TSA_LAB01(Group08)

Thijs Quast(thiqu264), Saewon Jun(saeju204)

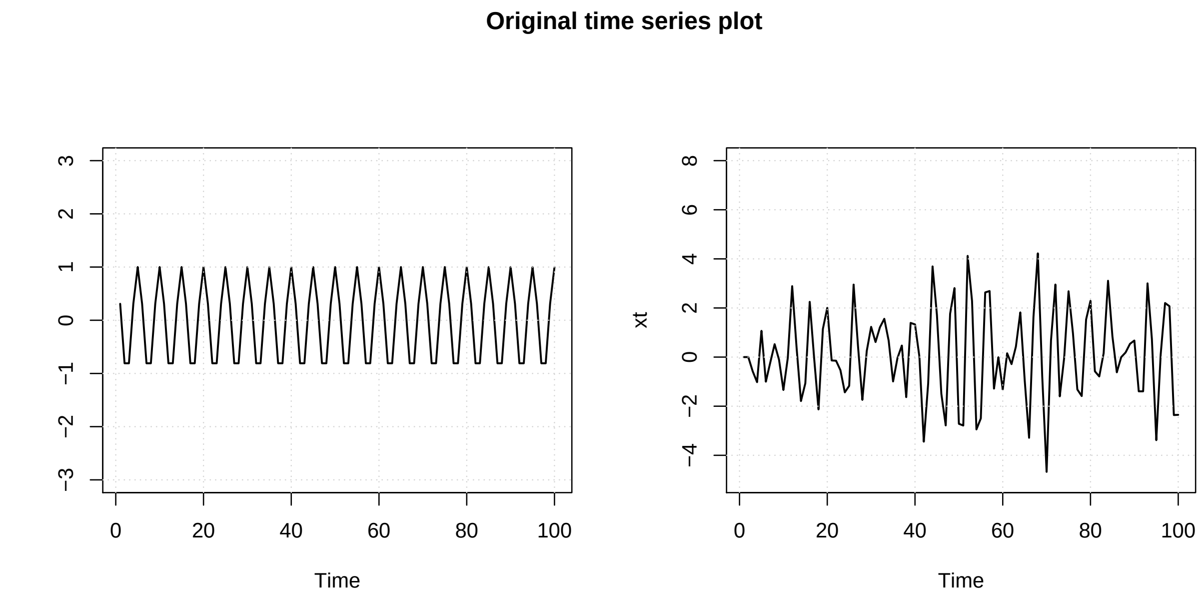
2019 9 15

Contents

#Assignment1. Computations with simulated data ##a) Generate time series and apply smoothing filter. Generate two time series $x_t = -0.8x_{t-2} + w_t$, where $x_0 = x_1 = 0$ and $x_t = cos(\frac{2pit}{5})$ with 100 observations each. Apply smoothing filter $v_t = 0.2(x_t + x_{t-1} + x_{t-2} + x_{t-3} + x_{t-4})$ to these two series and compare how the filter has affected them.

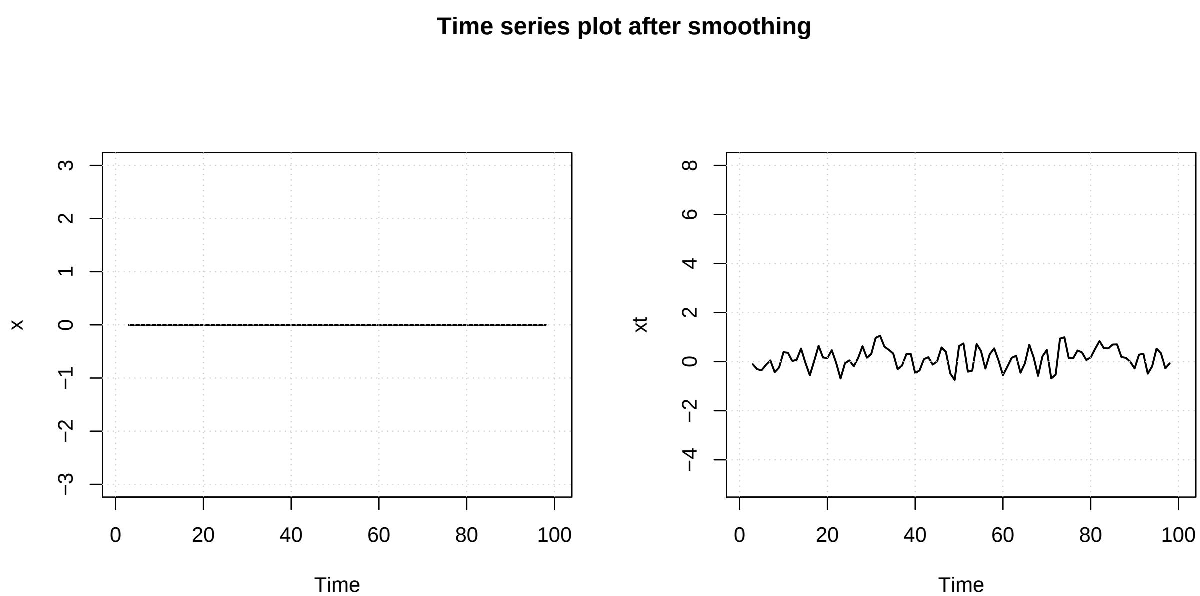
###Before applying smoothing filter

```
library(graphics)
set.seed(12345)

##generate x_t ... x
x <- cos(2*pi*(1:100)/5)
#x[1:2] <- 0

##generate time series x_t

w_t <- rnorm(100) #WN....normal distribution?
#x_t <- filter(x, c(0,-0.8), method="recursive") + w_t
x_t <- filter(w_t, c(0,-0.8), method="recursive")
x_t[1:2] <- 0

par(mfrow=c(1,2), oma=c(0,0,2,0))
plot.ts(x, ylab="", lwd=1.5, ylim=c(-3,3))
grid()
plot.ts(x_t, ylab="xt", lwd=1.5, ylim=c(-5,8))
grid()
title(main="Original time series plot", outer=TRUE)
```



###After applying smoothing filter

```r
smooth_x <- filter(x, rep(0.2,5), method="convolution")
smooth_xt <- filter(x_t, rep(0.2,5), method="convolution")

par(mfrow=c(1,2), oma=c(0,0,2,0))
plot.ts(smooth_x, ylab="x",lwd=1.5, ylim=c(-3,3))
grid()
plot.ts(smooth_xt, ylab="xt", lwd=1.5, ylim=c(-5,8))
grid()
title(main="Time series plot after smoothing", outer=TRUE)
```



It seems like there is not significant change for first time series after applying smoothing filter v_t. However, for the second time series, we can see that the trend slightly going up after applying smoothing filter.

##b) Casuality and invertibility of time series investigate whether the following time series is casual and invertible.

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

Given ARMA model can be also written as :

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

here we can use *polyroot*() function to check whether the roots are outside the unit circle. 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.

###polynomial root of AR $\phi(B)$

```
#polyroot(c(1,-4,2,0,0,1))

abs(polyroot(c(1,-4,2,0,0,1)))
```

```
## [1] 0.2936658 1.6793817 1.0000000 1.4239626 1.4239626
```

It is not causal (2 root inside the unit circle)

###polynomial root of MA $\theta(B)$

```
#polyroot(c(1,0,3,0,1,0,-4))
abs(polyroot(c(1,0,3,0,1,0,-4)))
```

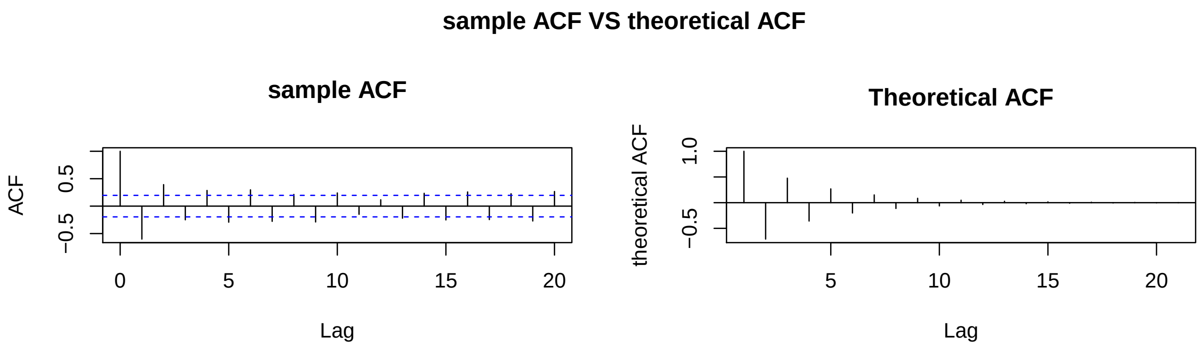
```
## [1] 0.6874372 0.6874372 0.6874372 0.6874372 1.0580446 1.0580446
```

It is not invertible (4 root inside the unit circle)

##c) Compute sample ACF, theorerical ACF, and compare the plot. Simulate the 100 observations from the process(seed:54321):

$$x_t + \frac{3}{4}x_{t-1} = w_t - \frac{1}{9}w_{t-2}$$

```
set.seed(54321)

##we can use arima.sim() to simulate from an ARIMA(1,0,2) model.
arima_sim <- arima.sim(n=100, list(order=c(1,0,2), ar=c(-3/4), ma=c(0,-1/9)))



##acf() for sample ACF
par(mfrow=c(1,2), oma=c(0,0,2,0))
acf(arima_sim, type="correlation", main="sample ACF")

##ARMAacf() for theoretical ACF
t_acf <- ARMAacf(ar=c(-3/4), ma=c(0,-1/9), lag.max=20) #lag.max ...?
plot(t_acf, xlab="Lag", ylab="theoretical ACF", type="h", main="Theoretical ACF")
abline(h=0)
title(main="sample ACF VS theoretical ACF", outer=TRUE)
```

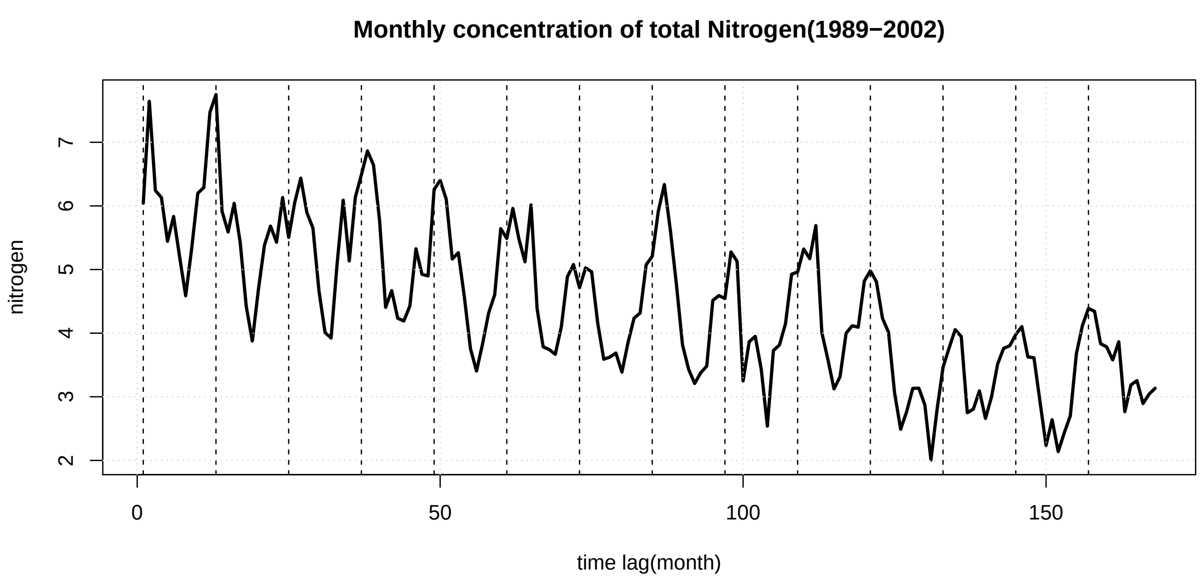


Both sample ACF and theoretical ACF shows similar patterns of fluctuation. However The theoretical autocorrelation seems to be more desirable, as it diminishes to zero after approximately 15 lags, whereas the sample autocorrelatoin seems to be agove the blue lines, still at lag 20.

#Assignment2. Visualization, detrending and residual analysis The data set *Rhine.csv* contains monthly concentrations of total nitrogen in the Rhine river in the period 1989-2002

##a) Explore the data ###Convert the data into ts object, and explore it by plotting the time series,

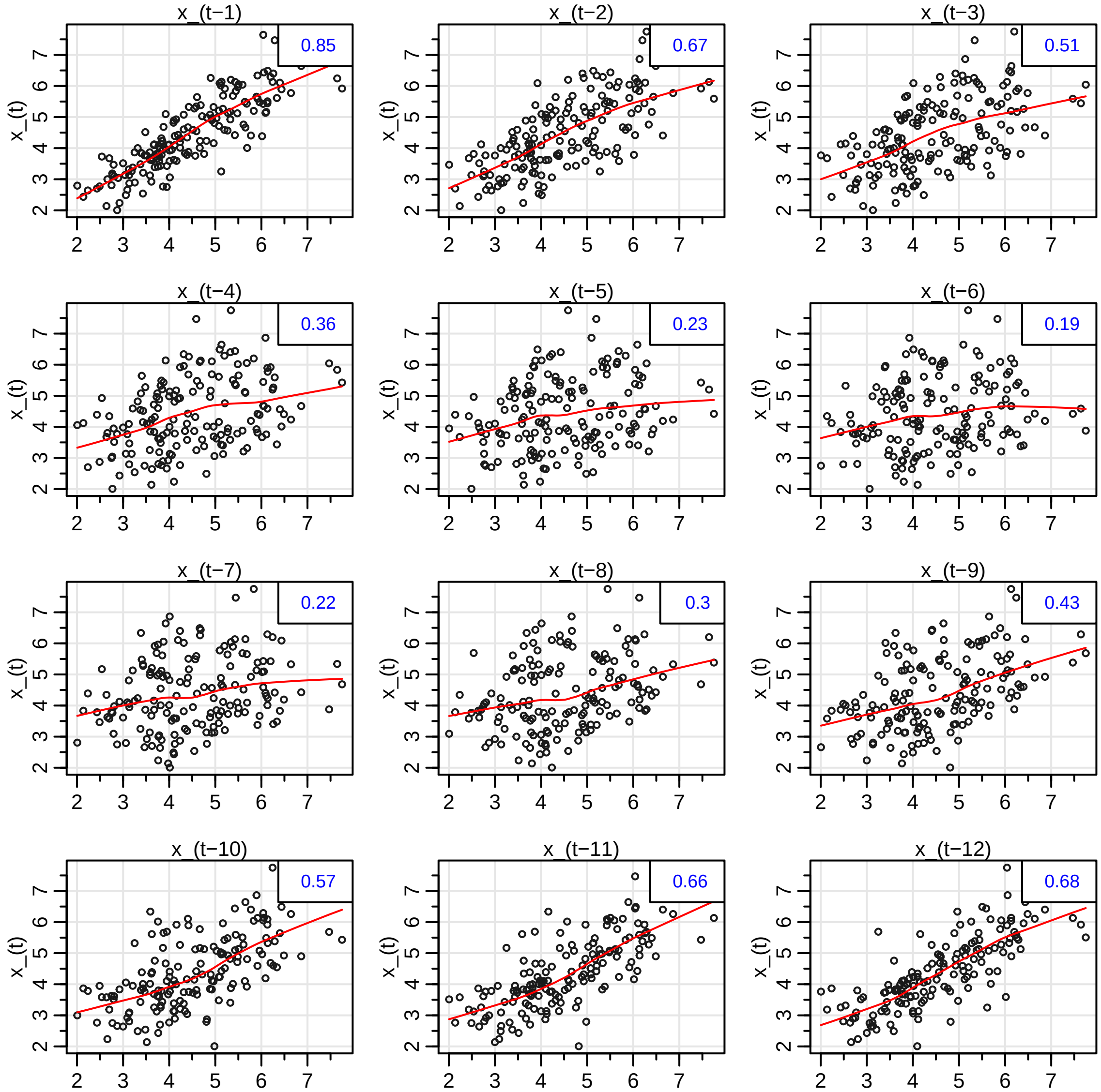
```
data <- read.csv2("data/Rhine.csv")
ts_data <- ts(data)

plot.ts(ts_data[,4], main="Monthly concentration of total Nitrogen(1989-2002)",
        ylab="nitrogen", xlab="time lag(month)", lwd=2.5)
abline(v=seq(1,168,12),  lty=2)
grid()
```



Are there any trends, linear or seasonal in the time series? Vertical line devides the time series into 12 months each. General fluctutation trend seems to be repeated every year, and we can also see there's down going trend for entire data set as well.
When during the year is the concentration highest?
Generally very beginning of the year and end of the year shows significantly high concentration level.

###Scatter plots of x_t against $x_{t-1}, ..., x_{t-12}$

```r
library(astsa)

x_ <- ts_data[,4]
lag1.plot(x_, max.lag=12, corr=T, smooth=T)
```



Are there any special patterns in the data or scatterplots? Does the variance seem to change over time? Which variables in the scatterplots seem to have a significant relation to each other? From the scatter plot above, we can see that the correlation between two time lag tends to decrease as the lag extends and increase again. We can relate this to the fluctutaion pattern of time series plot. We have mentioned that the general fluctuation pattern repeats each year(12 month), which means the time points with distance of 12 lag should share the similar information.

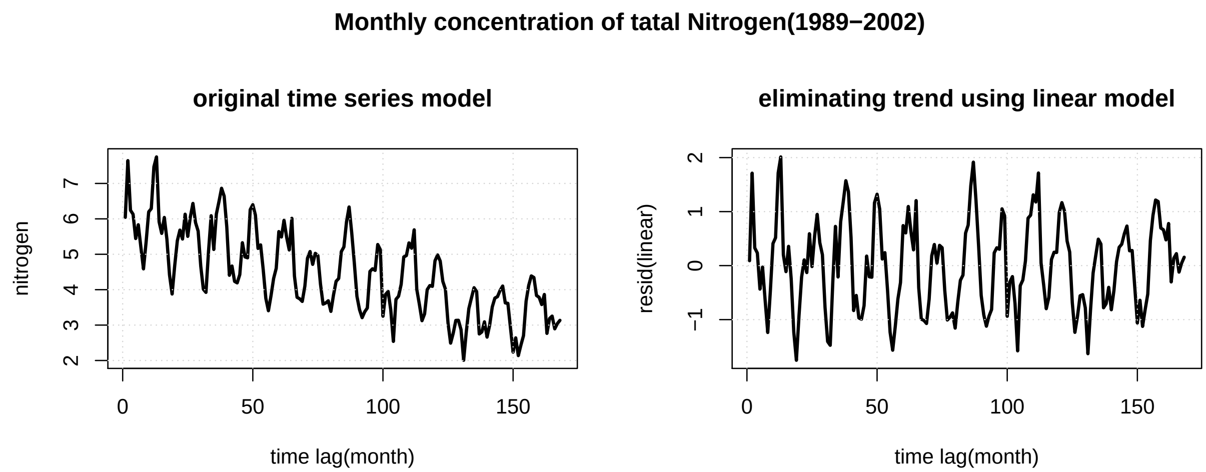
##b) Eliminate the trend - by fitting a linear model w.r.t t. Is there any significant time trend? (Look at the residual pattern and the sample ACF of the residuals and see how this pattern might be related to seasonality of the series.)

###Fitting a linear model

```r
#fit a linear model
linear <- lm(ts_data[,4] ~ time(ts_data[,4]))
summary(linear)
```

```
##
## Call:
## lm(formula = ts_data[, 4] ~ time(ts_data[, 4]))
##
## 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(ts_data[, 4]) -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
```

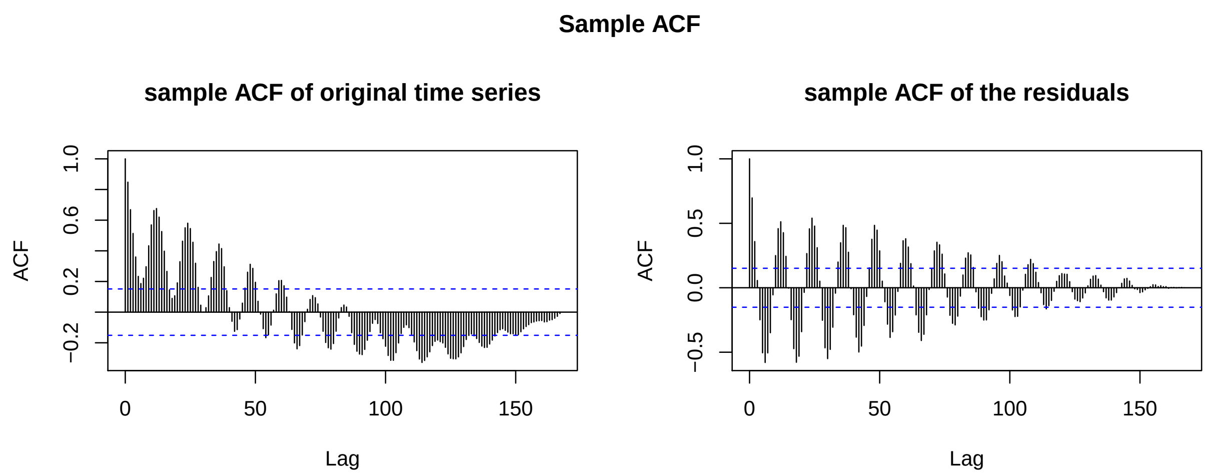
```r
#compare the plot
par(mfrow=c(1,2), oma=c(0,0,2,0))
plot.ts(ts_data[,4], main="original time series model",
        ylab="nitrogen", xlab="time lag(month)", lwd=2.5)
grid()
plot.ts(resid(linear), main="eliminating trend using linear model",
        xlab="time lag(month)", lwd=2.5)
grid()
title(main="Monthly concentration of tatal Nitrogen(1989-2002)", outer=TRUE)
```



There exists trend in the original time series model which is slightly going down. This trend seems to be eliminated after fitting linear model (in order to detrend the model).

###Compare sample ACF from original time series

```r
par(mfrow=c(1,2), oma=c(0,0,2,0))
acf(ts_data[,4], type="correlation", lag.max=168,
    main="sample ACF of original time series")
acf(resid(linear), lag.max=168, main="sample ACF of the residuals" )
title(main="Sample ACF", outer=TRUE)
```

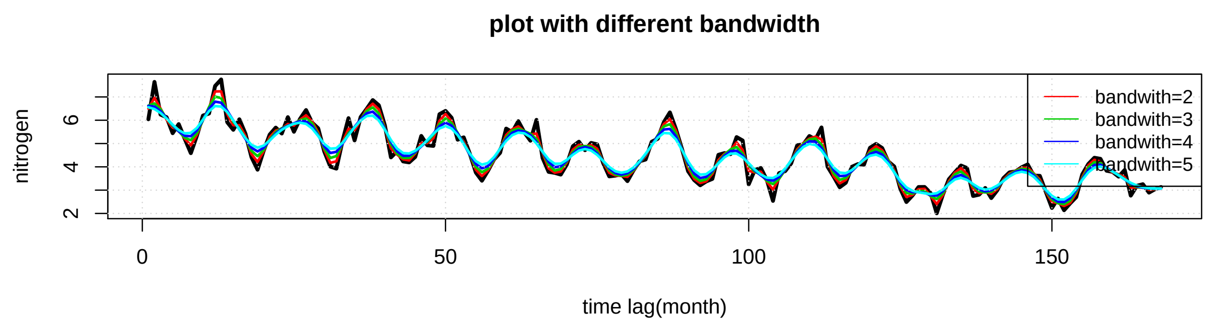


The ACF plot diminishes toward 0 faster after detrending using linear model. However, we can still see the pattern which means still ACF is dependent on time lag.

##c) Eliminate the trend by fitting a kernel smoother w.r.t to t. Analyze the residual pattern and the sample ACF of the residuals. Then compare it to the ACF from previous step. Do residuals seem to represent a stationary series?(Choose a reasonable bandwidth yourself so the fit looks reasonable)

###Choosing a reasonable bandwith

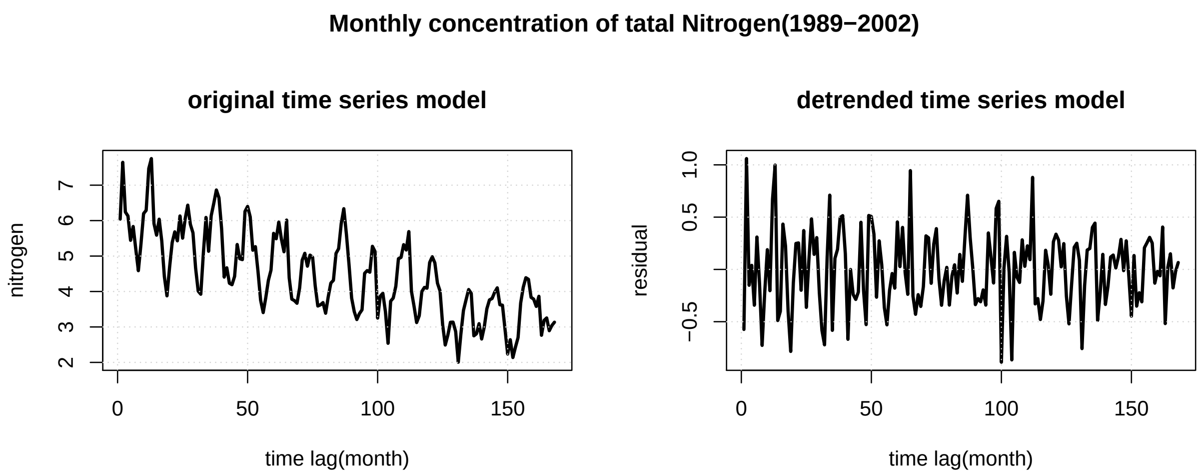
```r
#fit a kernel model- choose reasonable bandwidth
plot.ts(ts_data[,4], main="plot with different bandwidth",
        ylab="nitrogen", xlab="time lag(month)", lwd=3)
lines(ksmooth(time(ts_data[,4]), data[,4], "normal", bandwidth = 2), col = 2, lwd=2)
lines(ksmooth(time(ts_data[,4]), data[,4], "normal", bandwidth = 3), col = 3, lwd=2)
lines(ksmooth(time(ts_data[,4]), data[,4], "normal", bandwidth = 4), col = 4, lwd=2)
lines(ksmooth(time(ts_data[,4]), data[,4], "normal", bandwidth = 5), col = 5, lwd=2)
grid()
legend("topright",legend=c("bandwith=2","bandwith=3","bandwith=4","bandwith=5"),
       col=c(2,3,4,5),lty=c(1,1), cex=0.9)
```



bandwith value with 4 seems reasonable.

###Fitting a kernel smoother

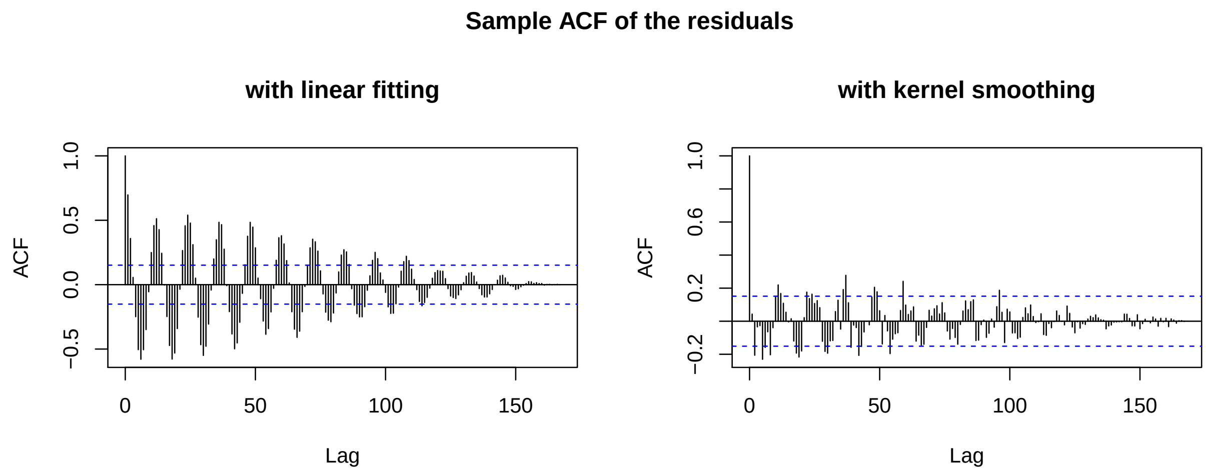
```
kernel <- ksmooth(time(ts_data[,4]), data[,4], "normal", bandwidth = 4)

#compare the plot
par(mfrow=c(1,2), oma=c(0,0,2,0))
plot.ts(ts_data[,4], main="original time series model",
        ylab="nitrogen", xlab="time lag(month)", lwd=2.5)
grid()
plot.ts(ts_data[,4]-kernel$y, main="detrended time series model",
        xlab="time lag(month)", ylab="residual", lwd=2.5)
grid()
title(main="Monthly concentration of tatal Nitrogen(1989-2002)", outer=TRUE)
```



There exists trend in the original time series model which is slightly going down. This trend seems to be eliminated after applying kernel smoother(in order to detrend the model).

###Compare sample ACF from previous step

```
par(mfrow=c(1,2), oma=c(0,0,2,0))
acf(resid(linear), lag.max=168, main="with linear fitting")
acf(ts_data[,4]-kernel$y, lag.max=168, main="with kernel smoothing")
title(main="Sample ACF of the residuals", outer=TRUE)
```



The ACF function has significantly improved after applying smoothing filter. Most of the ACF falls between blue line without any significant pattern, diminishing towards 0. However weak seasonal pattern still exists. Considering all these, we can conclude that residuals are close to white noise which is stationary.

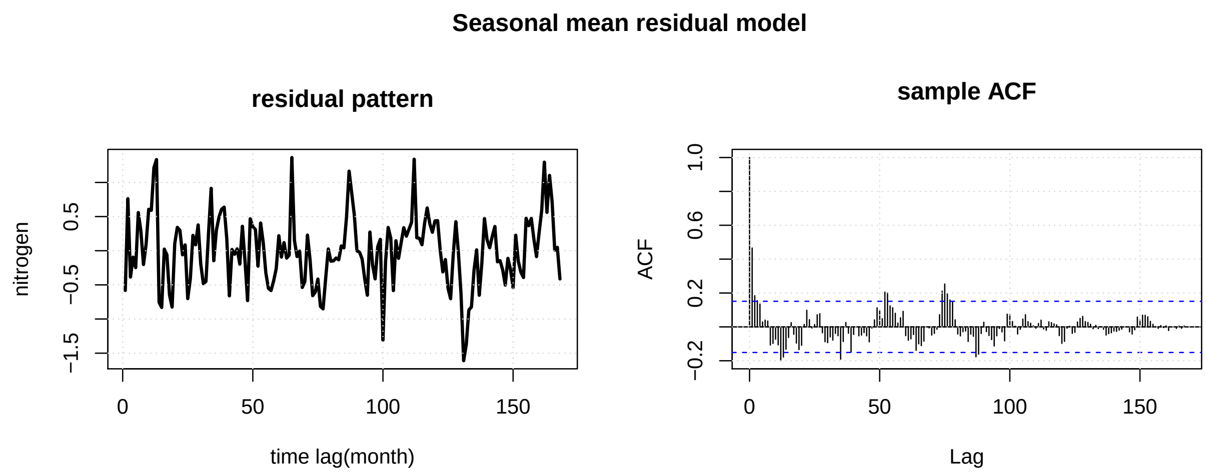
##d) Eliminate the trend by fitting seasonal mean model:

$$x_t = \alpha_0 + \alpha_1 t + \beta_1 I(month = 2) + \cdot + \beta_{12}(month = 12) + w_t$$

where $I(x) = 1$ if x is true and 0 otherwise. (fitting of this model will require you to augment data with a categorical variable showing the current month, and then fitting a usual linear regression.)

###Analyze the residual pattern and the ACF of residuals.

```
data[,2] <- as.factor(data[,2])
smm <- lm(ts_data[,4] ~ data[,2] + time(ts_data[,4]))

par(mfrow=c(1,2), oma=c(0,0,2,0))
plot.ts(residuals(smm), main="residual pattern",
     ylab="nitrogen", xlab="time lag(month)", lwd=2.5)
grid()
acf(residuals(smm), lag.max=168, main="sample ACF")
grid()
title(main="Seasonal mean residual model", outer=TRUE)
```



The residual pattern shows detrended model, and ACF has improved significantly. Most of ACF are inside blue line, and seasonal pattern has been removed.

##e) Perform stepwise variable selection in model from previous step. Which model gives you the lowest AIC? which varaibles are left in the model?

```r
###create dummy variable
mon2 <- as.numeric(data[,2]==2)
mon3 <- as.numeric(data[,2]==3)
mon4 <- as.numeric(data[,2]==4)
mon5 <- as.numeric(data[,2]==5)
mon6 <- as.numeric(data[,2]==6)
mon7 <- as.numeric(data[,2]==7)
mon8 <- as.numeric(data[,2]==8)
mon9 <- as.numeric(data[,2]==9)
mon10 <- as.numeric(data[,2]==10)
mon11 <- as.numeric(data[,2]==11)
mon12 <- as.numeric(data[,2]==12)

step(lm(ts_data[,4] ~ mon2+mon3+mon4+mon5+mon6+mon7+mon8+mon9
        +mon10+mon11+mon12+time(ts_data[,4])), direction="both")
```

```
## Start:  AIC=-202.02
## ts_data[, 4] ~ mon2 + mon3 + mon4 + mon5 + mon6 + mon7 + mon8 +
##     mon9 + mon10 + mon11 + mon12 + time(ts_data[, 4])
##
##                     Df Sum of Sq     RSS      AIC
## - mon3               1     0.011  43.248 -203.979
## - mon12              1     0.220  43.456 -203.170
## <none>                            43.237 -202.023
## - mon2               1     0.535  43.772 -201.955
## - mon4               1     0.840  44.076 -200.790
## - mon11              1     3.944  47.180 -189.358
## - mon5               1     5.196  48.432 -184.958
## - mon10              1     5.345  48.582 -184.441
## - mon9               1    10.694  53.930 -166.894
## - mon6               1    11.128  54.365 -165.545
## - mon7               1    18.090  61.326 -145.303
## - mon8               1    20.509  63.745 -138.804
## - time(ts_data[, 4])  1   118.387 161.624   17.499
##
## Step:  AIC=-203.98
## ts_data[, 4] ~ mon2 + mon4 + mon5 + mon6 + mon7 + mon8 + mon9 +
##     mon10 + mon11 + mon12 + time(ts_data[, 4])
##
##                     Df Sum of Sq     RSS     AIC
## - mon12              1     0.363  43.611 -204.57
## <none>                            43.248 -203.98
## - mon2               1     0.614  43.862 -203.61
## + mon3               1     0.011  43.237 -202.02
## - mon4               1     1.253  44.501 -201.18
## - mon11              1     5.542  48.790 -185.72
## - mon5               1     7.254  50.502 -179.93
## - mon10              1     7.457  50.704 -179.26
```

```
## - mon9                 1    14.724  57.971 -156.75
## - mon6                 1    15.314  58.562 -155.05
## - mon7                 1    24.726  67.973 -130.01
## - mon8                 1    27.989  71.237 -122.14
## - time(ts_data[, 4])  1   118.376 161.624   15.50
##
## Step:  AIC=-204.57
## ts_data[, 4] ~ mon2 + mon4 + mon5 + mon6 + mon7 + mon8 + mon9 +
##     mon10 + mon11 + time(ts_data[, 4])
##
##                      Df Sum of Sq     RSS     AIC
## <none>                             43.611 -204.57
## + mon12               1     0.363  43.248 -203.98
## + mon3                1     0.154  43.456 -203.17
## - mon4                1     0.949  44.560 -202.96
## - mon2                1     1.090  44.701 -202.43
## - mon11               1     5.218  48.829 -187.59
## - mon5                1     6.989  50.600 -181.60
## - mon10               1     7.202  50.813 -180.90
## - mon9                1    14.882  58.493 -157.25
## - mon6                1    15.508  59.119 -155.46
## - mon7                1    25.623  69.234 -128.93
## - mon8                1    29.155  72.766 -120.57
## - time(ts_data[, 4])  1   119.298 162.908   14.83

##
## Call:
## lm(formula = ts_data[, 4] ~ mon2 + mon4 + mon5 + mon6 + mon7 +
##     mon8 + mon9 + mon10 + mon11 + time(ts_data[, 4]))
##
## Coefficients:
##        (Intercept)                mon2                mon4
##             6.5985              0.3222             -0.3007
##               mon5                mon6                mon7
##            -0.8159             -1.2153             -1.5622
##               mon8                mon9               mon10
##            -1.6665             -1.1907             -0.8285
##              mon11  time(ts_data[, 4])
##            -0.7052             -0.0174
```

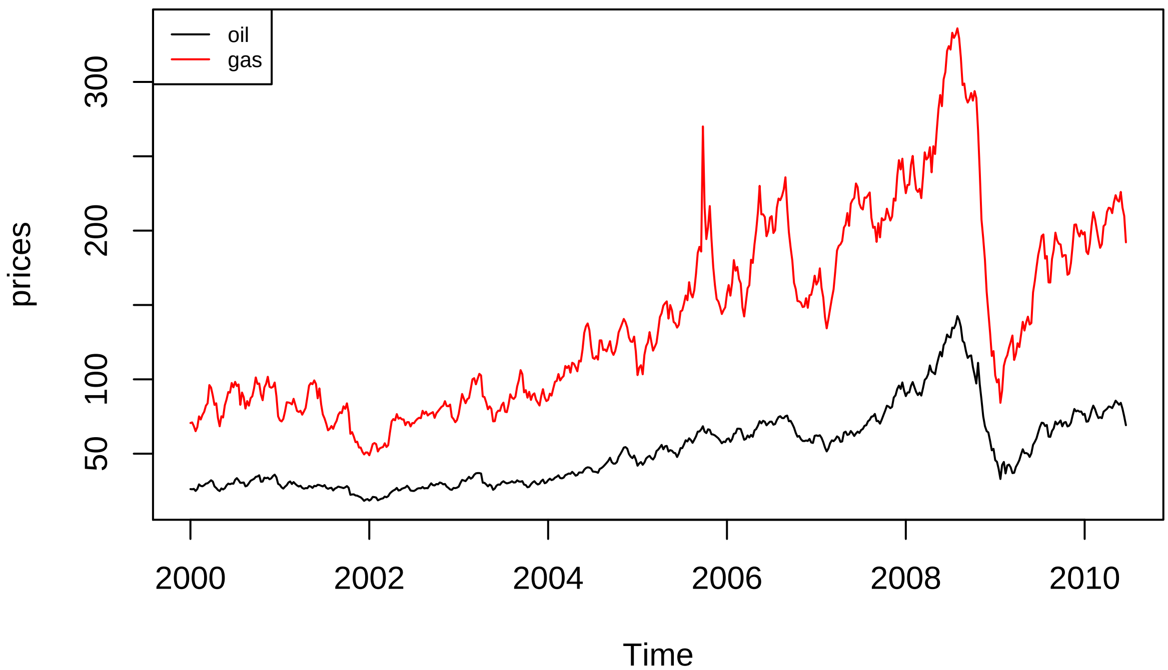
```
#step(smm, direction="both")
```

After performing stepwise model selection in both direction, mon3 and mon12 has been dropped with AIC=204.57. Our final model includes all month variable except 3rd month and 12th month.

#Assignment3. Analysis of oil and gas time series Weekly time series **oil** and **gas** present in the package *astsa* show the oil prices in dollars per barrel and gas prices in cents per dollar.

##a) Plot the given time series in the same graph. Do they look like stationary series? Do the processes seem to be related to each other?

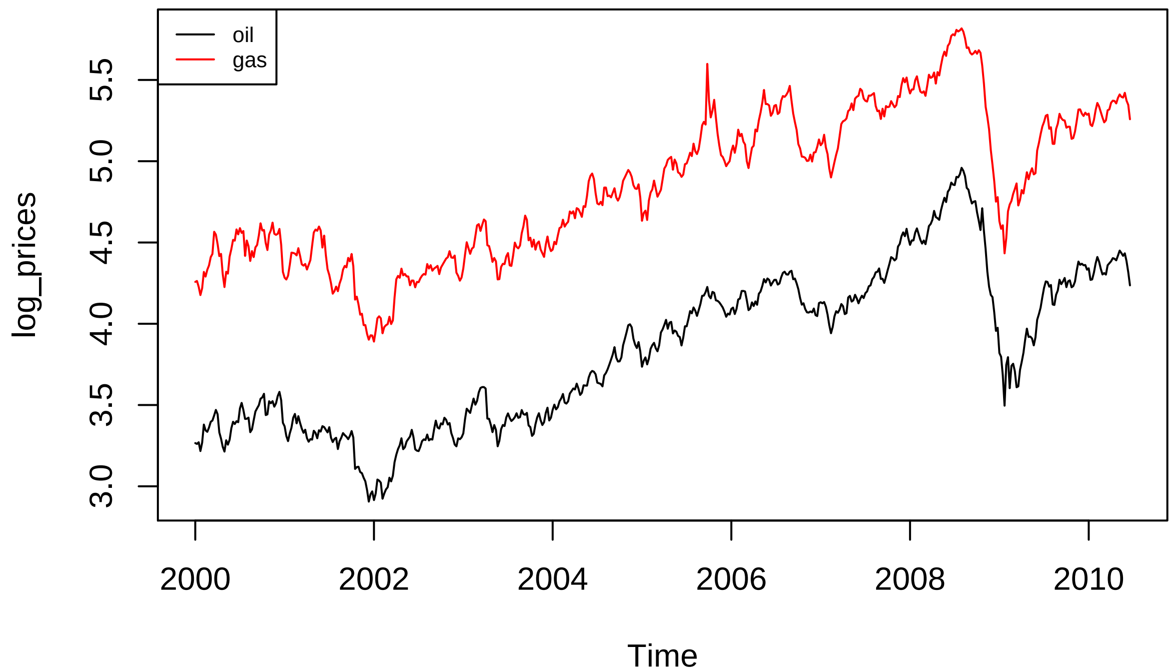
```
prices <- cbind(oil, gas)

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 general trend of fluctutation seems to be dependent on month(t) which can be also interpreted as the mean value of series depends on time t. We can conclude both process is not stationary. 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) Apply log-transformation to the series Apply log-transformation to the time series and plot the transformed data. In what respect did this transformtion made the data easier for the analysis?

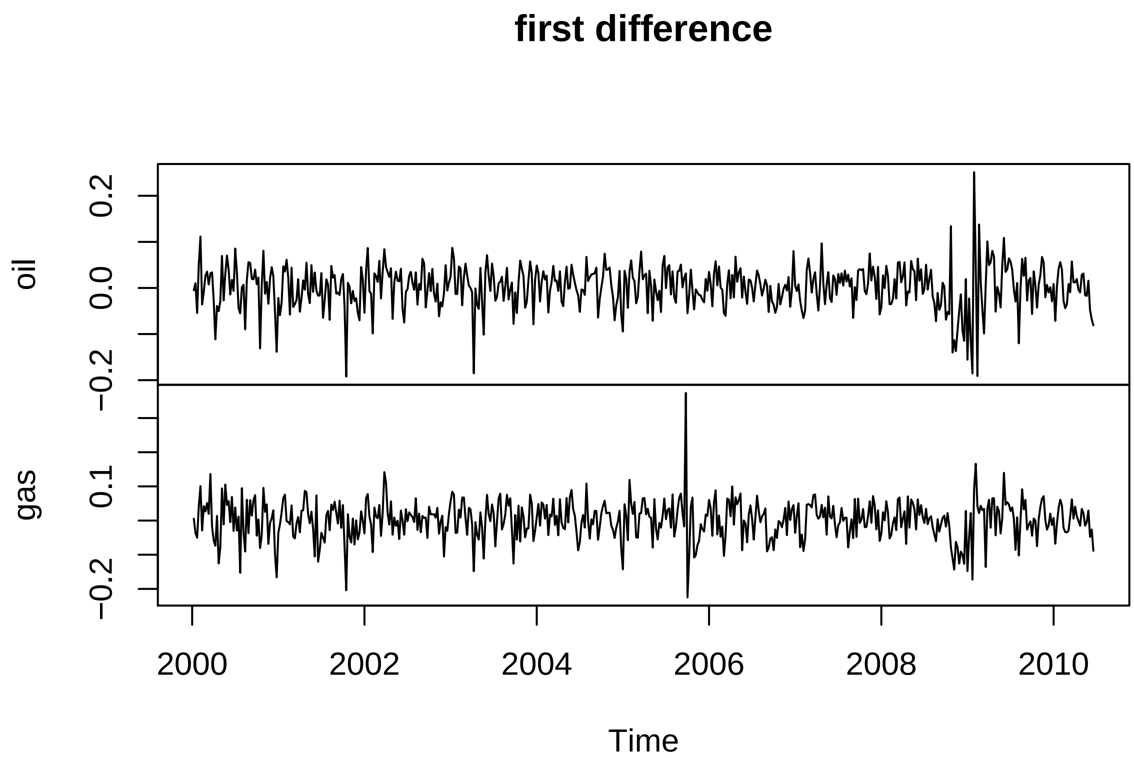
```
log_prices <- log(prices)

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. Generally log-transformation makes scale and patterns more interpretable.

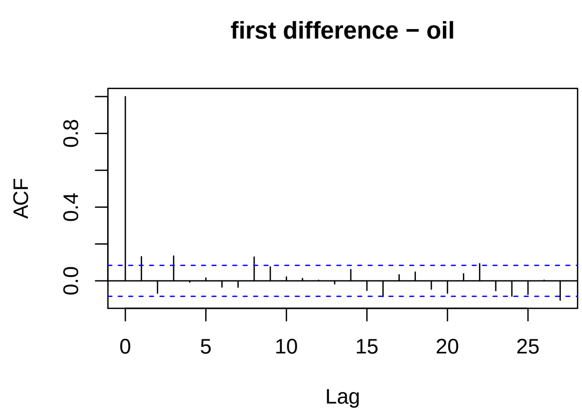
##c) Compute the first difference to eliminate the trend. To eliminate trend, compute the first difference of the transformed data, plot the detrended series, check their ACFs and analyze the obtained plots. Denote the data obtained here as x_t(oil) and y_t(gas).

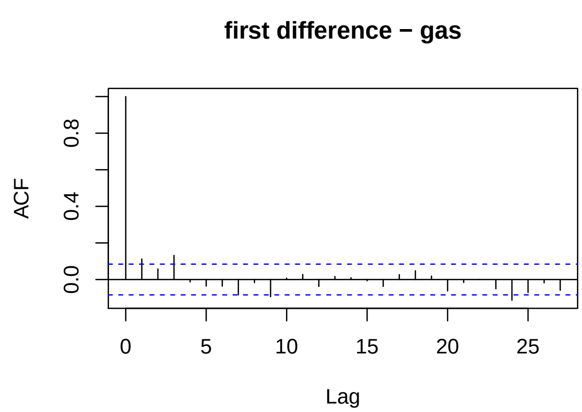
```
diff_log_prices <- as.data.frame(diff(log_prices))
plot(diff(log_prices), type="l", main="first difference")
```



The trend seems to be no longer exist.

```
log_prices_df <- as.data.frame(log_prices)

par(mfrow=c(1,2), oma=c(0,0,2,0))
acf(diff(log_prices_df$oil), main="first difference - oil")
acf(diff(log_prices_df$gas), main="first difference - gas")
```



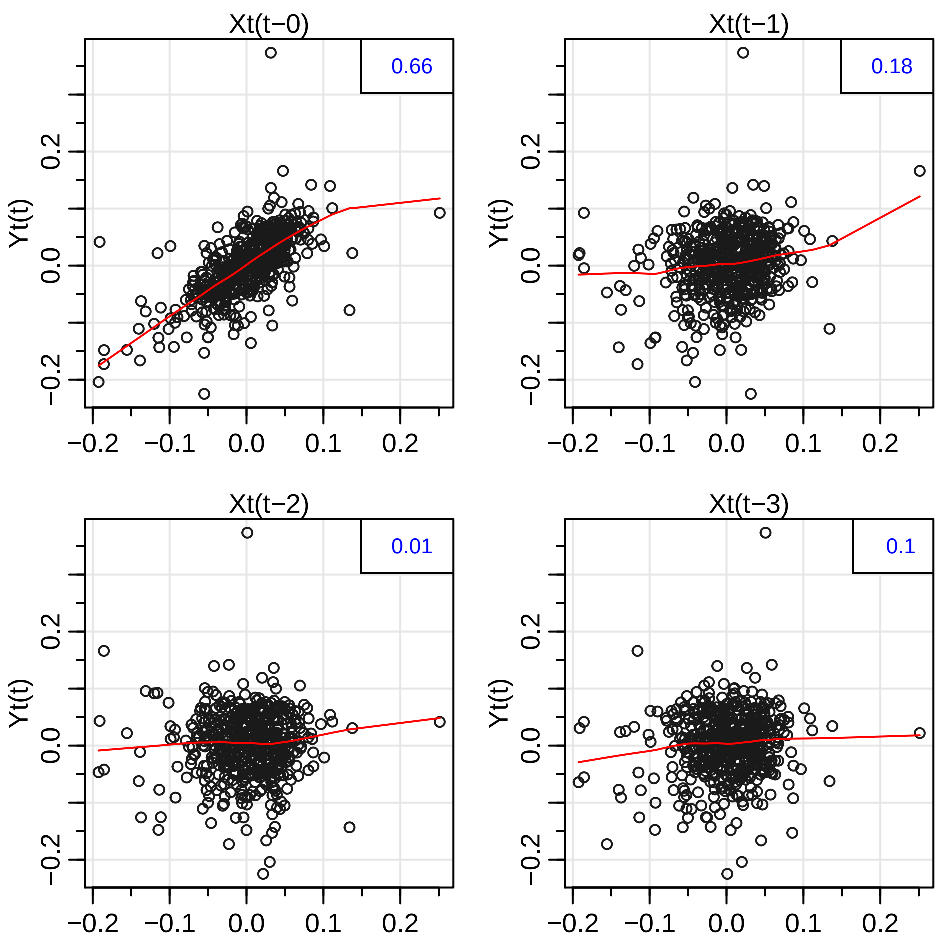


ACF tends to stay in blue line and it doesn't show any significant pattern for both case. Detrending with differenciation on log-transformed data gets more close to white noise compare to original data. However in general we would say there is a bit more autocorrelation in oil than in gas.

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

##d) Exhibit scatterplots of x_t and y_t for up to three weeks of lead time of xt include a nonparametric smoother in each plot and comment the results.

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



Are there outlier? Are there relationship linear? Are there changes in the trend? There is a outlier. We might say that there is linear relationship between Y_t and X_{t-0}, but it is hard to say there are linear relationship between y_t and x_t given lags for the rest. We can see that as lag extends it becomes less linearly related.

##e) Fit the following model(Regression with lagged variables)

$$y_t = a_0 + a_1 I(x_t > 0) + \beta_1 x_t + \beta_2 x_{t-1} + w_t$$

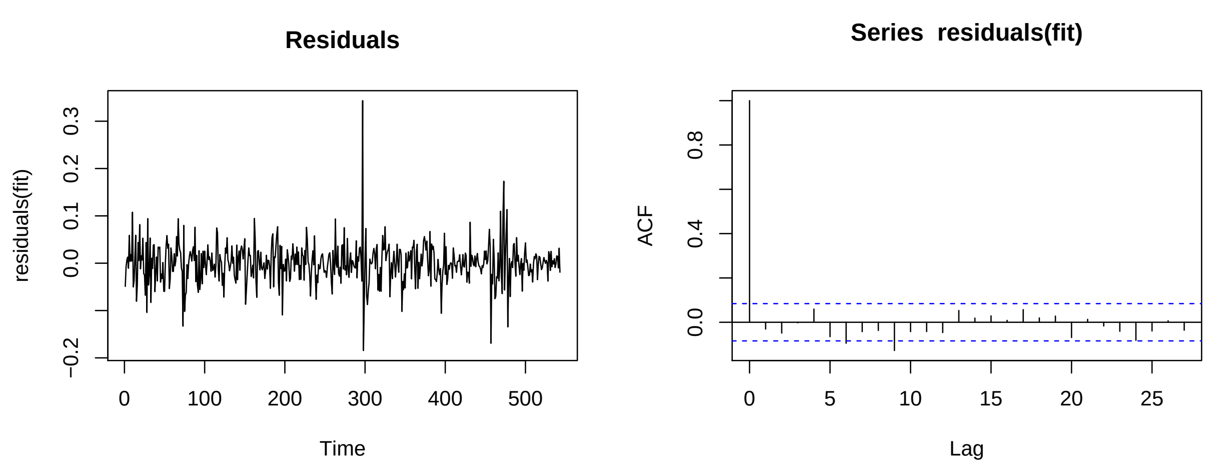
Check which coefficients seem to be significant. How can this be interpreted? Analyze the residual pattern and the ACF of the residuals.

```
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. x_t seems to be the most significant. This can be interpreted that x_t plays the most important role when predicting y_t. Thus in determining the price of gas at time = t, is largely explained by the price of oil at time = t, rather than the price of oil at t=-1, or whether the price change in oil at time=t has been positive or negative

```r
par(mfrow=c(1,2))
plot.ts(residuals(fit), main="Residuals")
acf(residuals(fit))
```



The residual plot above shows that residual fluctuates around 0 (except for two spikes) where we can assumen zero mean. ACF function also shows that it stays inside blue line without any significant pattern. We can conclude that the residual follows zero mean white noise.