Lab 1

Sridhar Adhikarla (sriad858)

September 16, 2019

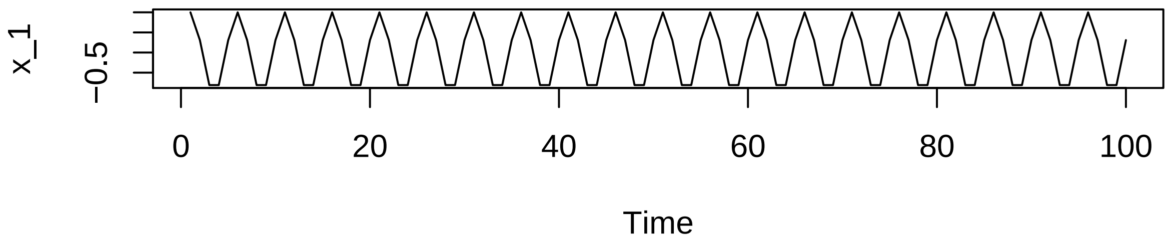
Assignment 1

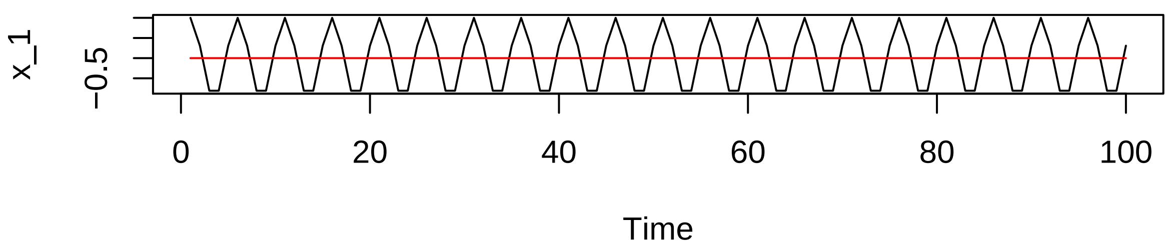
Q1

The first time series has no randomness in it. It is just a cos function of (constant x time). Since cos(2 x pi) is 1 amd we are multiplying 2 x pi with t/5 to generate the time series, at every t that is a multiple of 5 the time series reaches 1. Applying the smoothing filter to this time series results in a horizontal straight line close to zero.

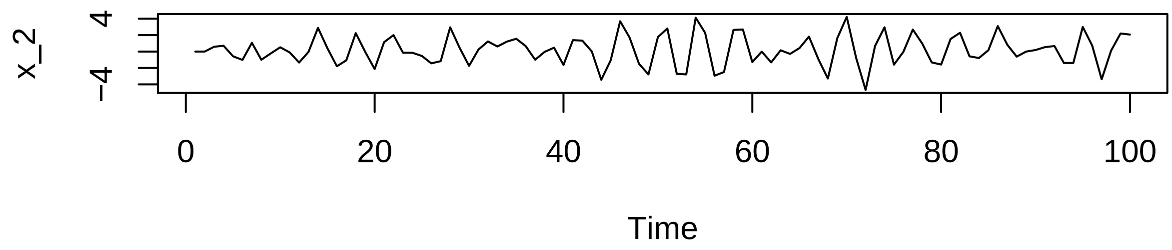
The second time series is completely random as it is a weighted sum of white noise up to that time step. It is hard to spot a trend in this time series as it is a random walk model. Applying a smoothing filter to this model reduces the frequency of the oscillations.

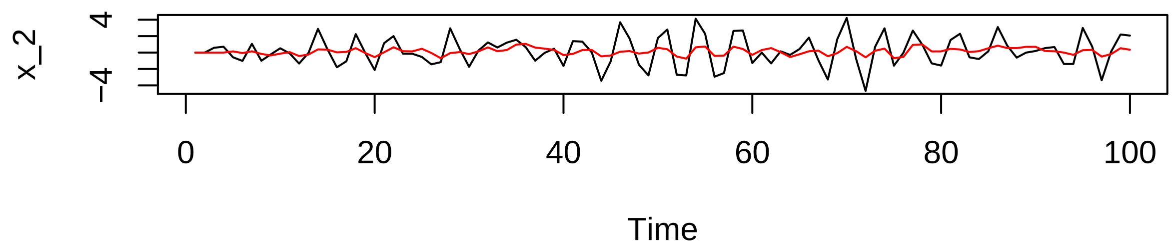
```
set.seed(12345)

#A1
par(mfrow=c(2,1))
##a
#time
t = 0:99
#time series 1
x_1 = cos((2*pi*t)/5)
x_1 = ts(x_1)
plot(x_1)

#applying smoothing filter
v_1 = NULL
v_1[1:4] = 0
for (i in 5:100) {
  v_1[i] = 0.2*(x_1[i] + x_1[i-1] + x_1[i-2] + x_1[i-3] + x_1[i-4])
}
v_1 = ts(v_1)
plot(x_1)
lines(v_1, col="red")
```






```
#time series 2
x_2 = NULL
x_2[1] = 0
x_2[2] = 0
for (i in 3:100) {
  x_2[i] = -(x_2[i-2]*0.8) + rnorm(1)
}
x_2 = ts(x_2)
plot(x_2)
#applying smoothing filter
v_2 = NULL
v_2[1:4] = 0
for (i in 5:100) {
  v_2[i] = 0.2*(x_2[i] + x_2[i-1] + x_2[i-2] + x_2[i-3] + x_2[i-4])
}
v_2 = ts(v_2)
plot(x_2)
lines(v_2, col="red")
```



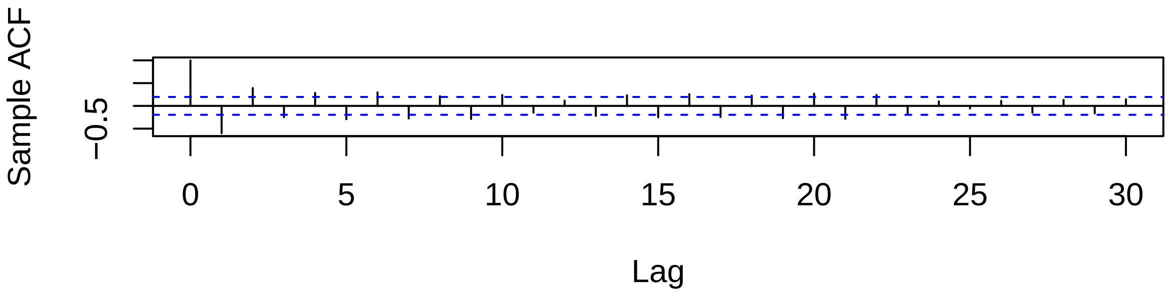


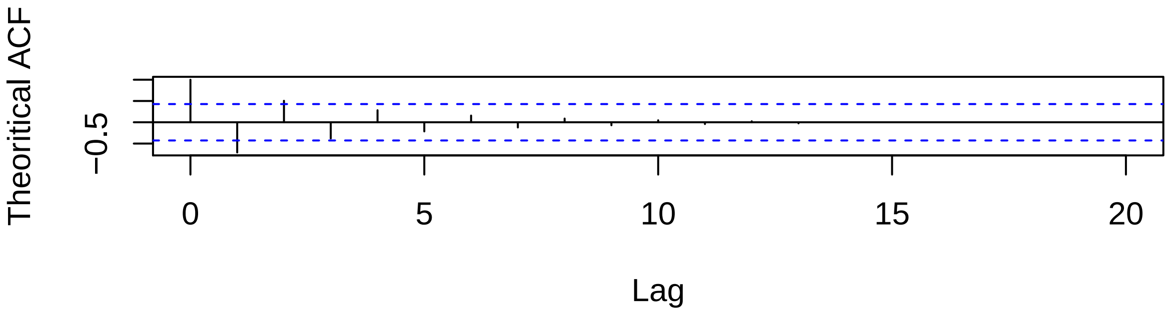
Q2

Q3

The main difference between the theoritical and the sample ACF is that the autocorrelation in the theoritical ACF cuts off after lag 3. Whereas in the sample ACF there appears to be some autocorrelation even upto lag 20. It appears to start decreasing after lag 20.

```
set.seed(54321)
par(mfrow=c(2,1))
#generte model
model = arima.sim(n=100, list(ar=c(-3/4), ma=c(0, -1/9)))

acf(model, ylab="Sample ACF", main="", lag.max = 30)
acf(ARMAacf(ar=c(-3/4), ma=c(0, -1/9), lag.max = 20), main="", ylab="Theoritical ACF", xlim=c(0, 20))
```

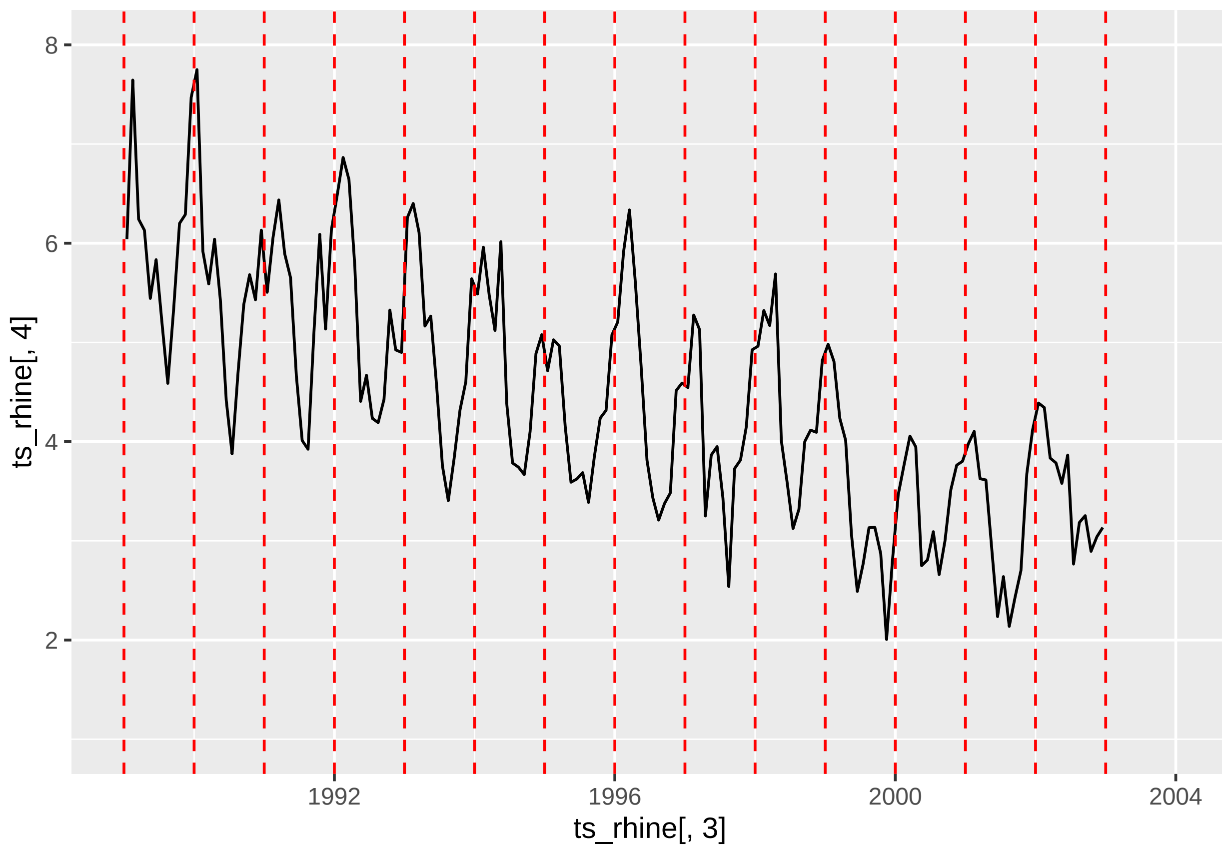




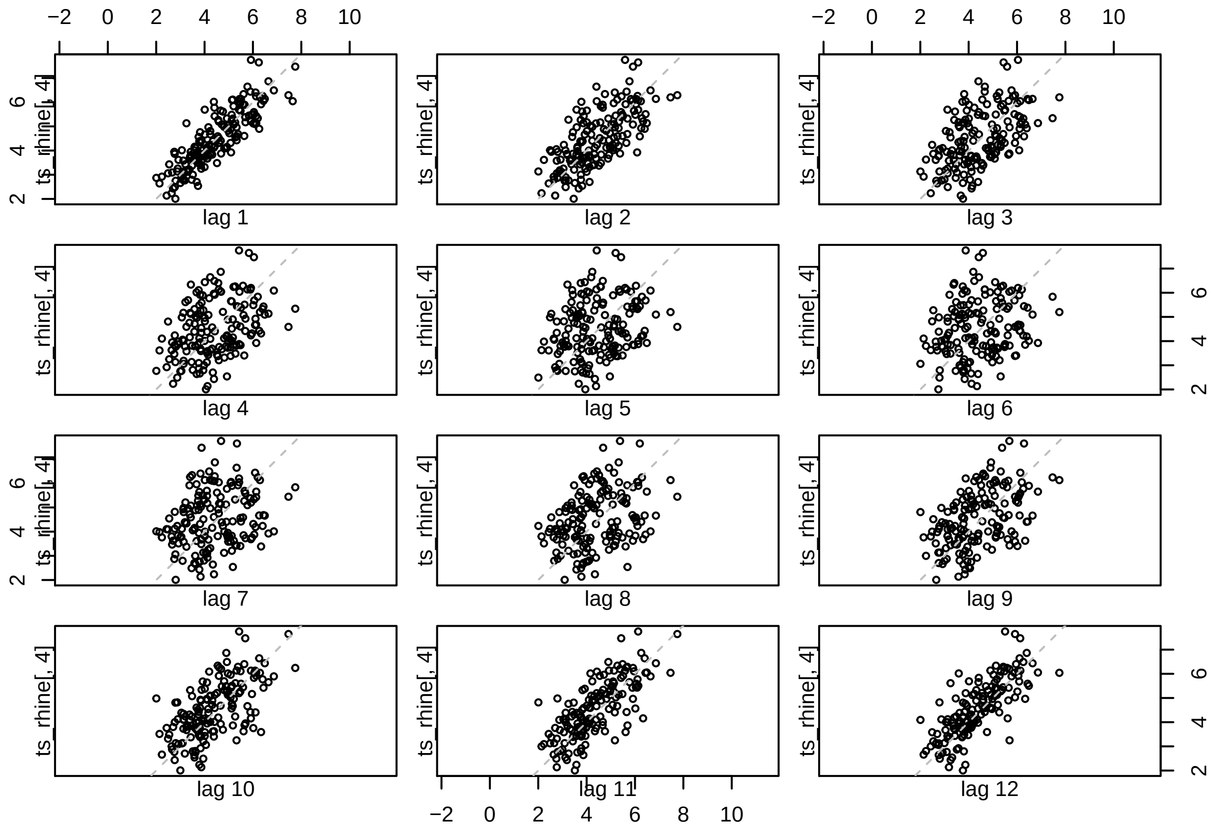
Assignment 2

Q1

```r
rhine = read.csv("Rhine.csv", header = T, sep = ";",dec = ",")

ts_rhine = ts(rhine)
ggplot() +
  geom_line(aes(ts_rhine[,3], ts_rhine[,4])) +
  geom_vline(aes(xintercept=1989:2003), col="red", lty=2) +
  xlim(1989, 2004) + ylim(1, 8)
```




```
lag.plot(x = ts_rhine[,4], lag = 12)
```

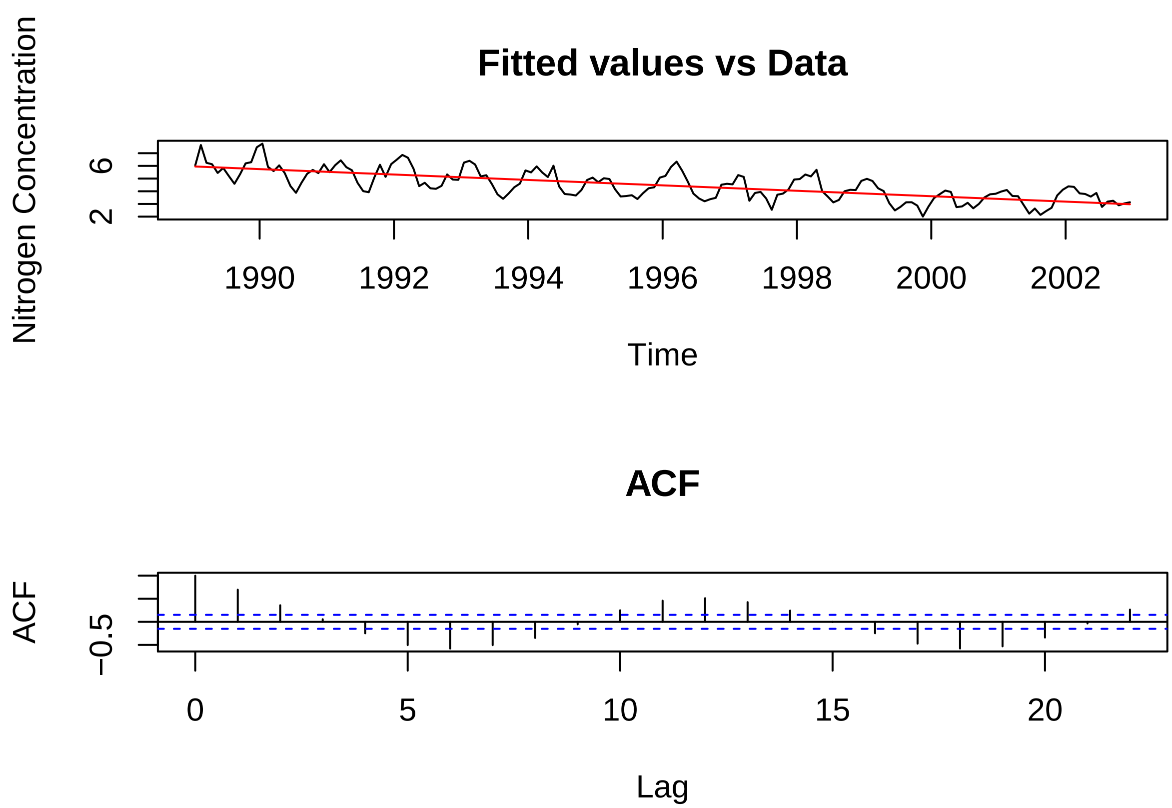


Q2

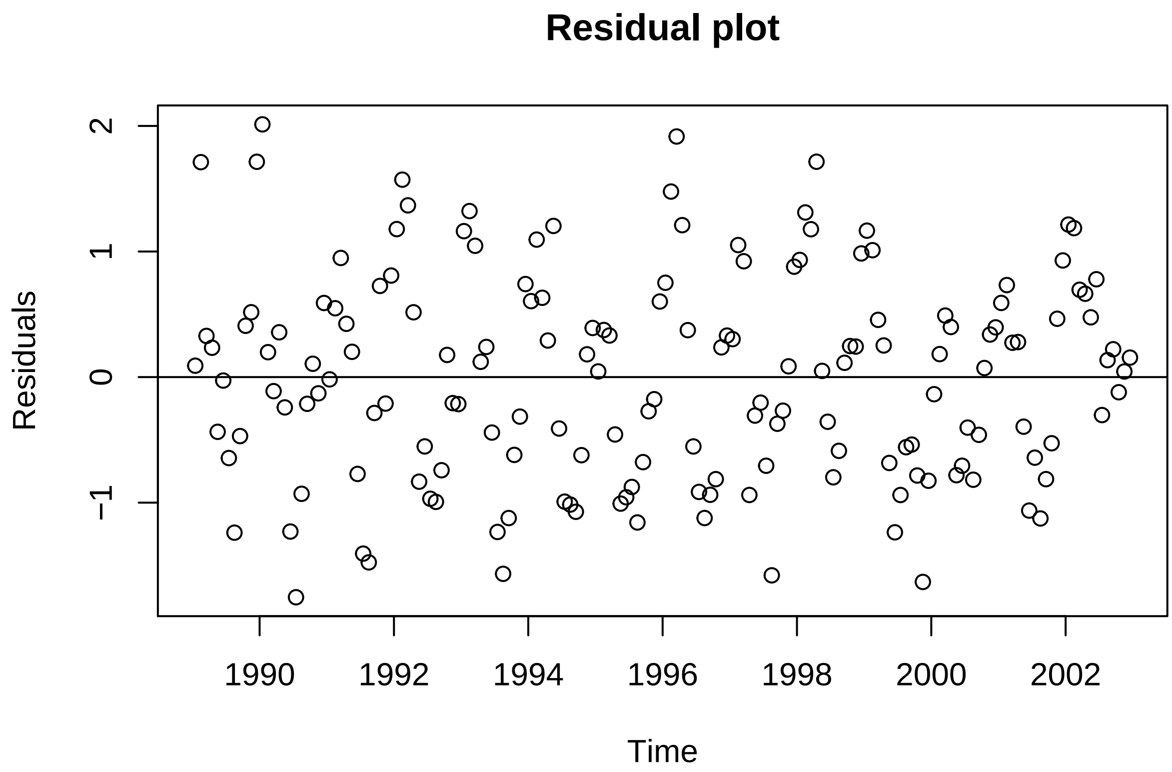
```
#Eliminating the trend by fitting a linear model
lrMod <- lm(TotN_conc ~ Time, data = rhine)
summary(lrMod)
```

```
##
## Call:
## lm(formula = TotN_conc ~ Time, data = rhine)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.75325 -0.65296  0.06071  0.52453  2.01276
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 430.70725   31.26570   13.78   <2e-16 ***
## Time         -0.21355    0.01566  -13.63   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8205 on 166 degrees of freedom
## Multiple R-squared:  0.5282, Adjusted R-squared:  0.5254
## F-statistic: 185.9 on 1 and 166 DF,  p-value: < 2.2e-16
```

```r
#Plot fitted values vs data
par(mfrow = c(2,1))
plot(y = rhine[,4], x = rhine[,3], type = "l",
main = "Fitted values vs Data",
ylab = "Nitrogen Concentration",
xlab = "Time")
lines(y = lrMod$fitted.values, x = lrMod$model$Time,
col = "red")

#Plot ACF
acf(lrMod$residuals, main = "ACF")
```




```r
par(mfrow=c(1,1))
#Plot Residuals
plot(lrMod$residuals, main = "Residual plot",
x = lrMod$model$Time, ylab = "Residuals",
xlab = "Time", type = "p")
abline(h = 0)
```



Q3

Q4

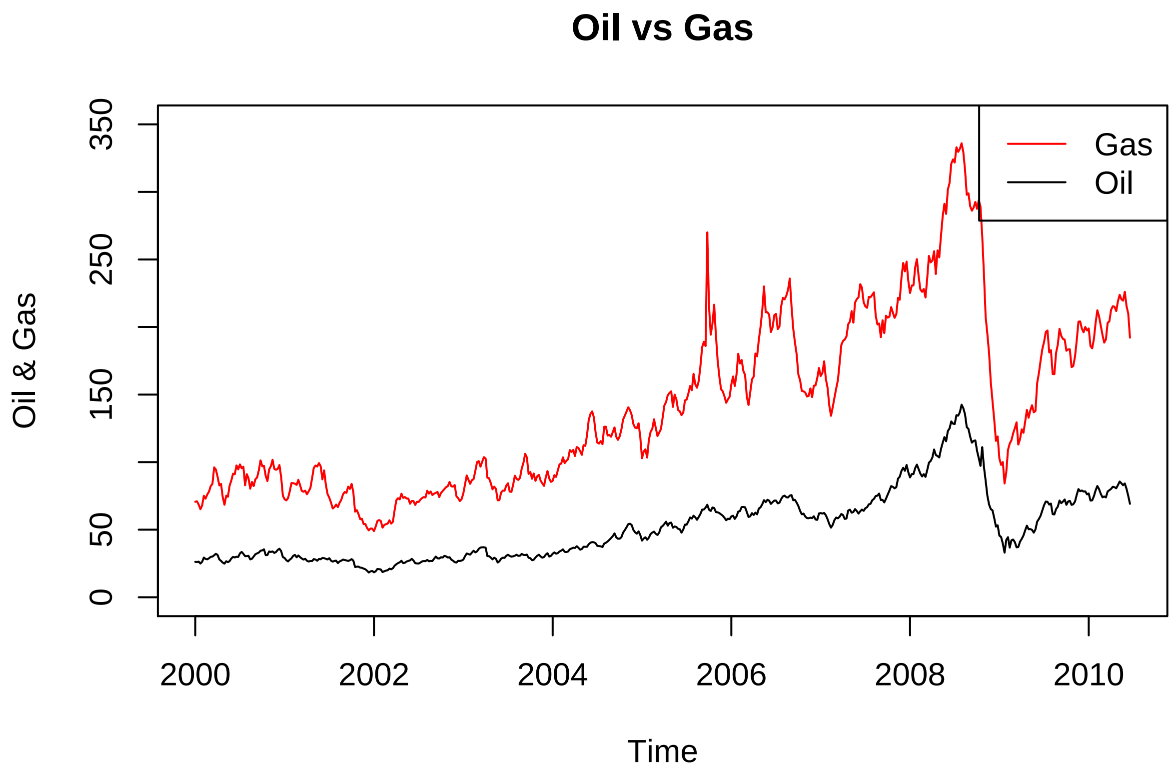
Q5

Assignment 3

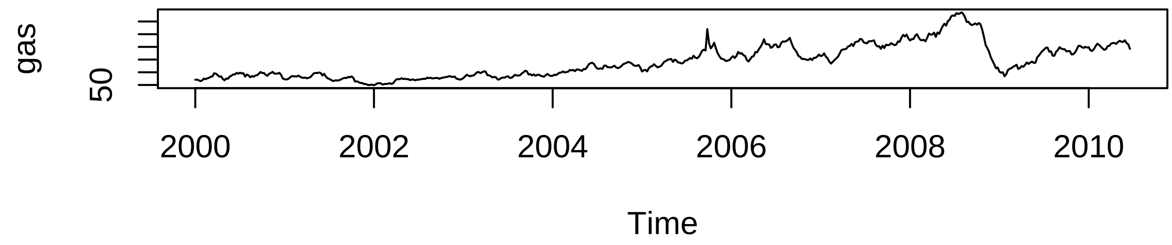
Q1

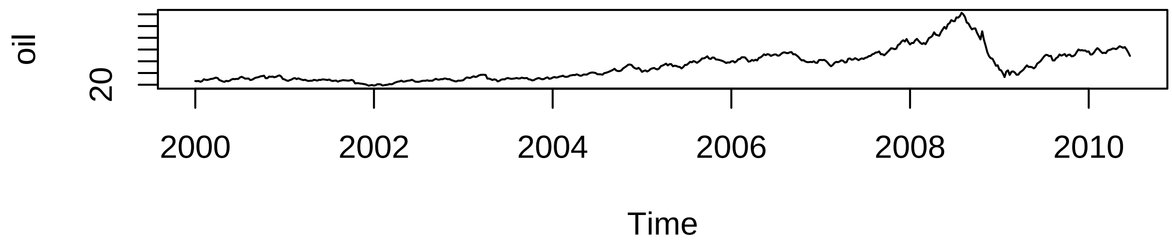
Both the time series for gas and oil are not stationary. They both follow a similar increasing trend, with a dip after 2008. Though gas seems to have a larger variance than oil, it is just because they both are not in the same units. Oil prices given are in Dollars per barrel and gas prices are in Cents per gallon. Both the time series have highs and lows during the same time periods in the time series data. This becoms clearer when we plot the two in different plots.

```
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"))
```




```
par(mfrow=c(2,1))
plot(gas)
plot(oil)
```

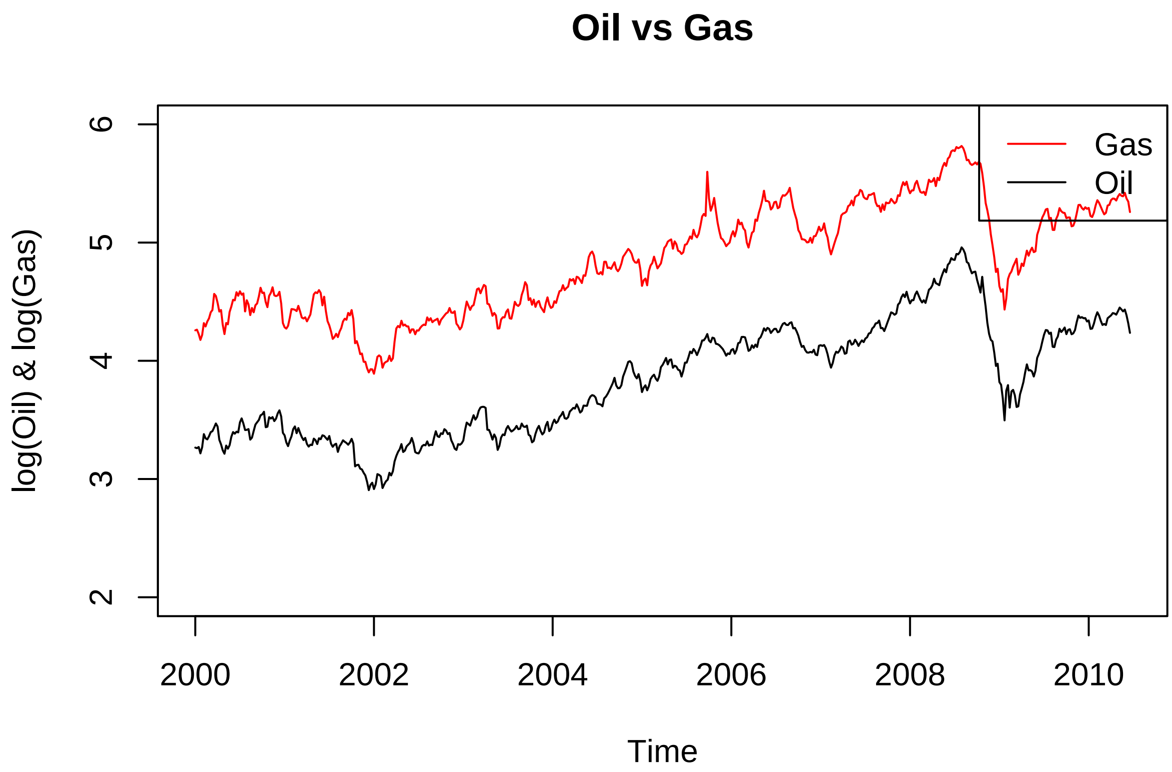




Q2

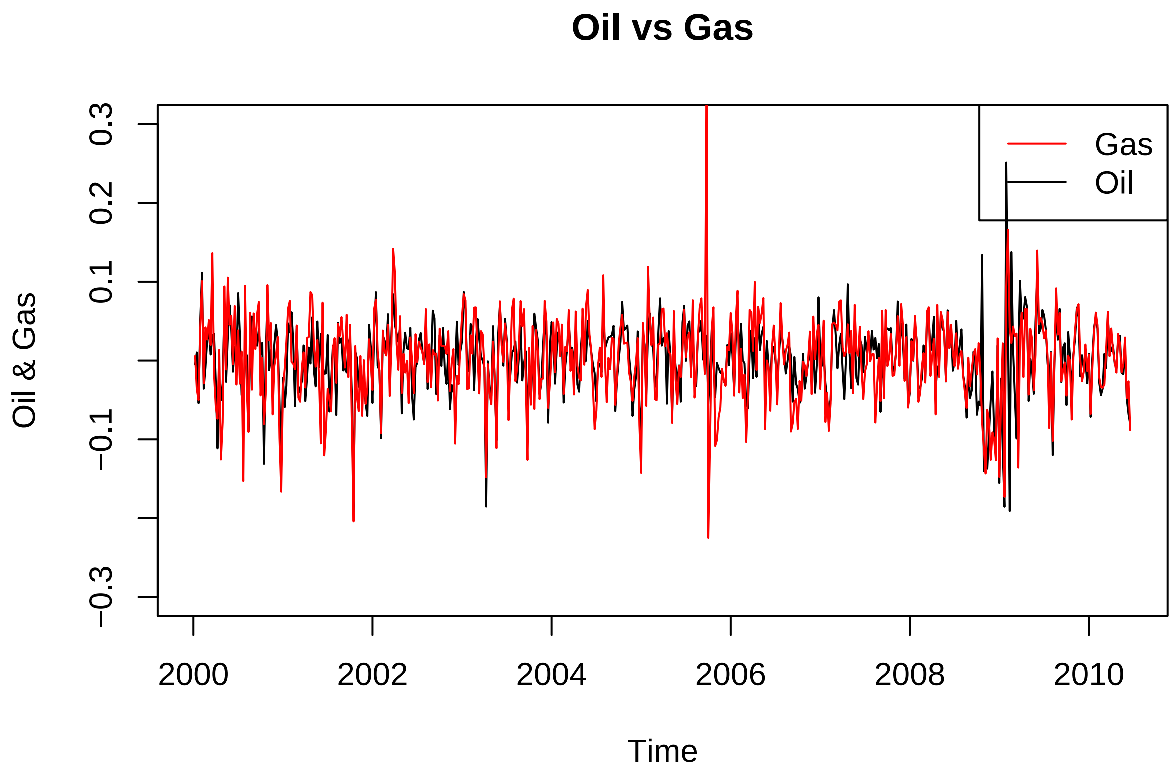
It becomes clearer that both the time series are following the same trend when we plot the log transformation of the data.

```r
plot(log(oil), ylab = "log(Oil) & log(Gas)",
xlab = "Time", main = "Oil vs Gas",
ylim = c(2,6))
lines(log(gas), col = "red")
legend("topright", legend = c("Gas", "Oil"), lty = c(1,1), col = c("red", "black"))
```

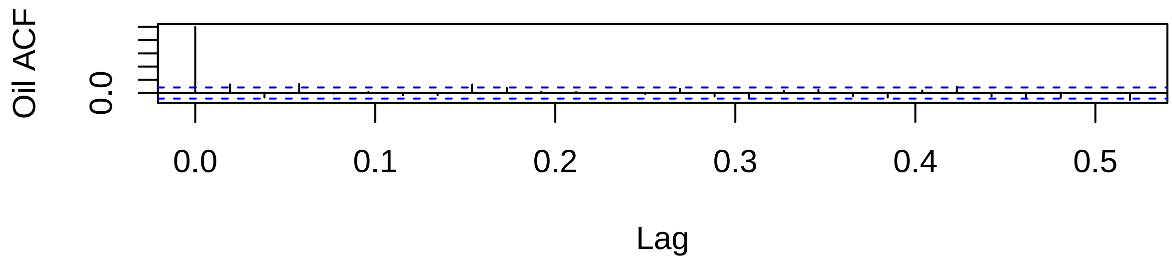


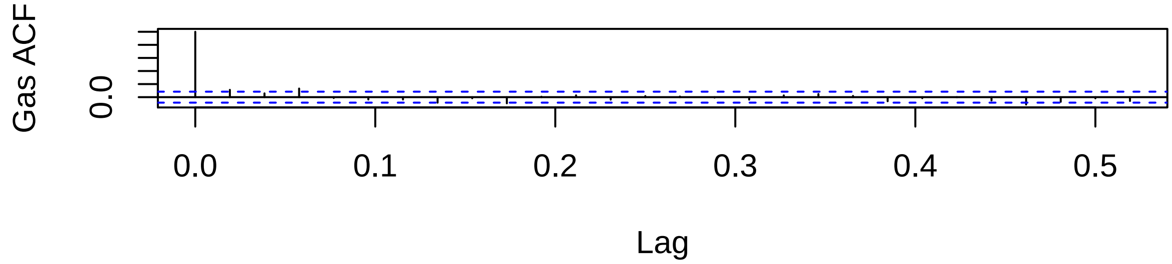
Q3

```r
xt = diff(log(oil))
yt = diff(log(gas))

plot(xt, ylab = "Oil & Gas",
xlab = "Time", main = "Oil vs Gas",
ylim = c(-0.3,0.3))
lines(yt, col = "red")
legend("topright", legend = c("Gas", "Oil"), lty = c(1,1), col = c("red", "black"))
```



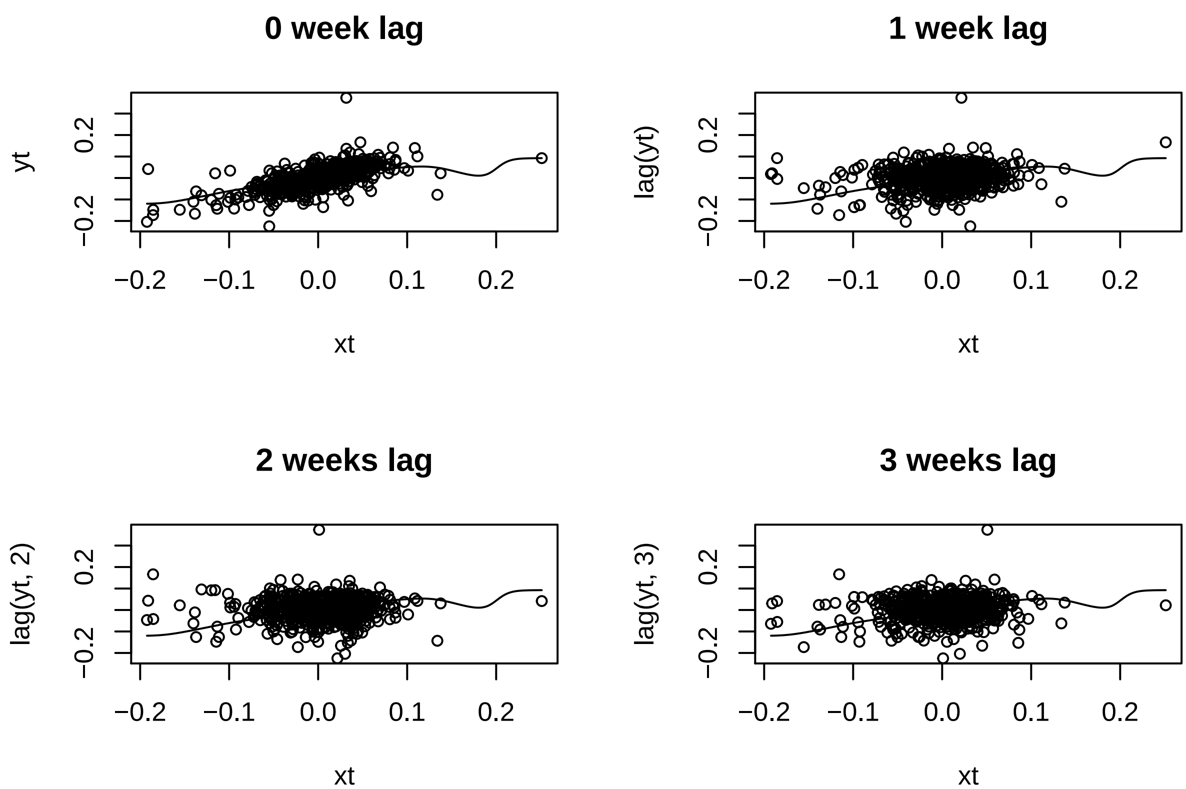

```r
par(mfrow=c(2,1))
acf(xt, ylab="Oil ACF", main="")
acf(yt, ylab="Gas ACF", main="")
```





Q4

```r
par(mfrow = c(2,2))
plot(x = xt , y = yt, main = "0 week lag")
lines(ksmooth(x = xt, y = yt, bandwidth = 0.08, kernel = "normal"))

plot(x = xt, y = lag(yt), main = "1 week lag")
lines(ksmooth(x = xt, y = lag(yt), bandwidth = 0.08, kernel = "normal"))

plot(x = xt, y = lag(yt, 2), main = "2 weeks lag")
lines(ksmooth(x = xt, y = lag(yt,2), bandwidth = 0.08, kernel = "normal"))

plot(x = xt, y = lag(yt, 3), main = "3 weeks lag")
lines(ksmooth(x = xt, y = lag(yt,3), bandwidth = 0.08, kernel = "normal"))
```



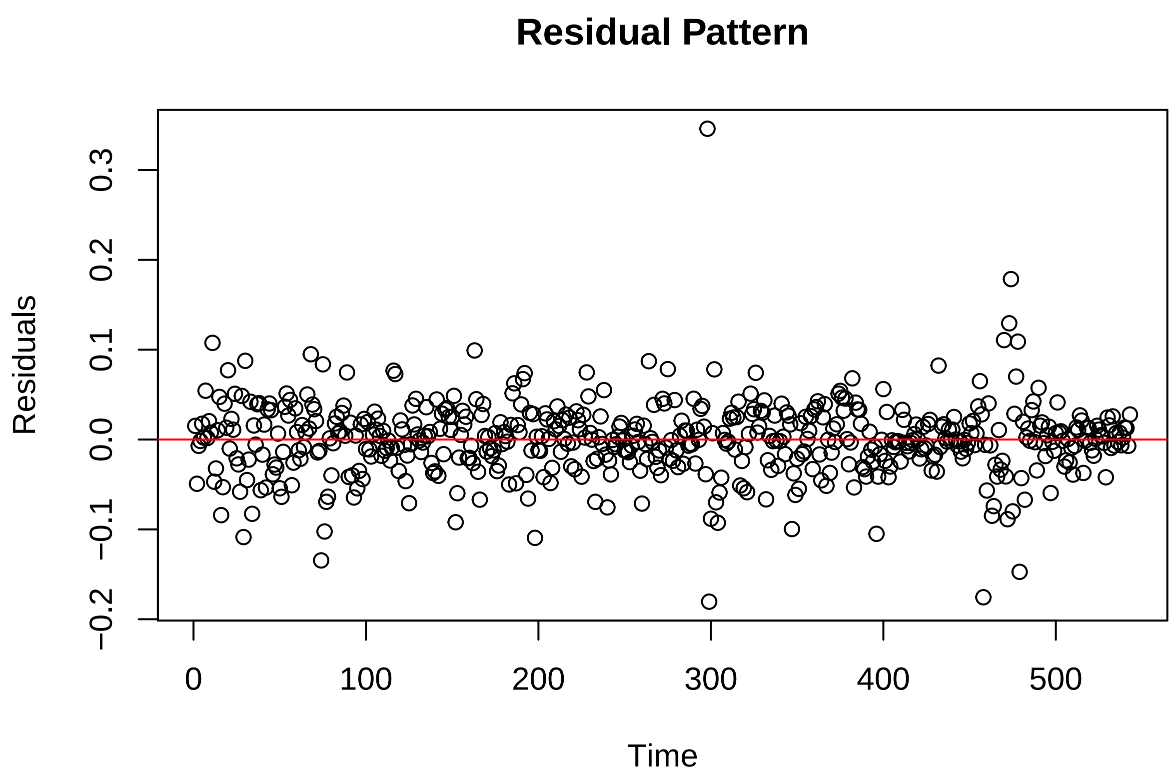
Q5

```r
df <- ts.intersect(y = yt, xt = xt, xt1 = lag(xt,1), xtbin = xt>0)

lrMod = lm(y ~ xt + xt1 + xtbin, data = df)
summary(lrMod)
```

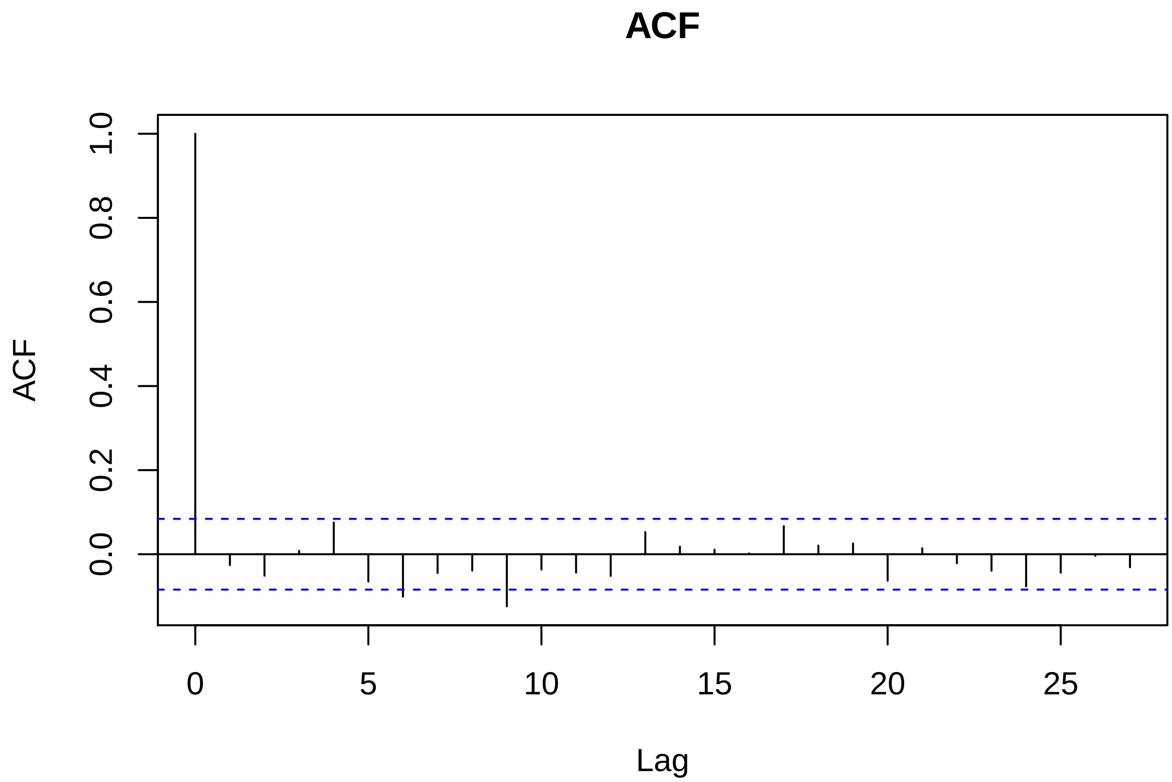
```
##
## Call:
## lm(formula = y ~ xt + xt1 + xtbin, data = df)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.18044 -0.02103  0.00003  0.02170  0.34592
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.006176   0.003470  -1.780   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
plot(residuals(lrMod), ylab = "Residuals",
xlab = "Time", main = "Residual Pattern")
abline(h = 0, col="red")
```




```r
acf(residuals(lrMod), main = "ACF")
```


ACF
ACF
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
25
Lag