Lab A

Thijs Quast, Saewon Jun

9/9/2019

Contents

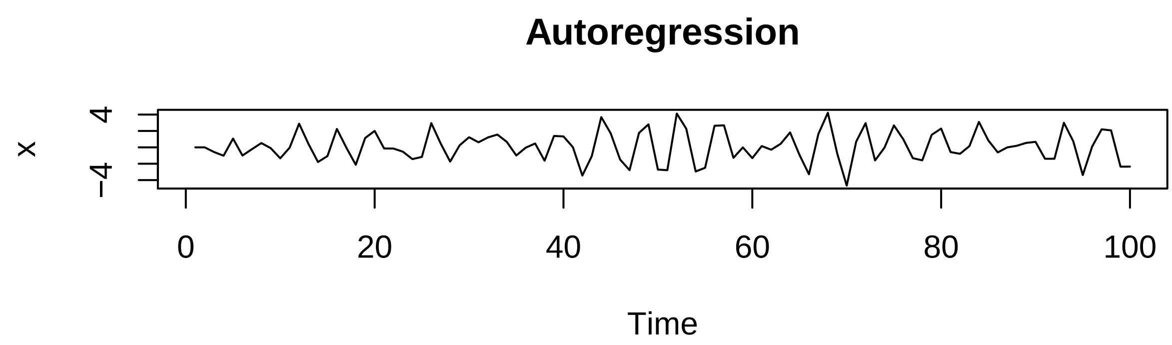
Assignment 1

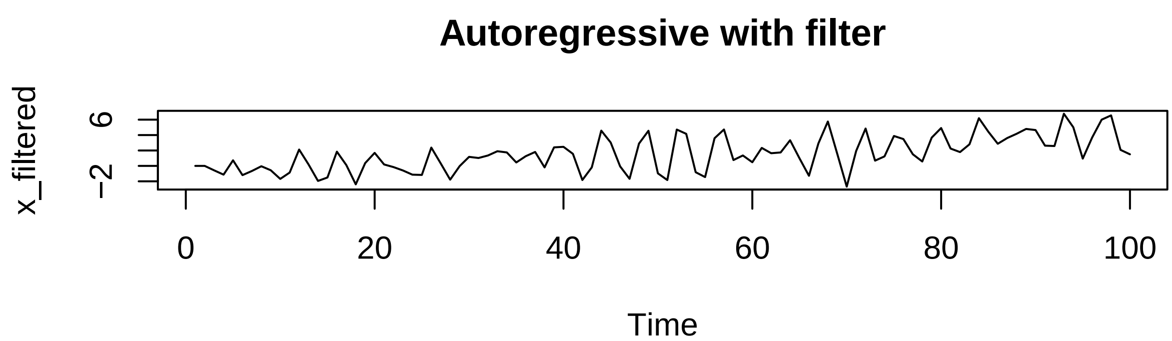
a

```
set.seed(12345)
w <- rnorm(100)

x <- filter(w, filter = c(0, -0.8), method = "recursive")
# x0 = x1 = 0
x[1] <- 0
x[2] <- 0
```

```
x_filtered <- filter(x, filter = c(0.2, 0.2, 0.2, 0.2, 0.2), method = "recursive", sides = 1)
```

```
par(mfrow=c(2,1))
plot.ts(x, main = "Autoregression")
plot.ts(x_filtered, main="Autoregressive with filter")
```





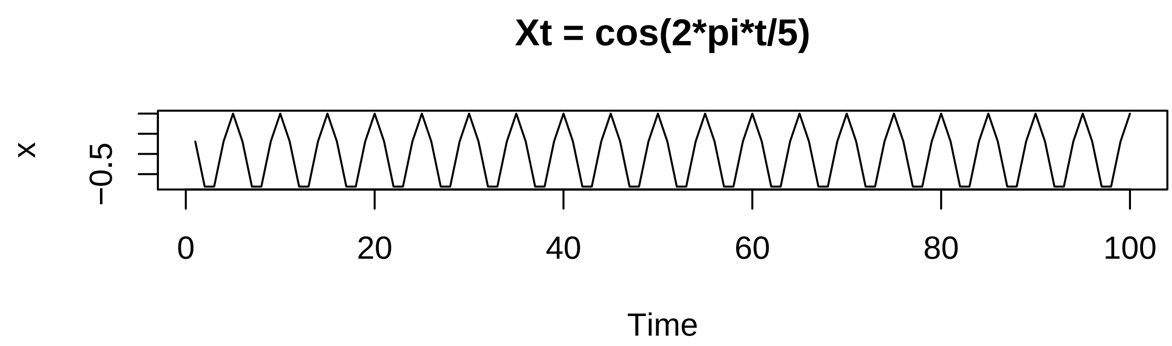
In this model it seems as if the smoothing filter has created a slight drift in the time series.

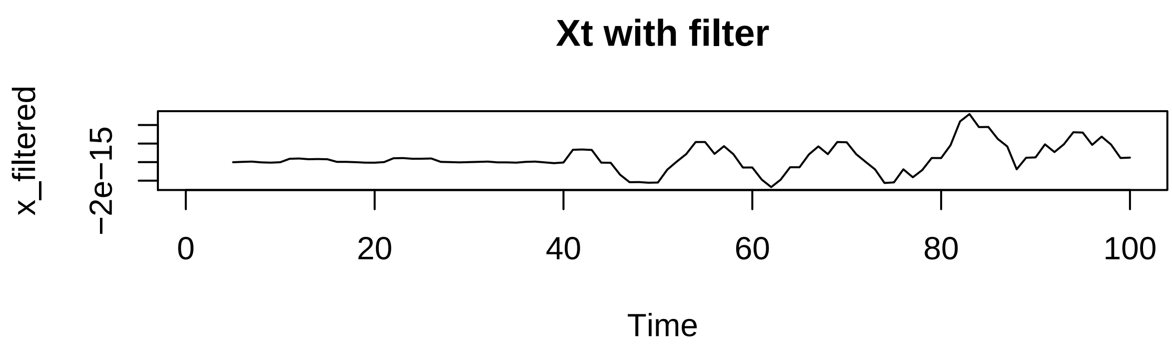
```
t <- 100
x <- c()

for (i in 1:t){
```

```
  x[i] <- cos((2*pi*i)/5)
}
```

```
x_filtered <- filter(x, filter = c(0.2, 0.2, 0.2, 0.2, 0.2), method = "convolution", sides = 1)
```

```
par(mfrow=c(2,1))
plot.ts(x, main = "Xt = cos(2*pi*t/5)")
plot.ts(x_filtered, main="Xt with filter")
```





In this timeseries, the filter has changed the pattern of the time series.

b

$$X_t - 4X_{t-1} + 2X_{t-2} + X_{t-5} = W_t + 3W_{t-2} + W_{t-4} - 4W_{t-6}$$
$$X_t = 4X_{t-1} - 2X_{t-2} - X_{t-5} + W_t + 3W_{t-2} + W_{t-4} - 4W_{t-6}$$

$$X_t - 4X_{t-1} + 2X_{t-2} + X_{t-5} = W_t + 3W_{t-2} + W_{t-4} - 4W_{t-6}$$
$$\phi(B)X_t = \phi(B)W_t$$
$$\phi(B) = 1 - 4B + 2B^2 + B^5$$

```r
# Causal
# Coefficients in front of B's for 3rd and 4th power are 0
z <- c(1, -4, 2, 0, 0,  1)
polyroot(z)
```

```
## [1]  0.2936658+0.000000i -1.6793817+0.000000i  1.0000000-0.000000i
## [4]  0.1928579-1.410842i  0.1928579+1.410842i
```

$$\theta(B) = 1 + 3B^2 + B^4 - 4B^6$$

```r
# Invertible
# Coefficients in front of B's for 3rd and 5th power are 0
z <- c(1, 0, 3, 0, 1, 0, -4)
polyroot(z)
```

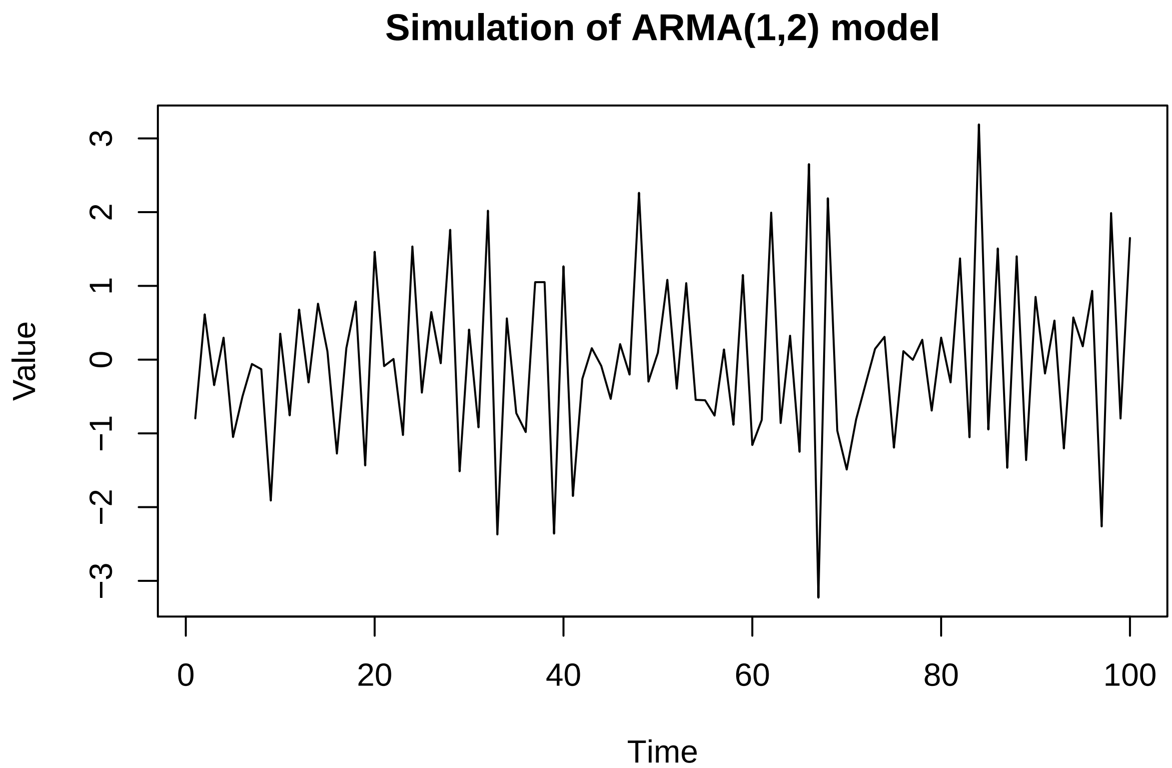
```
## [1]  0.1375513+0.6735351i -0.1375513+0.6735351i -0.1375513-0.6735351i
## [4]  0.1375513-0.6735351i  1.0580446+0.0000000i -1.0580446+0.0000000i
```

For the time series to be causal and invertible, the unit roots for the AR process should be outside the unit circle and the unit roots for the MA process as well. First check causality, the first root is within the unit circle, so the process is not causal. Check invertability; first root is within unit circle, so process is not invertible.

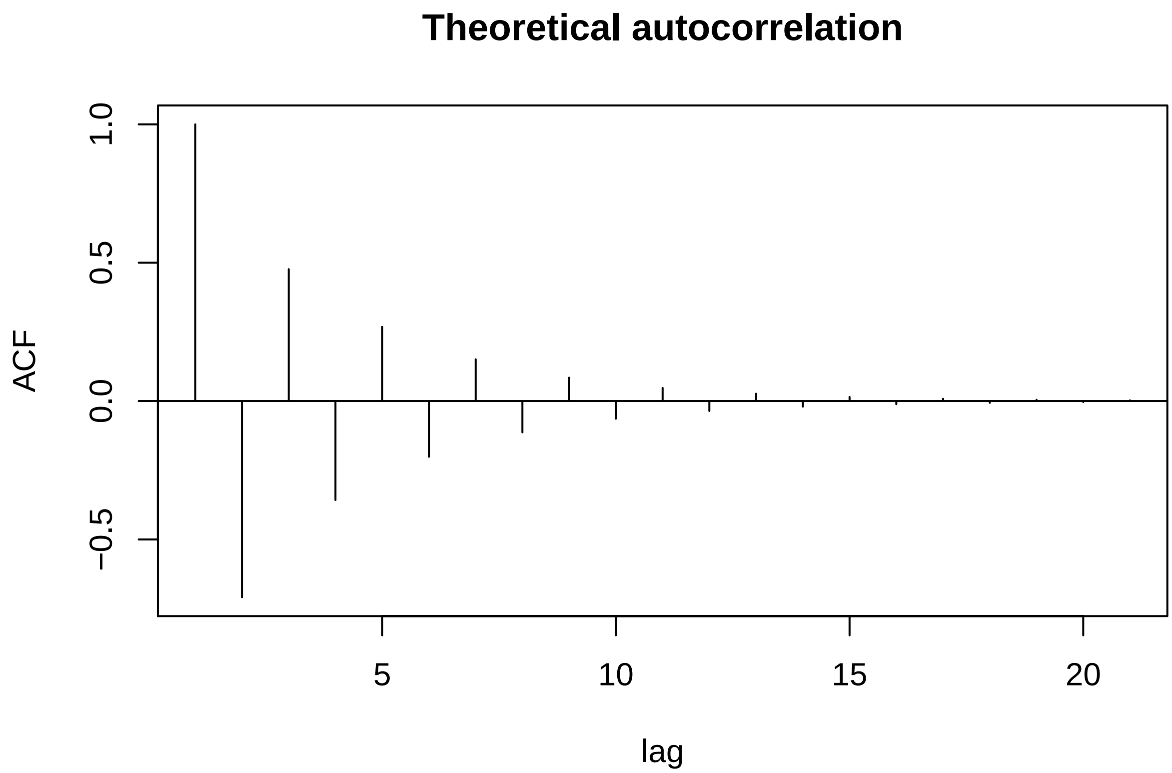
c

$$X_t + \frac{3}{4}X_{t-1} = W_t - \frac{1}{9}W_{t-2}$$
$$X_t = -\frac{3}{4}X_{t-1} + W_t - \frac{1}{9}W_{t-2}$$
$$ARMA(1,2)$$

```r
set.seed(54321)
arma2 <- arima.sim(list(order = c(1,0,2),ar=c(-0.75), ma=c(0, -(1/9))), n = 100)
plot(arma2, xlab="Time", ylab = "Value", main="Simulation of ARMA(1,2) model")
```

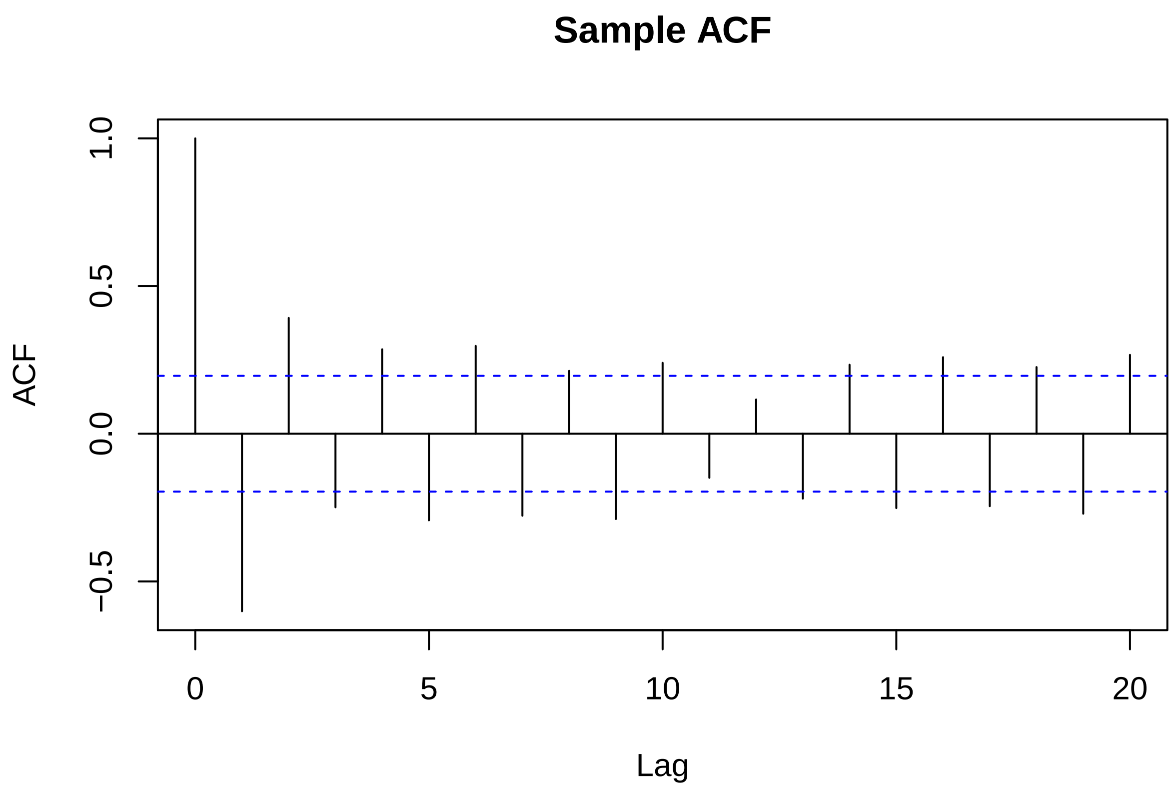



```r
set.seed(54321)
# Theoretical Autocorrelation
ACF = ARMAacf(ar = c(-0.75), ma = c(0, -(1/9)), lag.max = 20)
plot(ACF, type="h", xlab="lag", main = "Theoretical autocorrelation") +
  abline(h = 0)
```




```
## integer(0)
```

```r
# Sample Autocorrelation
acf_sample <- acf(arma2, main="Sample ACF")
```

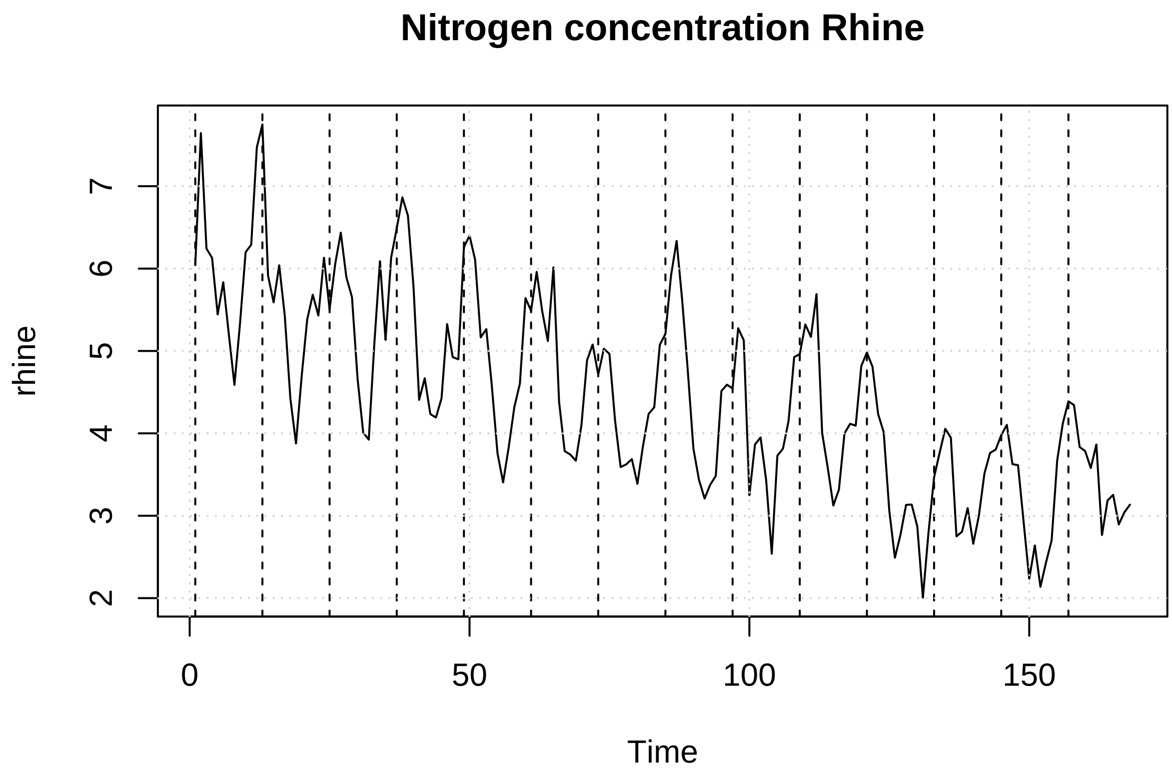


The theoretical autocorrelation seems to be more desirable, as it diminishes to zero after approximately 15 lags, whereas the sample autocorrelatoin seems to be agove the blue lines, still at lag 20. Meaning in practice there is more autocorrelation than desirable.

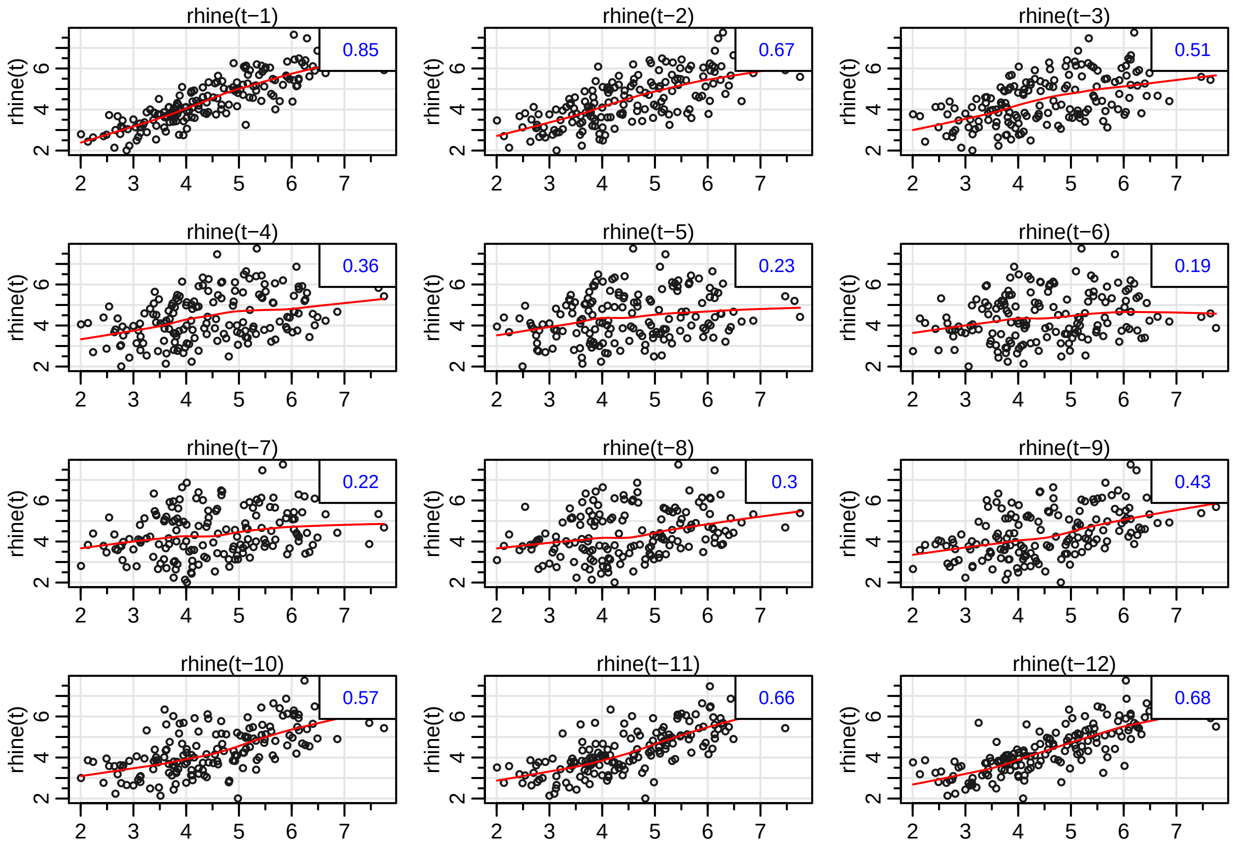
Assignment 2

```r
rhine <- read.csv2("Rhine.csv")
rhine_data <- rhine
rhine <- ts(rhine$TotN_conc)
```

```r
# Plot time series, with vertical lines to divide in years.
plot.ts(rhine, main="Nitrogen concentration Rhine")
abline(v=seq(1,168,12),  lty=2)
grid()
```




```
library(astsa)
lag1.plot(rhine, 12)
```



In the time series plot, one can see that there is a linear downward trend overall. Whereas there seems to be some seasonality as the is a pattern in the time series that seems to repeat itself. One can see that the concentration is the highest in the beginning and end of the year. Variance does not really seem to change over time. Data seems to be autocorrelated with lags, 1, 2, 3, 10, 11 and 12.

b

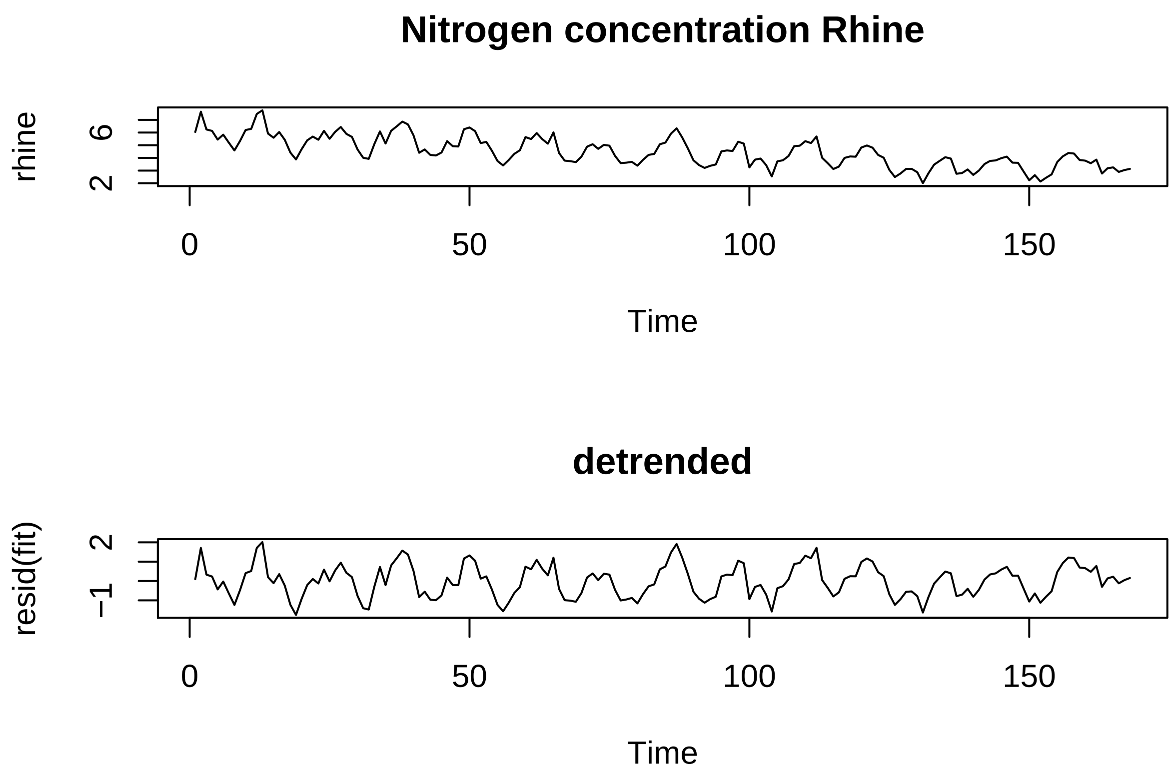
```
fit <- lm(rhine~time(rhine), na.action = NULL)
summary(fit)
```

```
##
## Call:
## lm(formula = rhine ~ time(rhine), na.action = NULL)
##
## 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)  5.968486   0.127177   46.93   <2e-16 ***
## time(rhine) -0.017796   0.001305  -13.63   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

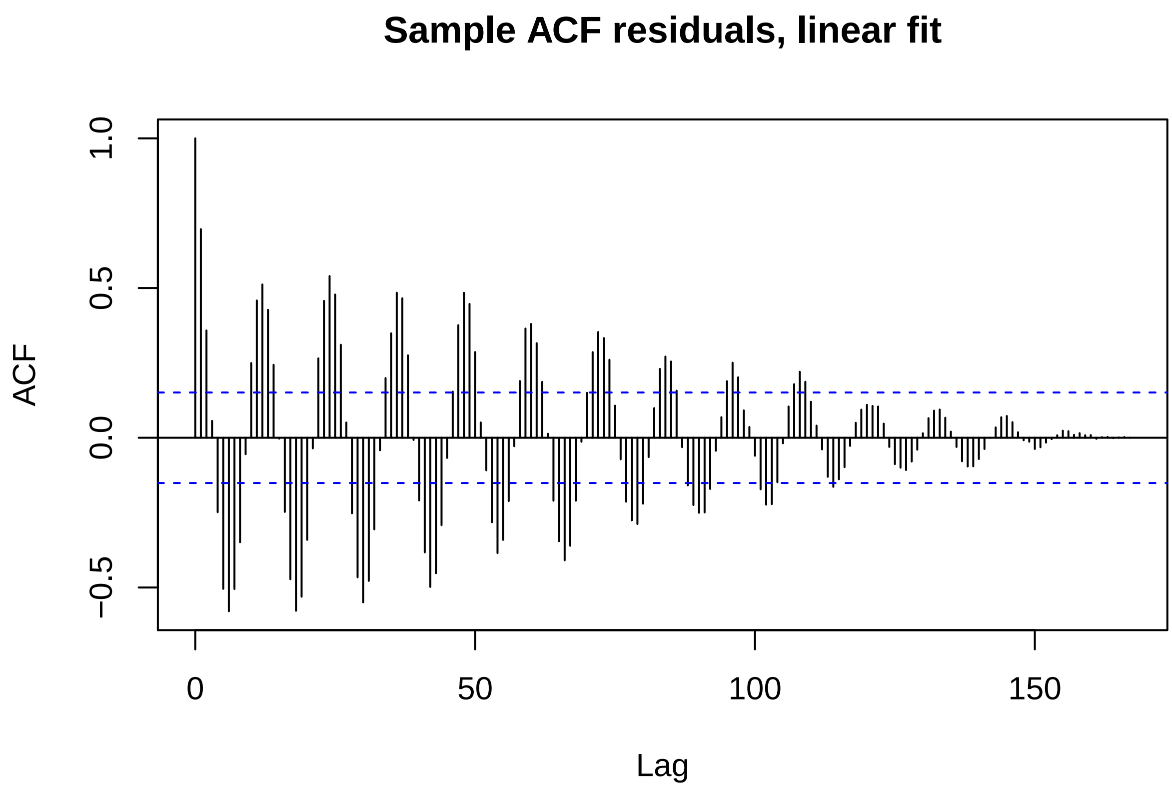
```
##
## Residual standard error: 0.8205 on 166 degrees of freedom
## Multiple R-squared:  0.5282, Adjusted R-squared:  0.5254
## F-statistic: 185.9 on 1 and 166 DF,  p-value: < 2.2e-16
```

By reading the significant negative coefficient with respect to time(rhine), one can see that there is a significant linear time trend.

```r
par(mfrow=c(2,1))
plot.ts(rhine, main="Nitrogen concentration Rhine")
plot(resid(fit), type="l", main="detrended")
```



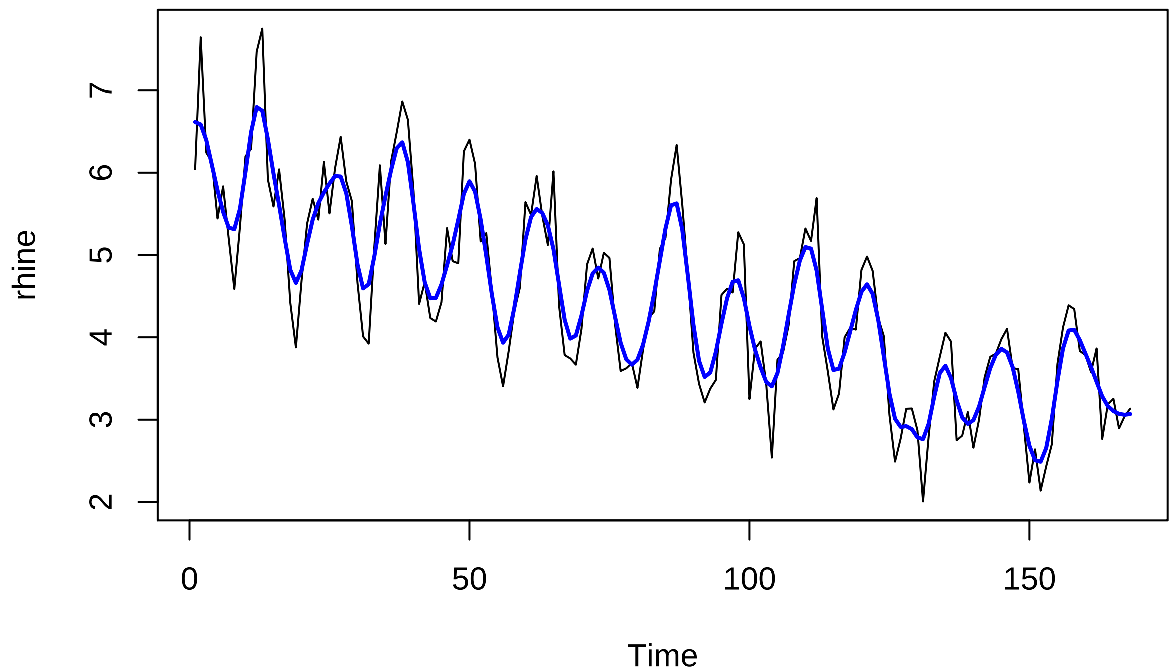

```r
#par(mfrow=c(2,1)) # plot ACFs
#acf(rhine, main="rhine", lag.max = 168)
acf(resid(fit), main="Sample ACF residuals, linear fit", lag.max = 168)
```



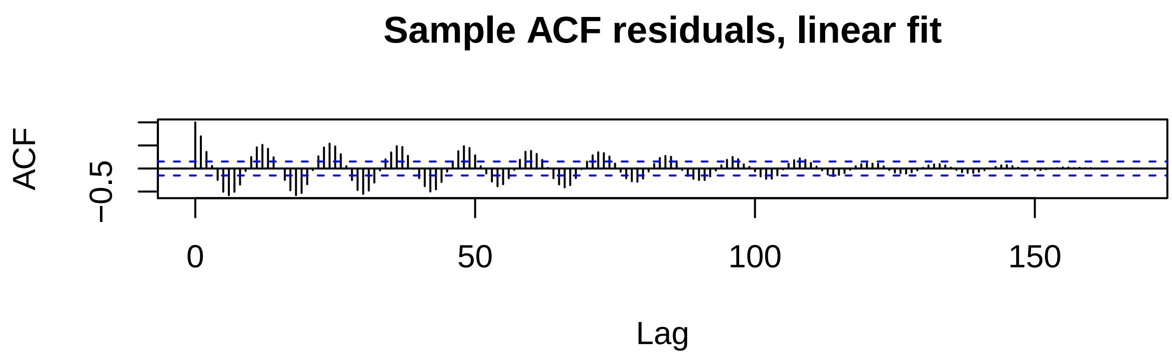
From the autocorrelation, one can see that it is positively correlated and negatively correlated with respect to certain time lags. Indicating seasonality.

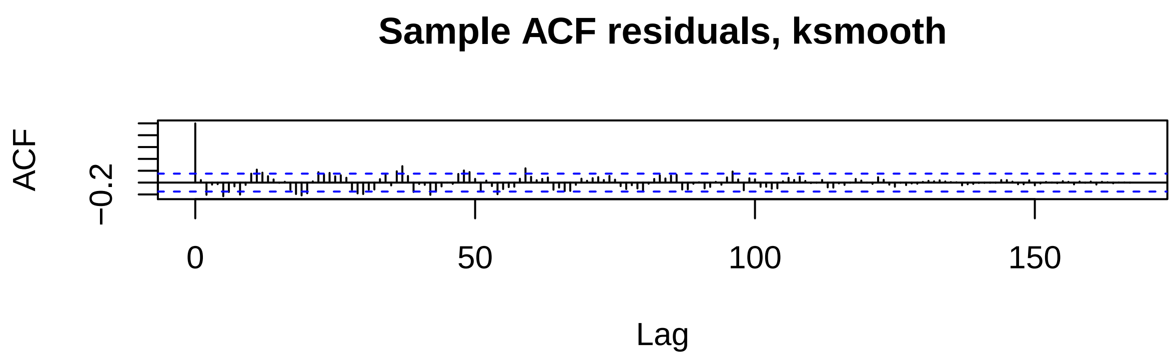
c

```
plot(rhine)
ksmooth <- ksmooth(time(rhine), rhine, "normal", bandwidth=4)
lines(ksmooth, lwd=2, col=4)
```




```r
par(mfrow=c(2,1))
acf(resid(fit), main="Sample ACF residuals, linear fit", lag.max = 168)
acf(rhine - ksmooth$y, main="Sample ACF residuals, ksmooth", lag.max = 168)
```





The ksmooth model seems to be a better model to elimate the time trend. Most autocorrelation lags are within the blue lines. Stationary?

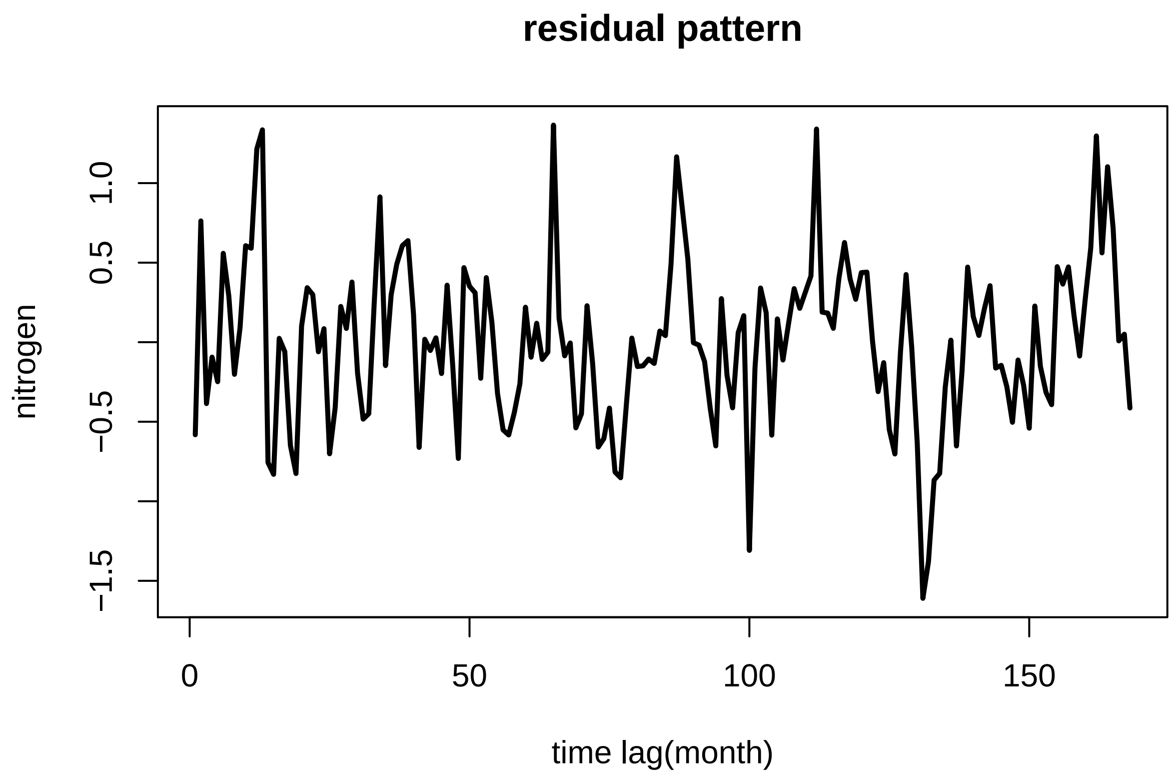
d

```
# Seasonal means model:
rhine_data$Month <- as.factor(rhine_data$Month)

# White noise is residual?
smm <- lm(rhine_data$TotN_conc ~  time(rhine_data$TotN_conc) + rhine_data$Month)
summary(smm)
```

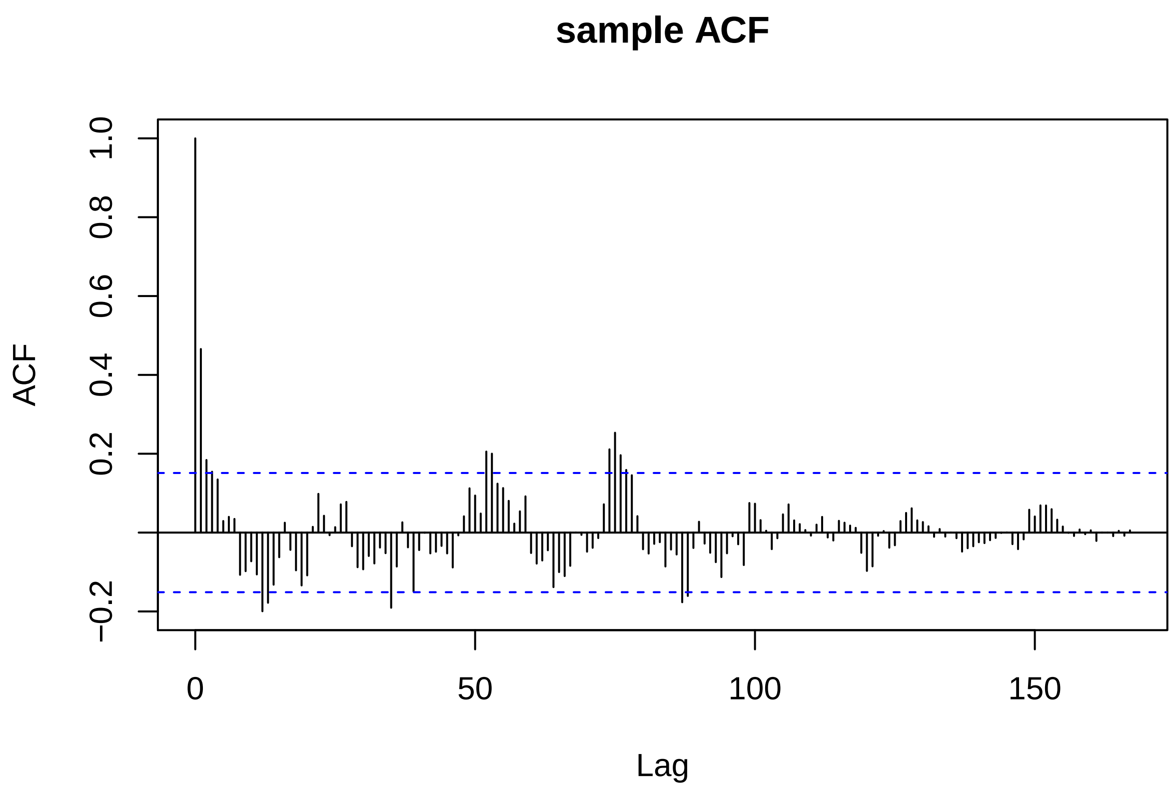
```
##
## Call:
## lm(formula = rhine_data$TotN_conc ~ time(rhine_data$TotN_conc) +
##     rhine_data$Month)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6101 -0.3119  0.0082  0.3115  1.3646
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 6.6405394  0.1560548  42.553  < 2e-16 ***
## time(rhine_data$TotN_conc) -0.0173536  0.0008424 -20.601  < 2e-16 ***
```

```
## rhine_data$Month2          0.2765862  0.1996248   1.386  0.16788
## rhine_data$Month3          0.0400625  0.1996301   0.201  0.84121
## rhine_data$Month4         -0.3464274  0.1996390  -1.735  0.08468 .
## rhine_data$Month5         -0.8616526  0.1996514  -4.316 2.82e-05 ***
## rhine_data$Month6         -1.2611407  0.1996674  -6.316 2.70e-09 ***
## rhine_data$Month7         -1.6080759  0.1996870  -8.053 2.00e-13 ***
## rhine_data$Month8         -1.7124154  0.1997101  -8.575 9.56e-15 ***
## rhine_data$Month9         -1.2366876  0.1997367  -6.192 5.11e-09 ***
## rhine_data$Month10        -0.8744578  0.1997669  -4.377 2.20e-05 ***
## rhine_data$Month11        -0.7512727  0.1998007  -3.760  0.00024 ***
## rhine_data$Month12        -0.1774468  0.1998379  -0.888  0.37594
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5282 on 155 degrees of freedom
## Multiple R-squared:  0.8175, Adjusted R-squared:  0.8034
## F-statistic: 57.85 on 12 and 155 DF,  p-value: < 2.2e-16
```

```r
plot.ts(residuals(smm), main="residual pattern",
     ylab="nitrogen", xlab="time lag(month)", lwd=2.5)
```




```r
acf(residuals(smm), lag.max=168, main="sample ACF")
```



e

```r
step_smm <- step(smm, direction = "both", trace = TRUE)
```

```
## Start:  AIC=-202.02
## rhine_data$TotN_conc ~ time(rhine_data$TotN_conc) + rhine_data$Month
##
##                               Df Sum of Sq     RSS      AIC
## <none>                                      43.237 -202.023
## - rhine_data$Month            11    68.524 111.761  -64.477
## - time(rhine_data$TotN_conc)   1   118.387 161.624   17.499
```

```r
summary(step_smm)
```

```
##
## Call:
## lm(formula = rhine_data$TotN_conc ~ time(rhine_data$TotN_conc) +
##     rhine_data$Month)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6101 -0.3119  0.0082  0.3115  1.3646
##
```

```
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 6.6405394  0.1560548  42.553  < 2e-16 ***
## time(rhine_data$TotN_conc) -0.0173536  0.0008424 -20.601  < 2e-16 ***
## rhine_data$Month2           0.2765862  0.1996248   1.386  0.16788
## rhine_data$Month3           0.0400625  0.1996301   0.201  0.84121
## rhine_data$Month4          -0.3464274  0.1996390  -1.735  0.08468 .
## rhine_data$Month5          -0.8616526  0.1996514  -4.316 2.82e-05 ***
## rhine_data$Month6          -1.2611407  0.1996674  -6.316 2.70e-09 ***
## rhine_data$Month7          -1.6080759  0.1996870  -8.053 2.00e-13 ***
## rhine_data$Month8          -1.7124154  0.1997101  -8.575 9.56e-15 ***
## rhine_data$Month9          -1.2366876  0.1997367  -6.192 5.11e-09 ***
## rhine_data$Month10         -0.8744578  0.1997669  -4.377 2.20e-05 ***
## rhine_data$Month11         -0.7512727  0.1998007  -3.760  0.00024 ***
## rhine_data$Month12         -0.1774468  0.1998379  -0.888  0.37594
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5282 on 155 degrees of freedom
## Multiple R-squared:  0.8175, Adjusted R-squared:  0.8034
## F-statistic: 57.85 on 12 and 155 DF,  p-value: < 2.2e-16
```

```r
# using MASS package
library(MASS)
stepwise_smm <- stepAIC(smm, direction = "both", trace = FALSE)
summary(stepwise_smm)
```

```
##
## Call:
## lm(formula = rhine_data$TotN_conc ~ time(rhine_data$TotN_conc) +
##     rhine_data$Month)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6101 -0.3119  0.0082  0.3115  1.3646
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 6.6405394  0.1560548  42.553  < 2e-16 ***
## time(rhine_data$TotN_conc) -0.0173536  0.0008424 -20.601  < 2e-16 ***
## rhine_data$Month2           0.2765862  0.1996248   1.386  0.16788
## rhine_data$Month3           0.0400625  0.1996301   0.201  0.84121
## rhine_data$Month4          -0.3464274  0.1996390  -1.735  0.08468 .
## rhine_data$Month5          -0.8616526  0.1996514  -4.316 2.82e-05 ***
## rhine_data$Month6          -1.2611407  0.1996674  -6.316 2.70e-09 ***
## rhine_data$Month7          -1.6080759  0.1996870  -8.053 2.00e-13 ***
## rhine_data$Month8          -1.7124154  0.1997101  -8.575 9.56e-15 ***
## rhine_data$Month9          -1.2366876  0.1997367  -6.192 5.11e-09 ***
## rhine_data$Month10         -0.8744578  0.1997669  -4.377 2.20e-05 ***
## rhine_data$Month11         -0.7512727  0.1998007  -3.760  0.00024 ***
## rhine_data$Month12         -0.1774468  0.1998379  -0.888  0.37594
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

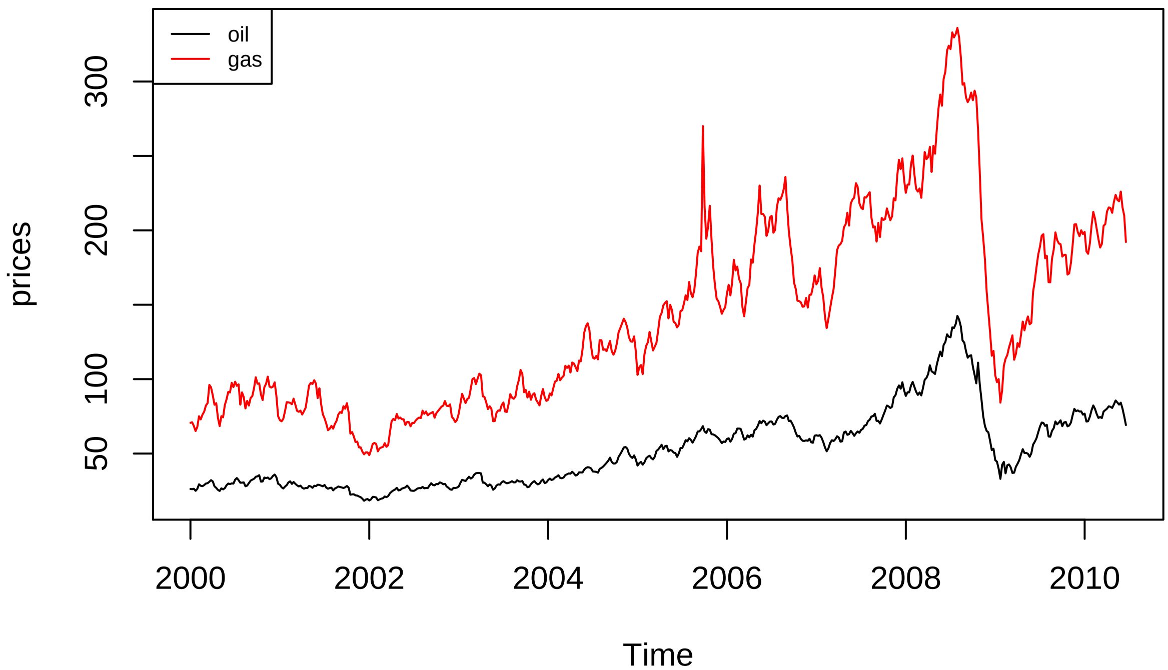
```
## Residual standard error: 0.5282 on 155 degrees of freedom
## Multiple R-squared:  0.8175, Adjusted R-squared:  0.8034
## F-statistic: 57.85 on 12 and 155 DF,  p-value: < 2.2e-16
```

All time lags are left in the model.

Assignment 3

a

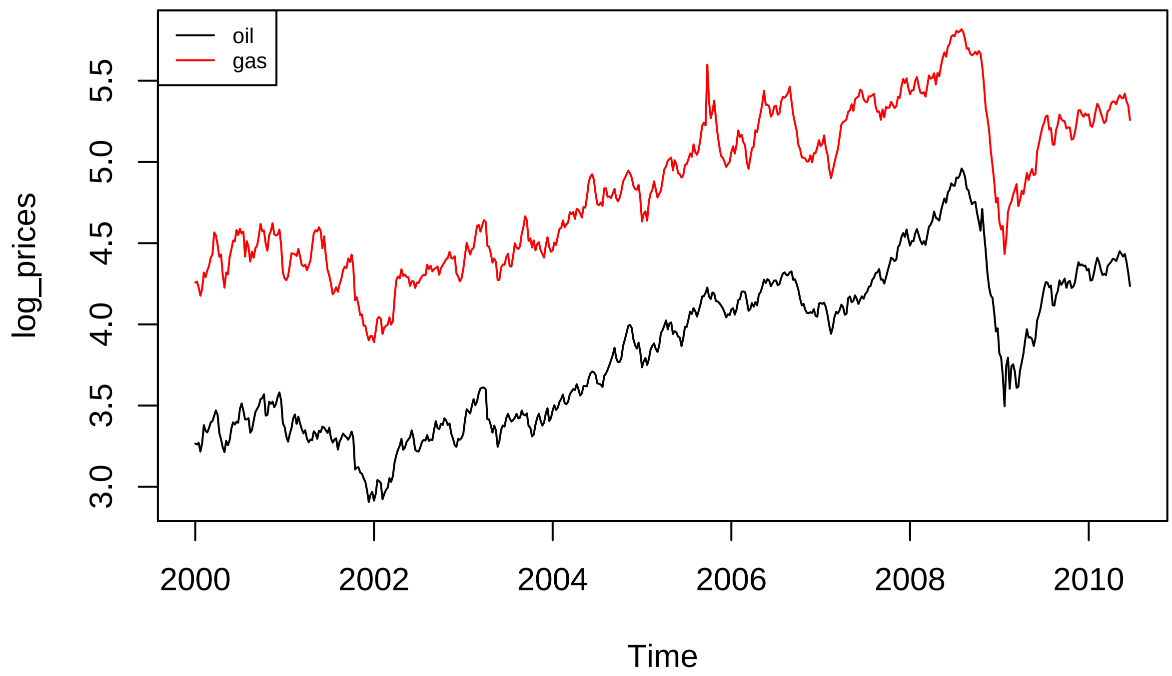
```r
prices <- cbind(oil, gas)

plot(prices, plot.type="single", col = 1:ncol(prices))
legend("topleft", colnames(prices), col=1:ncol(prices), lty=1, cex=.65)
```



The processes do not look stationary. Even if you think out the linear trend, e.g. variance around 2009 seems much higher than around 2002. The processes seem to be related to each other, as the movements of the timeseries are alike, just on a different scale. Oil prices is lower than gas prices.

b

```r
log_prices <- log(prices)

plot(log_prices, plot.type="single", col = 1:ncol(log_prices))
legend("topleft", colnames(log_prices), col=1:ncol(log_prices), lty=1, cex=.65)
```

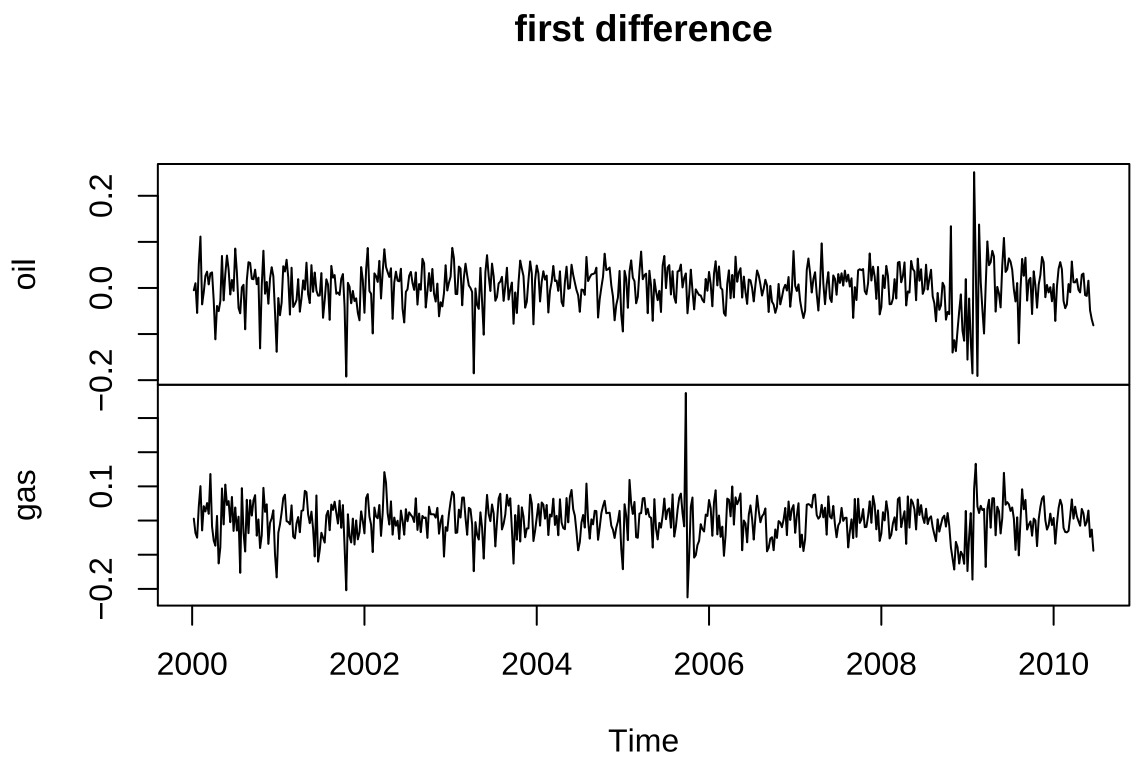


Taking the logarithm of the data seems to enable one to compare the time series better as they are now both on a similar scale.

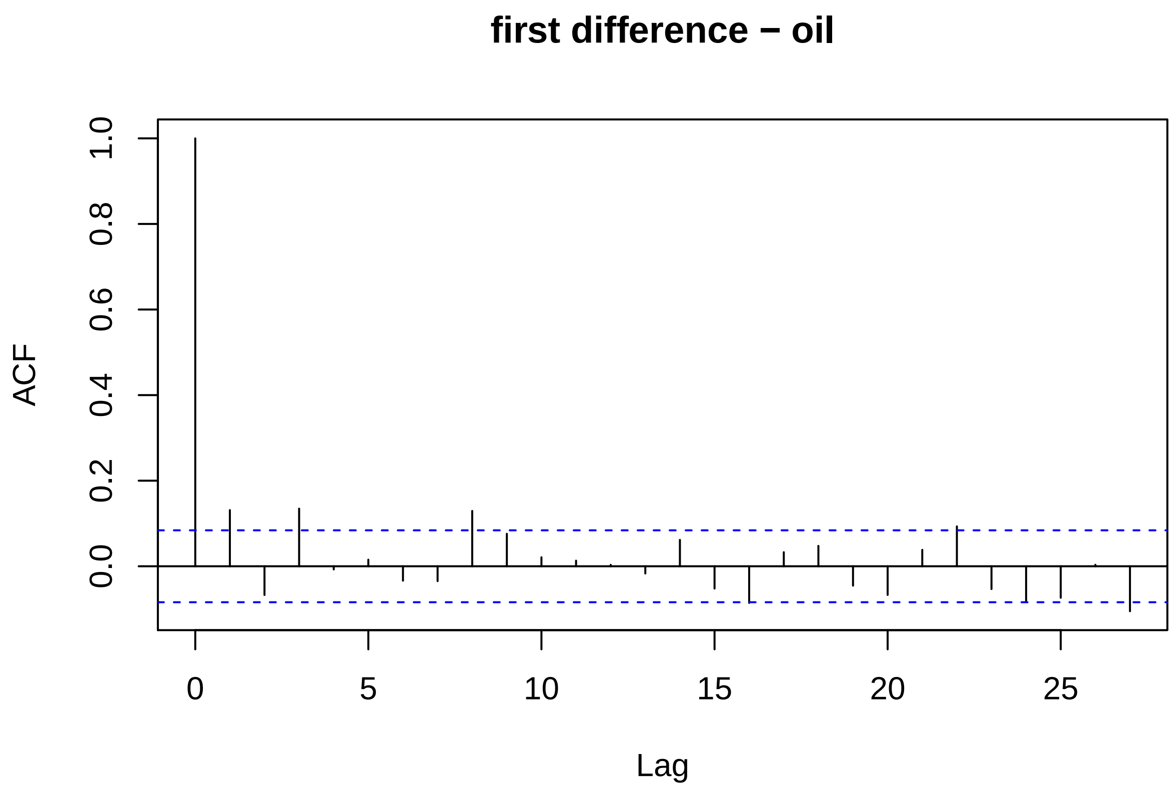
c

```r
diff_log_prices <- as.data.frame(diff(log_prices))
```

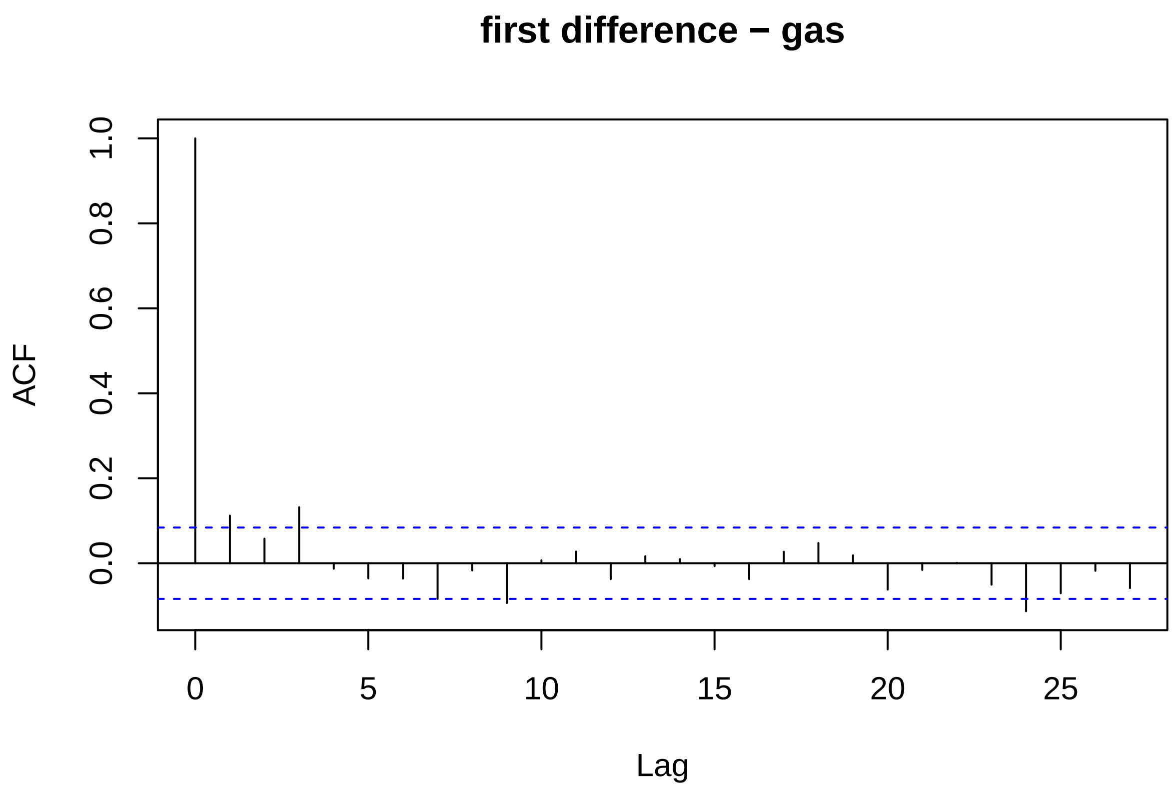
```r
plot(diff(log_prices), type="l", main="first difference")
```




```r
log_prices_df <- as.data.frame(log_prices)
acf(diff(log_prices_df$oil), main="first difference - oil")
```




```
acf(diff(log_prices_df$gas), main="first difference - gas")
```

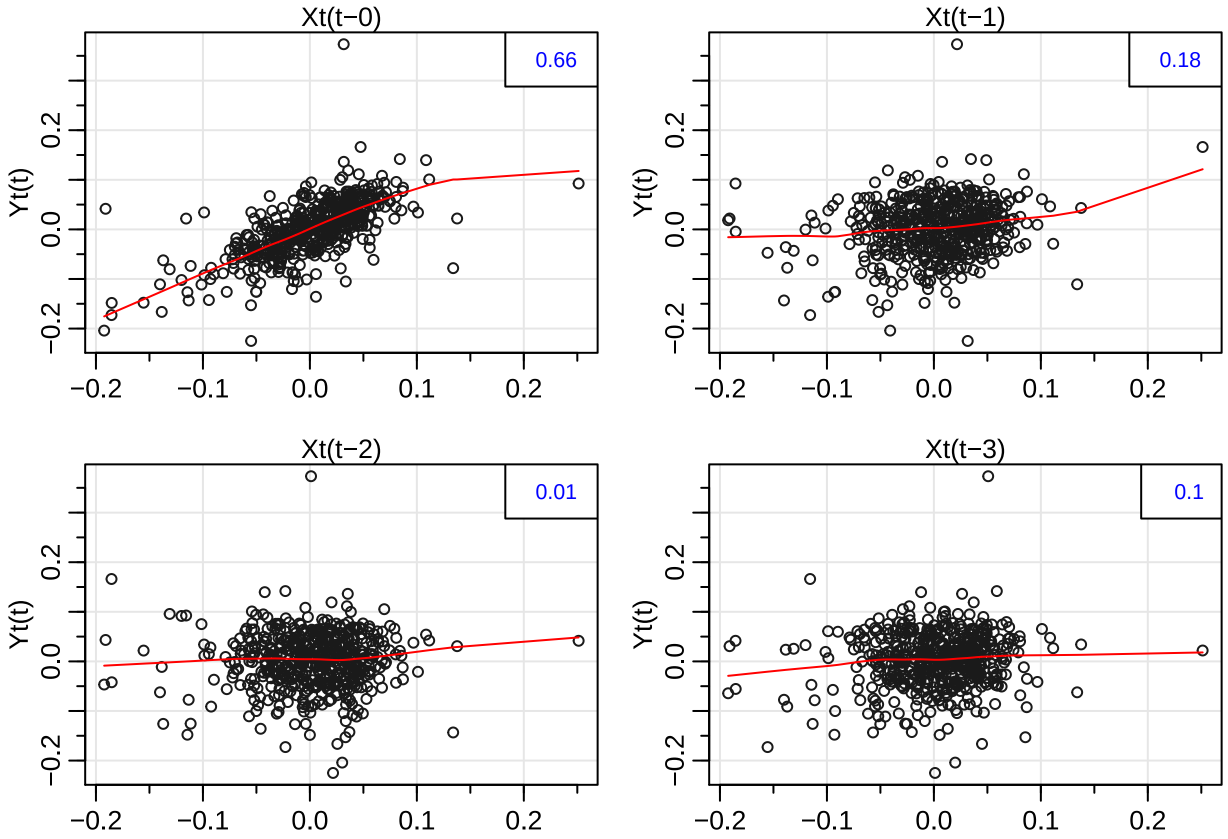


There does not seem to be much autocorrelation, however in general I would say there is a bit more autocorrelation in oil than in gas.

```
Xt <- ts(diff_log_prices[,1])
Yt <- ts(diff_log_prices[,2])
```

d

```
lag2.plot(Xt, Yt, max.lag = 3, smooth = TRUE)
```



There is a linear relationship between Yt and X(t-0). Thus a correlation between Yt and Xt, Yt is not correlated with the lags of Xt.

e

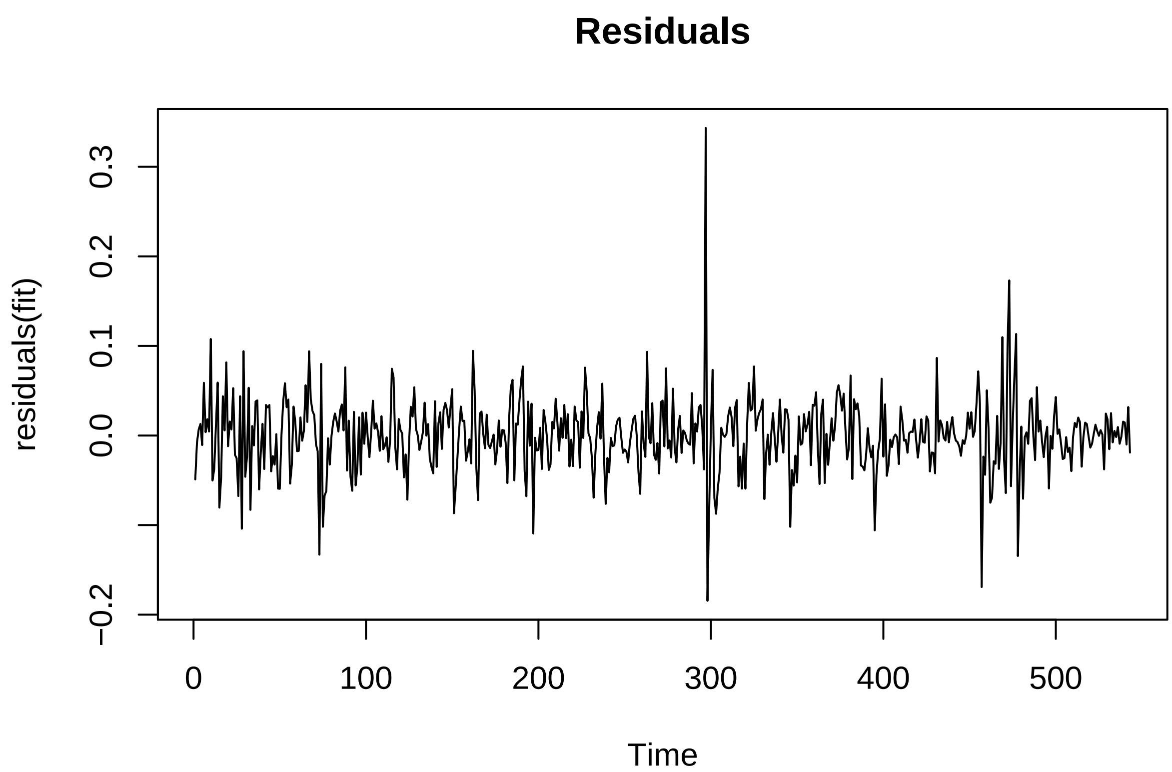
```
I <- ifelse(Xt < 0, 0, 1)
data <- ts.intersect(Yt, I, Xt, dXt=lag(Xt, -1))
fit <- lm(Yt ~ I + Xt + dXt, data = data)
summary(fit)
```

```
##
## Call:
## lm(formula = Yt ~ I + Xt + dXt, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.18451 -0.02161 -0.00038  0.02176  0.34342
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.006445   0.003464  -1.860  0.06338 .
## I            0.012368   0.005516   2.242  0.02534 *
## Xt           0.683127   0.058369  11.704  < 2e-16 ***
## dXt          0.111927   0.038554   2.903  0.00385 **
```

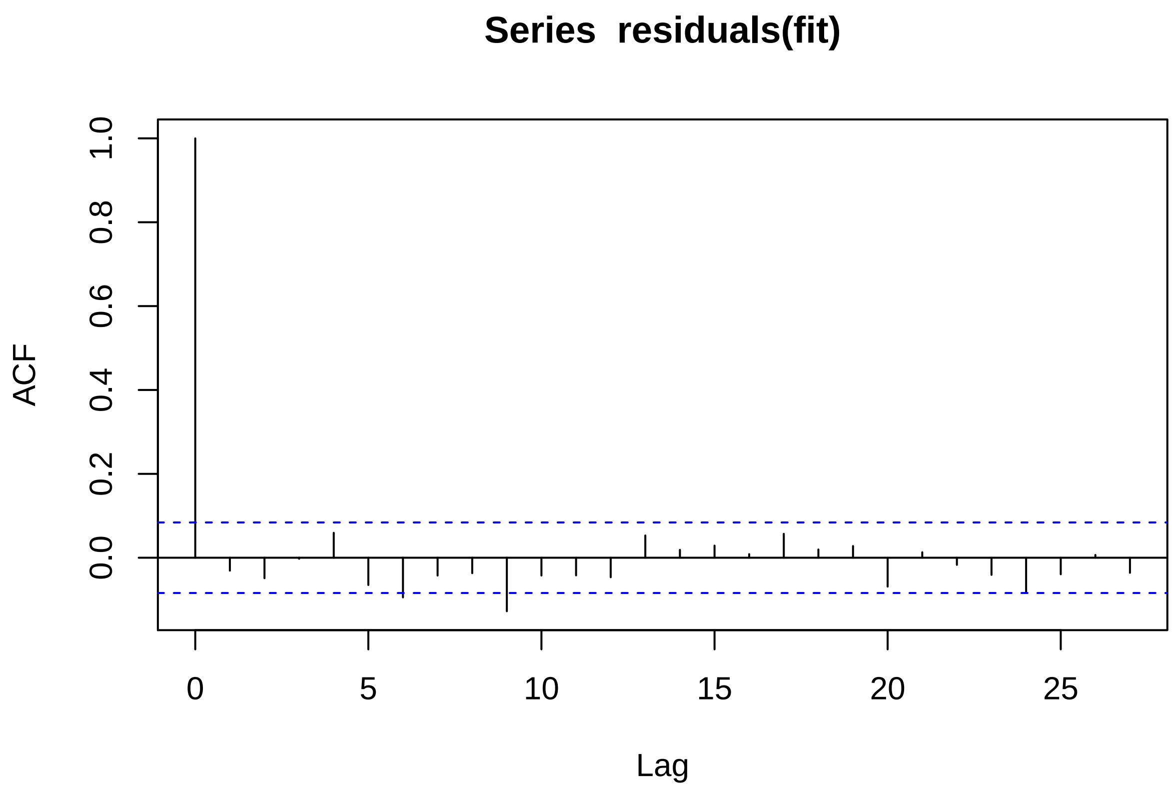
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04169 on 539 degrees of freedom
## Multiple R-squared:  0.4563, Adjusted R-squared:  0.4532
## F-statistic: 150.8 on 3 and 539 DF,  p-value: < 2.2e-16
```

All three variables show significant coefficients. Meaning they all have a positve effect on Yt.

```r
plot.ts(residuals(fit), main="Residuals")
```




```r
acf(residuals(fit))
```



Residuals seem rather stationary? Except for two spikes. There does not seem to be much autocorrelation.