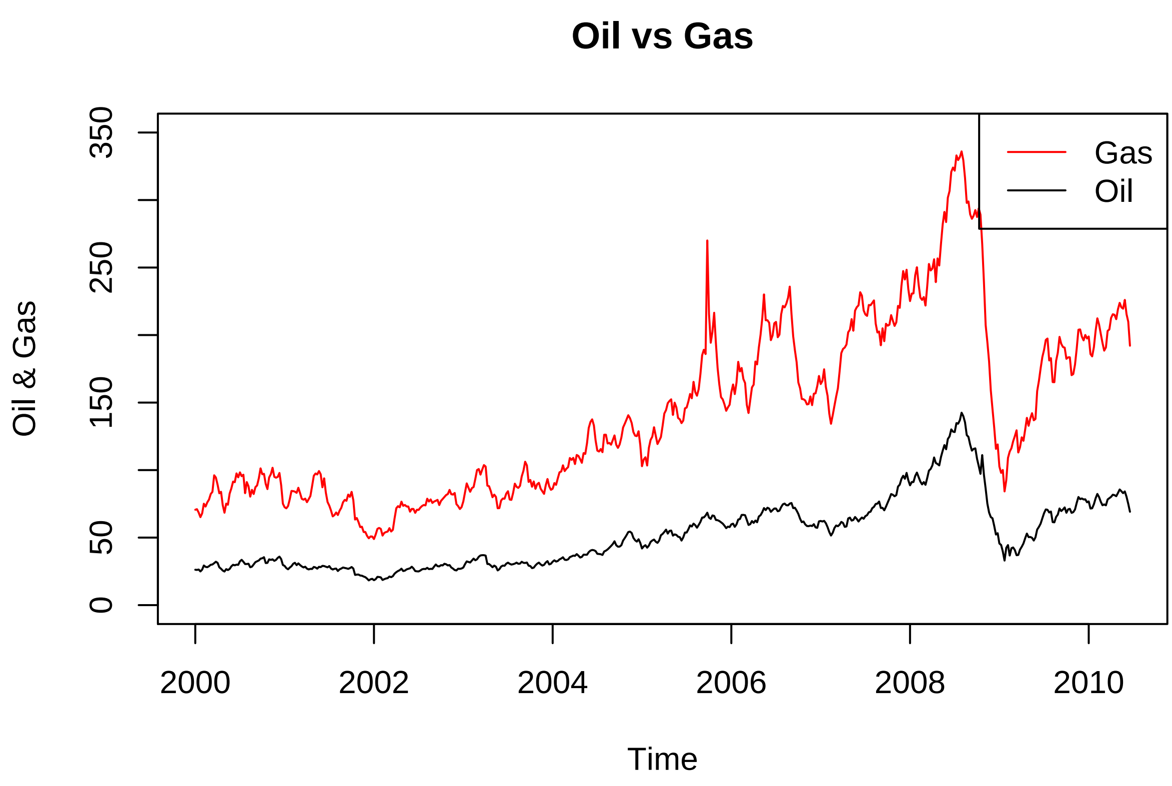
```
stepwiseSummary <- summary(stepwise)
```

Assignment 3

a)

Both Oil and Gas do not look stationary, especially the variable Gas since it is observed that the variance is dependent on time. The time series look like they are related as they both have an increasing trend followed by a crash after 2008.

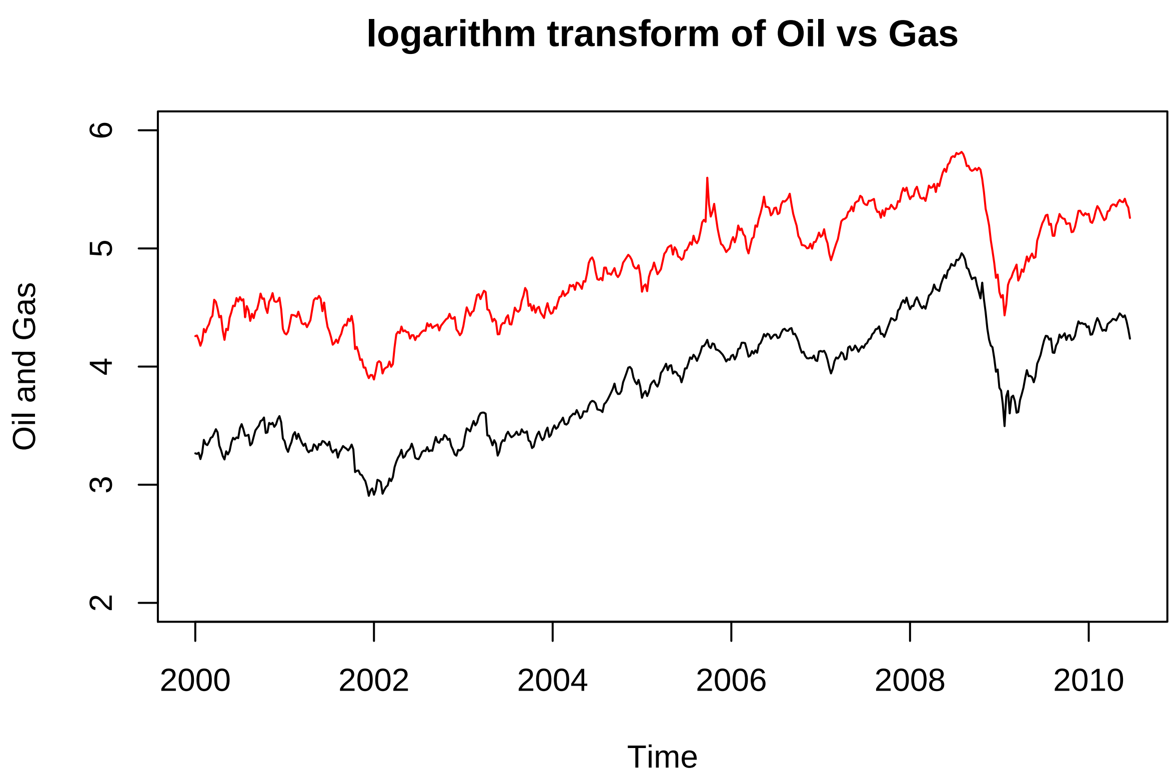
```
plot(oil, ylab = "Oil & Gas",
     xlab = "Time", main = "Oil vs Gas",
     ylim = c(0,350))
lines(gas, col = "red")
legend("topright", legend = c("Gas", "Oil"), lty = c(1,1), col = c("red", "black"))
```



b)

The log function reduces the exponential volatility in the data and hence, Log stabilizes the variance so its easier to compare the two different time series.

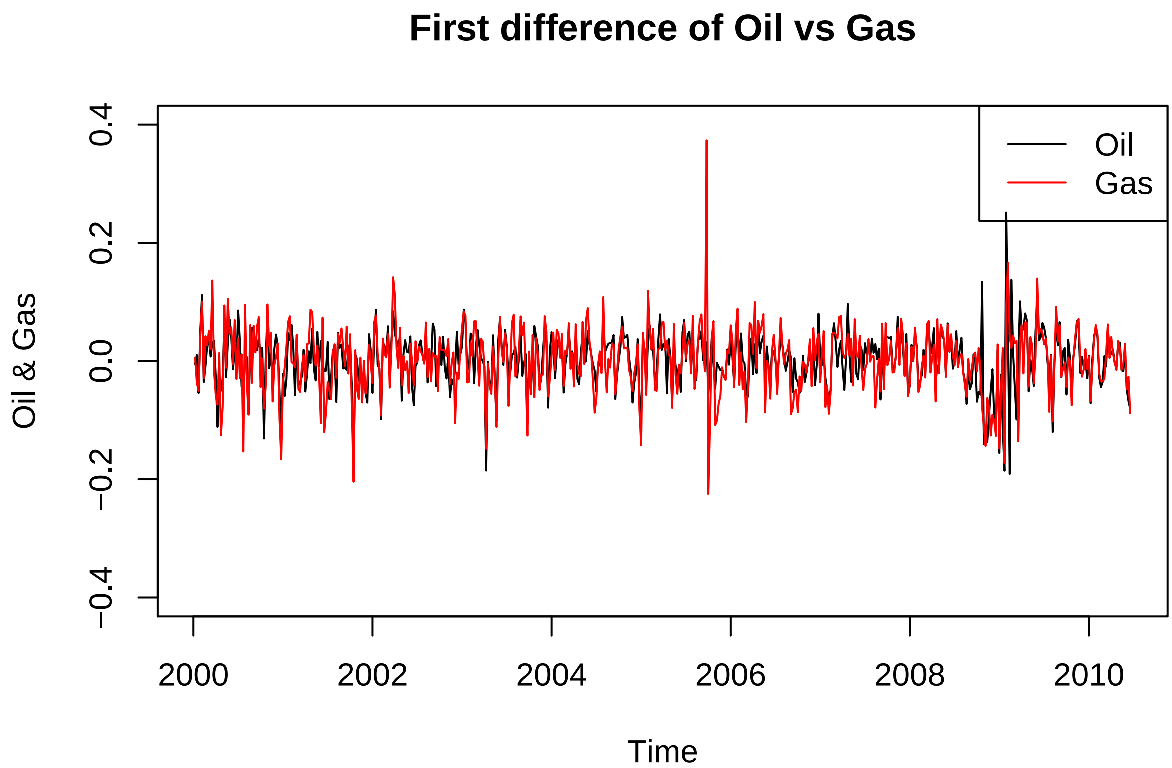
```r
#Log Transform of both time series
logOil <- log(oil)
logGas <- log(gas)
plot(logOil, ylab = "Oil and Gas",
     xlab = "Time", main = "logarithm transform of Oil vs Gas",
     ylim = c(2,6))
lines(logGas, col = "red")
```



c)

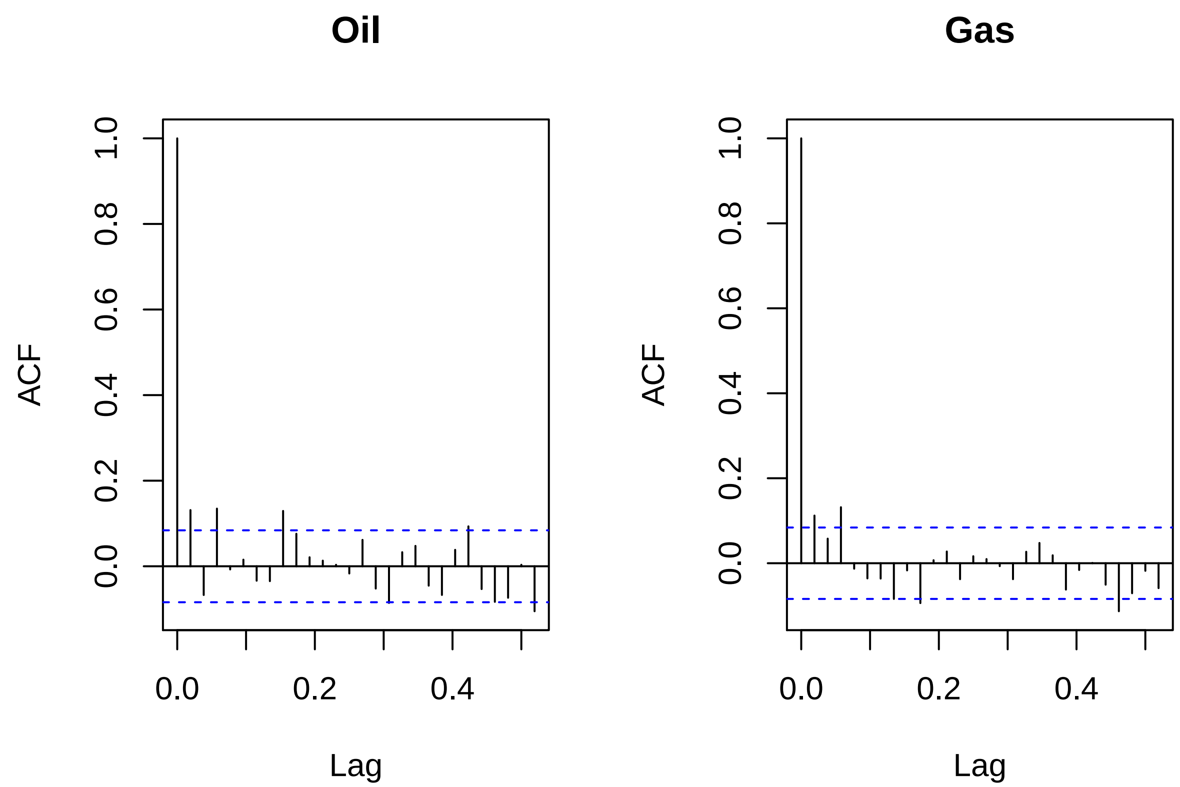
From Figure, the first difference of the series seem to be stationary even though we have a couple of peaks.

```r
oilDiff <- diff(logOil)
gasDiff <- diff(logGas)
plot(oilDiff, ylab = "Oil & Gas",
     xlab = "Time", main = "First difference of Oil vs Gas",
     ylim = c(-0.4,0.4))
lines(gasDiff, col = "red")
legend("topright", legend = c("Oil","Gas"), col = c("black", "red"),
       lty = c(1,1))
```



From the ACF plots it is observed that there isnt a significant correlation.

```
par(mfrow = c(1,2))
acf(oilDiff, main = "Oil")
acf(gasDiff, main = "Gas")
```

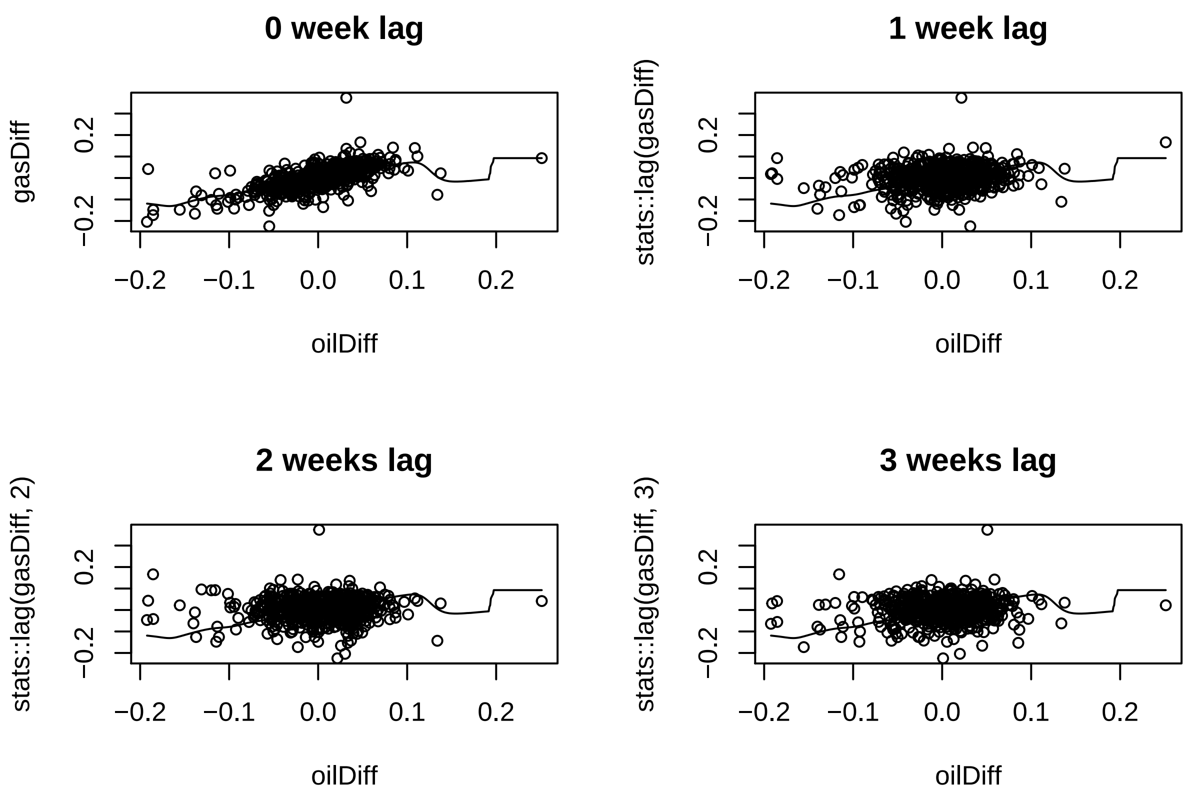



```
par(mfrow = c(1,1))
```

d)

It is observed from the fig of scatterplots, that the relationship seems to decrease as we increase the weekly lag but the difference is not large. The relationship seems to be somewhat linear with 0 lag. We do not have a linear relationship and few outliers are present.

```
par(mfrow = c(2,2))
plot(x = oilDiff , y = gasDiff, main = "0 week lag")
lines(ksmooth(x = oilDiff, y = gasDiff, bandwidth = 0.04, kernel = "normal"))
plot(x = oilDiff, y = stats::lag(gasDiff), main = "1 week lag")
lines(ksmooth(x = oilDiff, y = stats::lag(gasDiff), bandwidth = 0.04, kernel = "normal"))
plot(x = oilDiff, y = stats::lag(gasDiff, 2), main = "2 weeks lag")
lines(ksmooth(x = oilDiff, y = stats::lag(gasDiff,2), bandwidth = 0.04, kernel = "normal"))
plot(x = oilDiff, y = stats::lag(gasDiff, 3), main = "3 weeks lag")
lines(ksmooth(x = oilDiff, y = stats::lag(gasDiff,3), bandwidth = 0.04, kernel = "normal"))
```




```
par(mfrow = c(1,1))
```

e)

The following model is fit :

$$y_t = \alpha_0 + \alpha_1 I(x_t > 0) + \beta_1 x_t + \beta_2 x_{t-1} + w_t$$

According to the summary the lagged variable is not significant. The binary variable $I(x_t > 0)$ is significant at the 2nd confidence interval. The dummy variable has a positive coefficient which means that an increase in oil will also increase the gas. We can also see in Figure that the residuals seem to be constant over time with the exeption of some outliers.

```
df <-  ts.intersect(y = gasDiff, xt = oilDiff,
                    xt1 = stats::lag(oilDiff,1),xtbin = oilDiff>0)
model3 <- lm(y ~ xt + xt1 + xtbin, data = df)
summary(model3)
```

```
##
## Call:
## lm(formula = y ~ xt + xt1 + xtbin, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.18044 -0.02103  0.00003  0.02170  0.34592
##
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.006176   0.003470  -1.780   0.0757 .
## xt           0.694200   0.058898  11.786   <2e-16 ***
## xt1          0.012660   0.038729   0.327   0.7439
## xtbin        0.012376   0.005542   2.233   0.0259 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04202 on 539 degrees of freedom
## Multiple R-squared:  0.445,  Adjusted R-squared:  0.4419
## F-statistic: 144.1 on 3 and 539 DF,  p-value: < 2.2e-16
```

```r
par(mfrow = c(1,2))
plot(residuals(model3), ylab = "Residuals",
     xlab = "Time",  main = "Residual Pattern")
abline(h = 0)
acf(residuals(model3), main = "ACF")
```