Time Series (732A62) Lab1

Anubhav Dikshit(anudi287) and Maximilian Pfundstein(maxpf364)

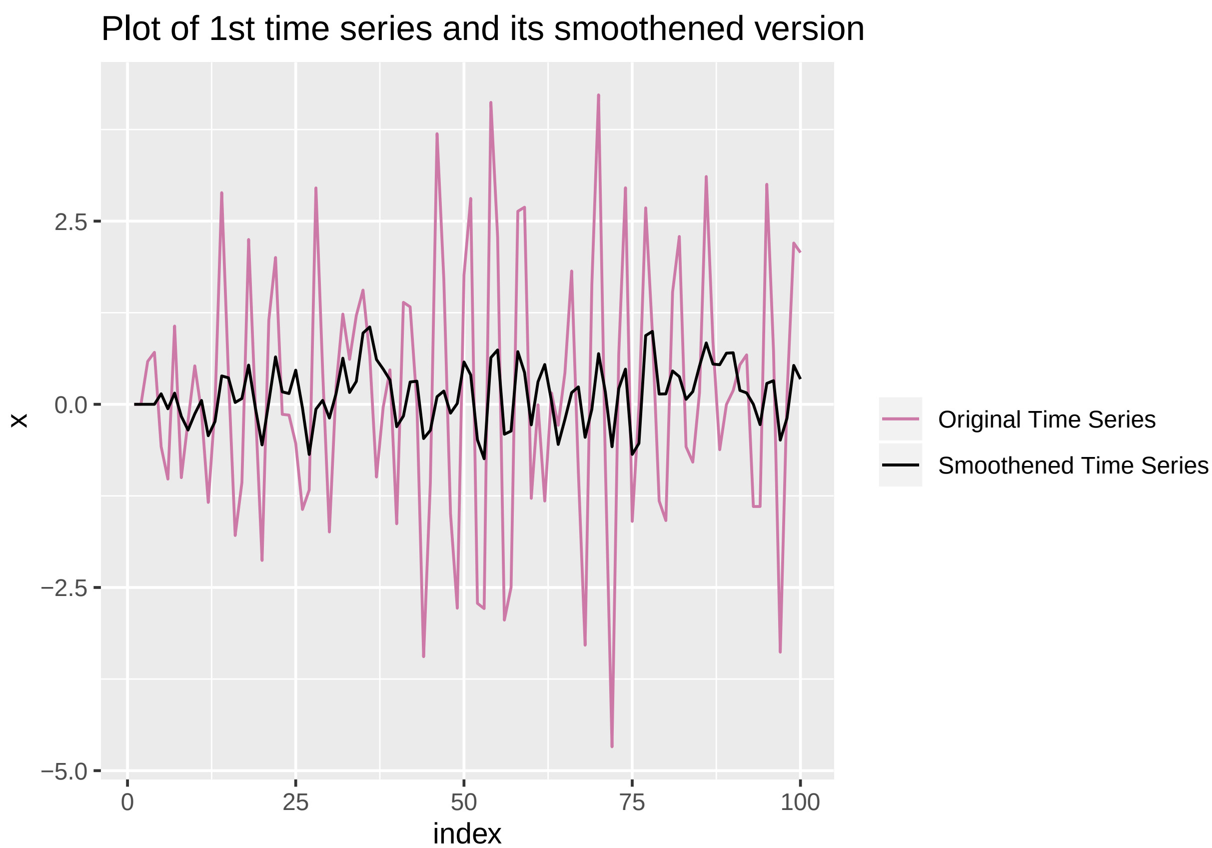
09 September, 2019

Contents

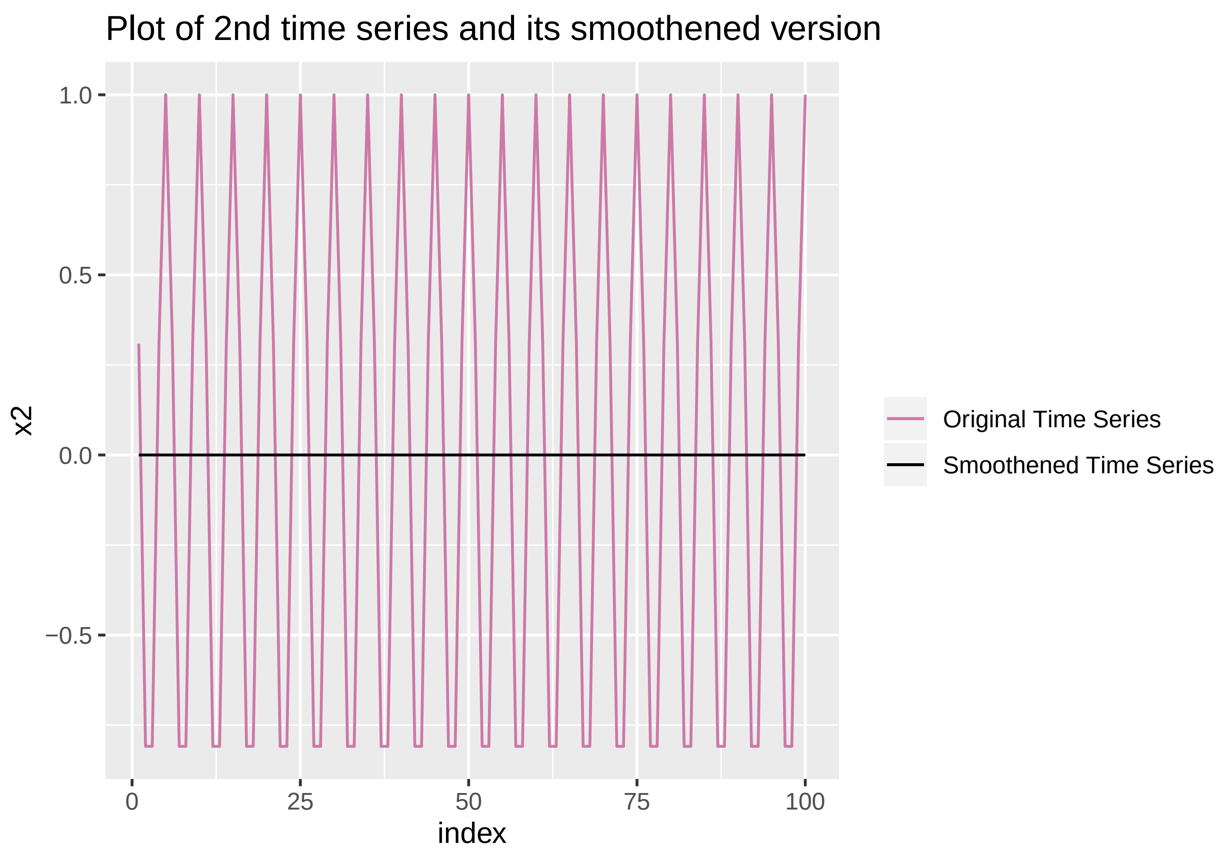
Assignment 1. Computations with simulated data

a) Generate two time series $x_t = -0.8x_{t-2} + w_t$, where $x_0 = x_1 = 0$ and $x_t = cos(\frac{2\pi t}{5})$ with 100 observations each. Apply a 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.

```
set.seed(12345)

n = 100
x <- vector(length = n)
x2 <- vector(length = n)

x[1] <- 0
x[2] <- 0

#first series generation
for(i in 3:n){
  x[i] <- -0.8 * x[i-2] + rnorm(1,0,1)
}

#second series generation
for(i in 1:n){
  x2[i] <- cos(0.4*pi*i)
}

# smoothing filter function
smoothing_filter <- function(x){
v <- vector(length = length(x))
for(i in 5:length(x)){
  v[i] = 0.2 * (x[i] + x[i-1] + x[i-2] + x[i-3] + x[i-4])
}
return(v)
}

#generate smoothed series
smooth_x <- smoothing_filter(x)
smooth_x2 <- smoothing_filter(x2)

#adding everything to a dataframe
df <- cbind(x,x2,smooth_x,smooth_x2) %>% as.data.frame() %>% mutate(index=1:100)

ggplot(df, aes(x=index)) +
  geom_line(aes(y=x, color="Original Time Series")) +
  geom_line(aes(y=smooth_x, color="Smoothened Time Series")) +
  ggtitle("Plot of 1st time series and its smoothened version") +
    scale_colour_manual("", breaks = c("Original Time Series", "Smoothened Time Series"),
                        values = c("#CC79A7", "#000000"))
```




```
ggplot(df, aes(x=index)) +
  geom_line(aes(y=x2, color="Original Time Series")) +
  geom_line(aes(y=smooth_x2, color="Smoothened Time Series")) +
  ggtitle("Plot of 2nd time series and its smoothened version") +
    scale_colour_manual("", breaks = c("Original Time Series", "Smoothened Time Series"),
                        values = c("#CC79A7", "#000000"))
```



b) Consider time series $x_t - 4x_{t-1} + 2x_{t-2} + x_{t-5} = w_t + 3w_{t-2} + w_{t-4} - 4w_{t-6}$. Write an appropriate R code to investigate whether this time series is casual and invertible.

Causality: ARMA(p,q) is causal iff roots $\phi(z') = 0$ are outside unit circle. eg: $x_t = 0.4x_{t-1} + 0.3x_{t-2} + w_t$, roots are -> $1 - 0.4B + 0.3B^2$

equation is: $\phi(Z) = 1 - 4B + 2B^2 + 0B^3 + 0B^4 + B^5$

```
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
```

```
any(Mod(polyroot(z))<=1)
```

```
## [1] TRUE
```

Invertible: ARMA(p,q) is causal iff roots $\theta(z') = 0$ are outside unit circle.

equation is: $\theta(Z) = 1 + 3B^2 + B^4 - 4B^6$

```
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
```

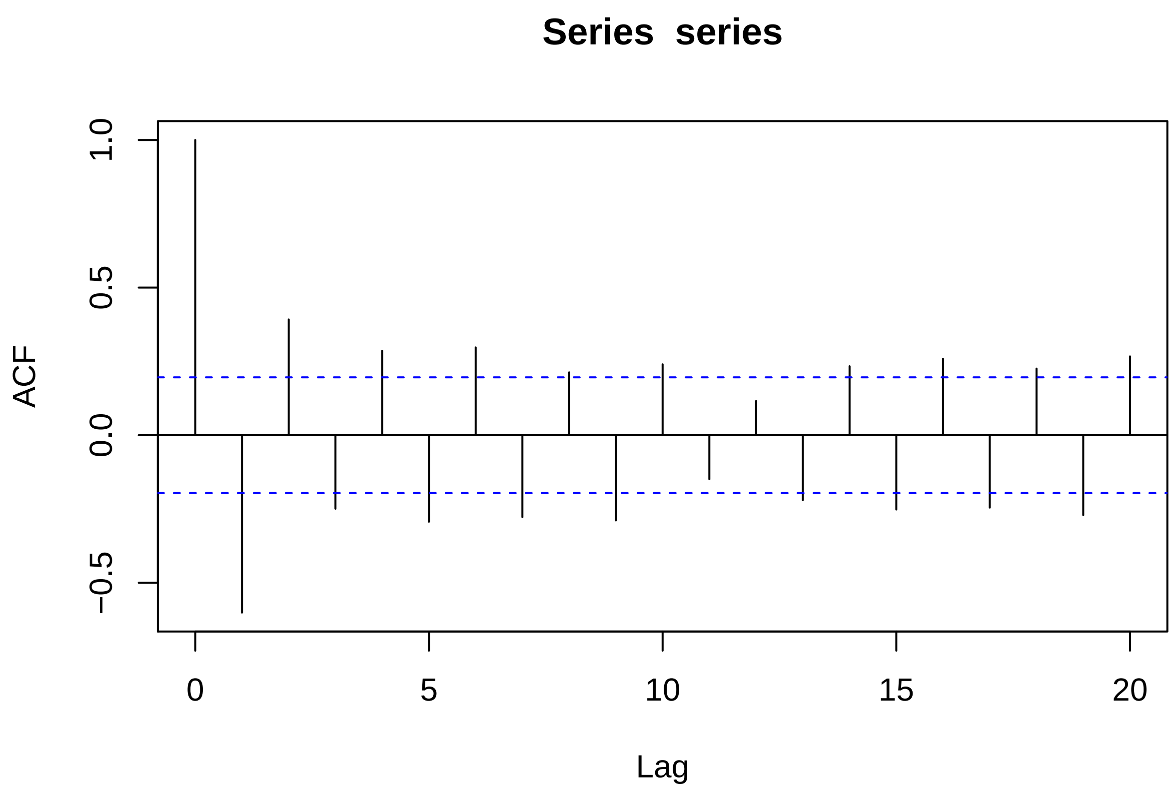
```r
any(Mod(polyroot(z))<=1)
```

```
## [1] TRUE
```

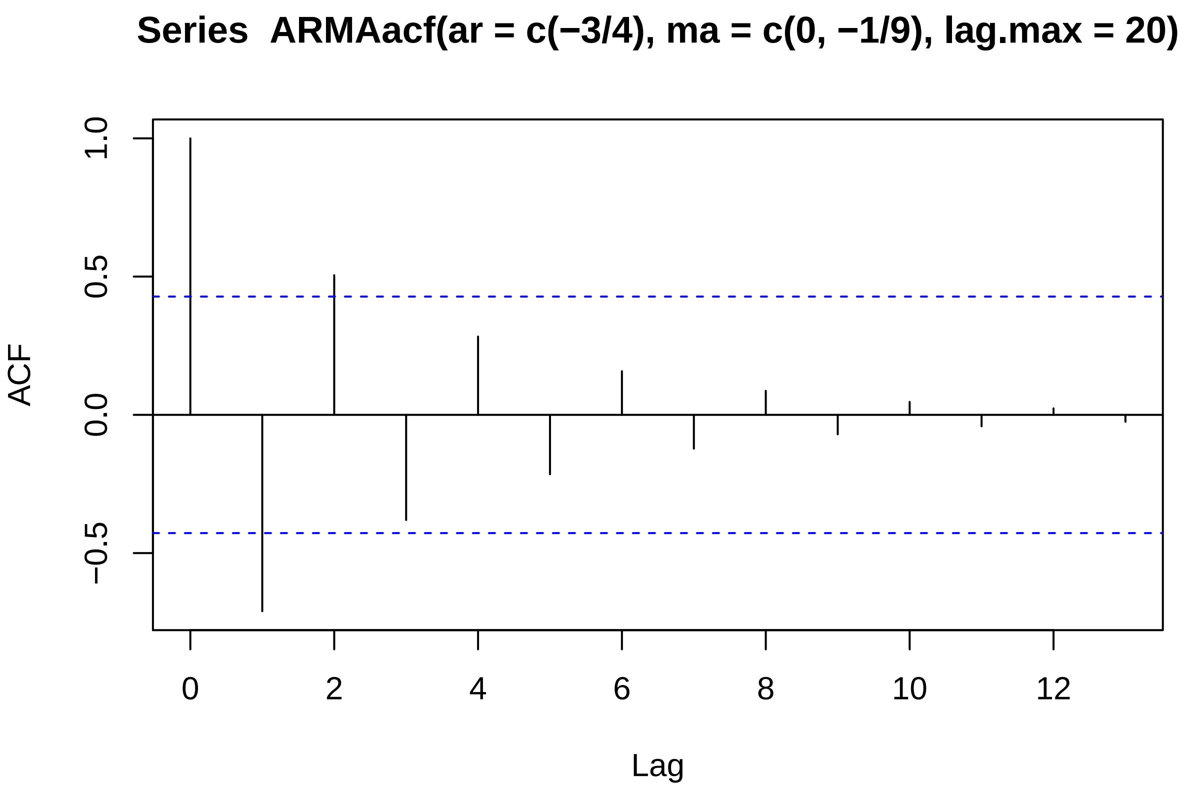
Analysis: Baring one of the roots all are inside the unit circle. Thus the time series is not invertiable.

c) Use built-in R functions to simulate 100 observations from the process $x_t + \frac{3}{4}x_{t-1} = w_t - \frac{1}{9}w_{t-2}$ compute sample ACF and theoretical ACF, use seed 54321. Compare the ACF plots.

```r
set.seed(54321)

series <- arima.sim(n = 100, list(ar = c(-3/4), ma = c(0,-1/9)))

acf(series)
```




```r
acf(ARMAacf(ar = c(-3/4), ma =  c(0,-1/9), lag.max = 20))
```



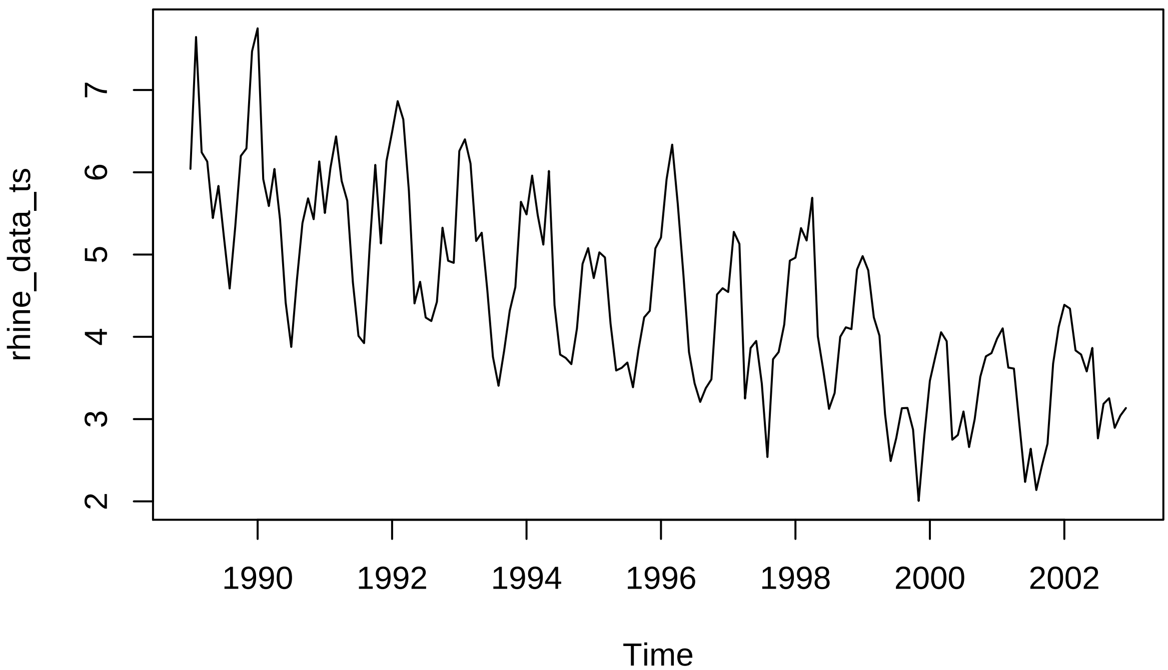
Analysis: In the theortical ACF, only the 1 and 2nd lag components were significant, while using the sample ACF function we get many more lag components as significant.

Assignment 2. Visualization, detrending and residual analysis of Rhine data.

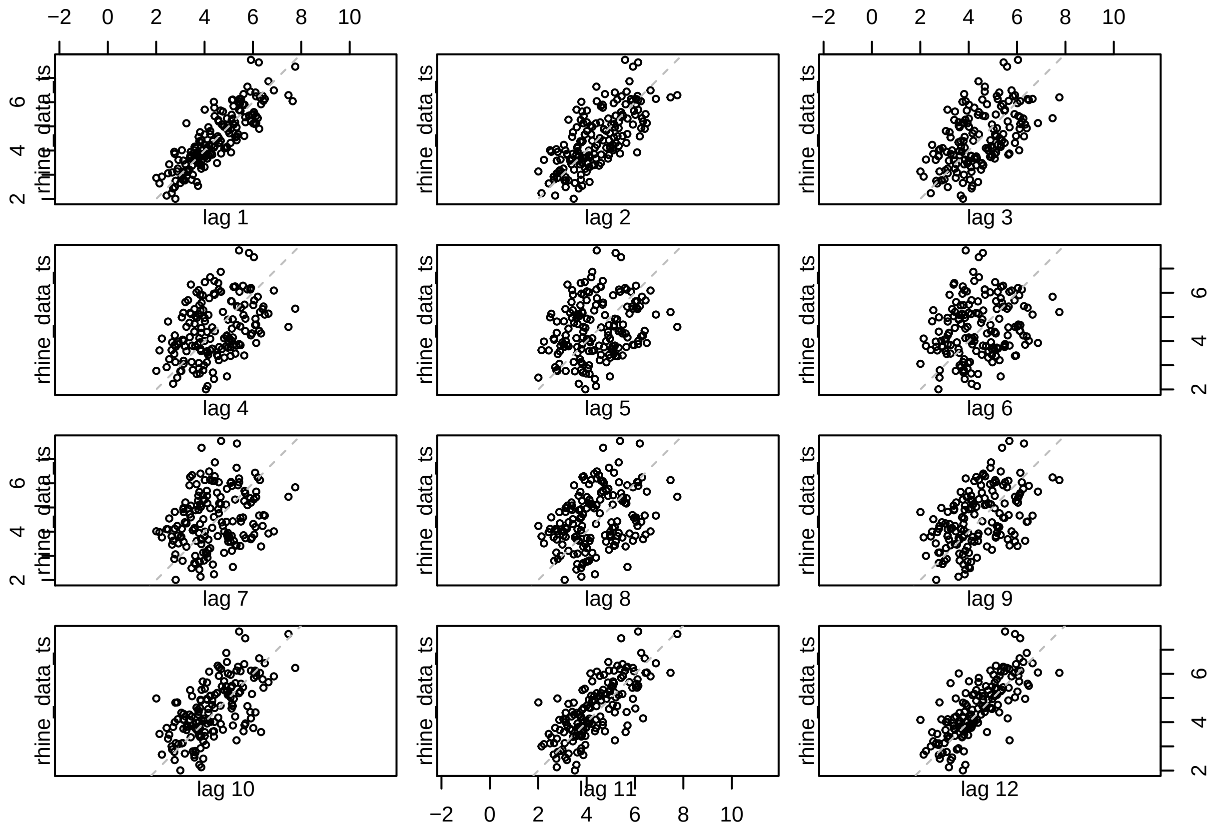
The data set Rhine.csv contains monthly concentrations of total nitrogen in the Rhine River in the period 1989-2002.

a) Import the data to R, convert it appropriately to ts object (use function ts()) and explore it by plotting the time series, creating scatter plots of x_t against $x_{t-1}, \ldots x_{t-12}$. Analyze the time series plot and the scatter plots: Are there any trends, linear or seasonal, in the time series? When during the year is the concentration highest? 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?

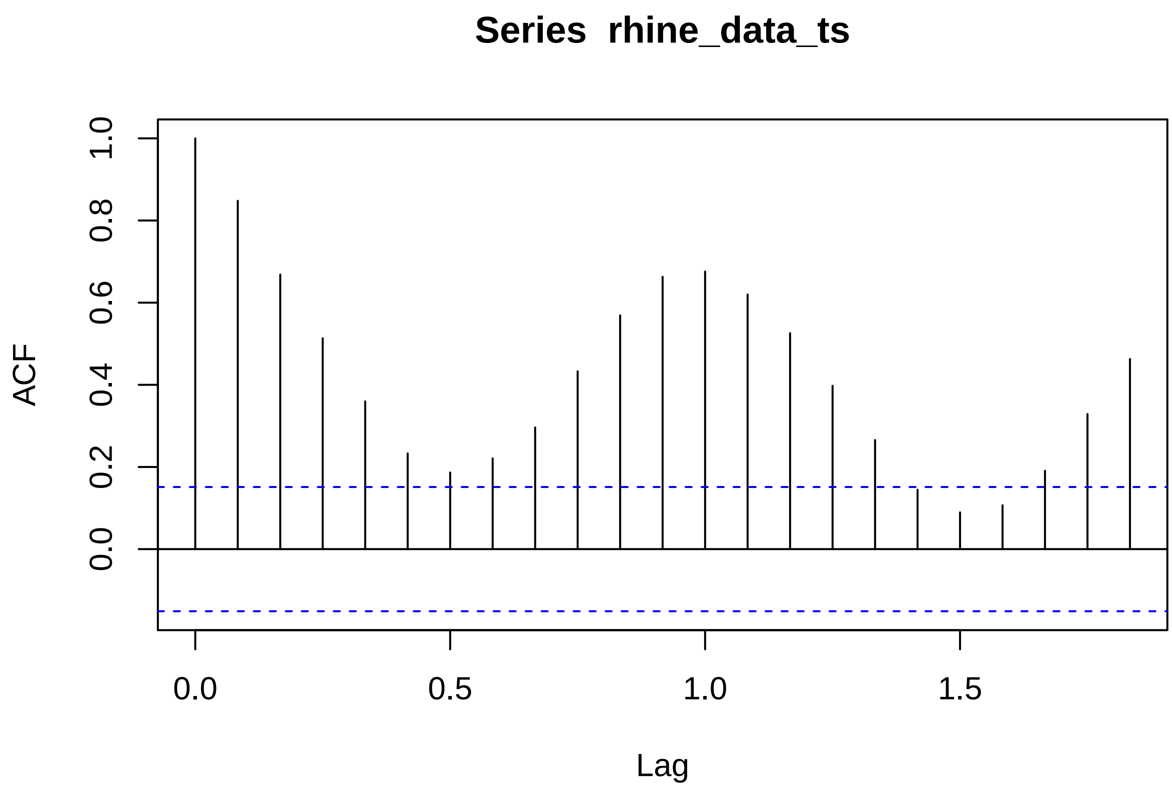
```
set.seed(12345)
rhine_data <- read.csv2("Rhine.csv")
rhine_data_ts <- ts(data = rhine_data$TotN_conc,
                    start = c(1989,1),
                    frequency = 12)

plot.ts(rhine_data_ts)
```



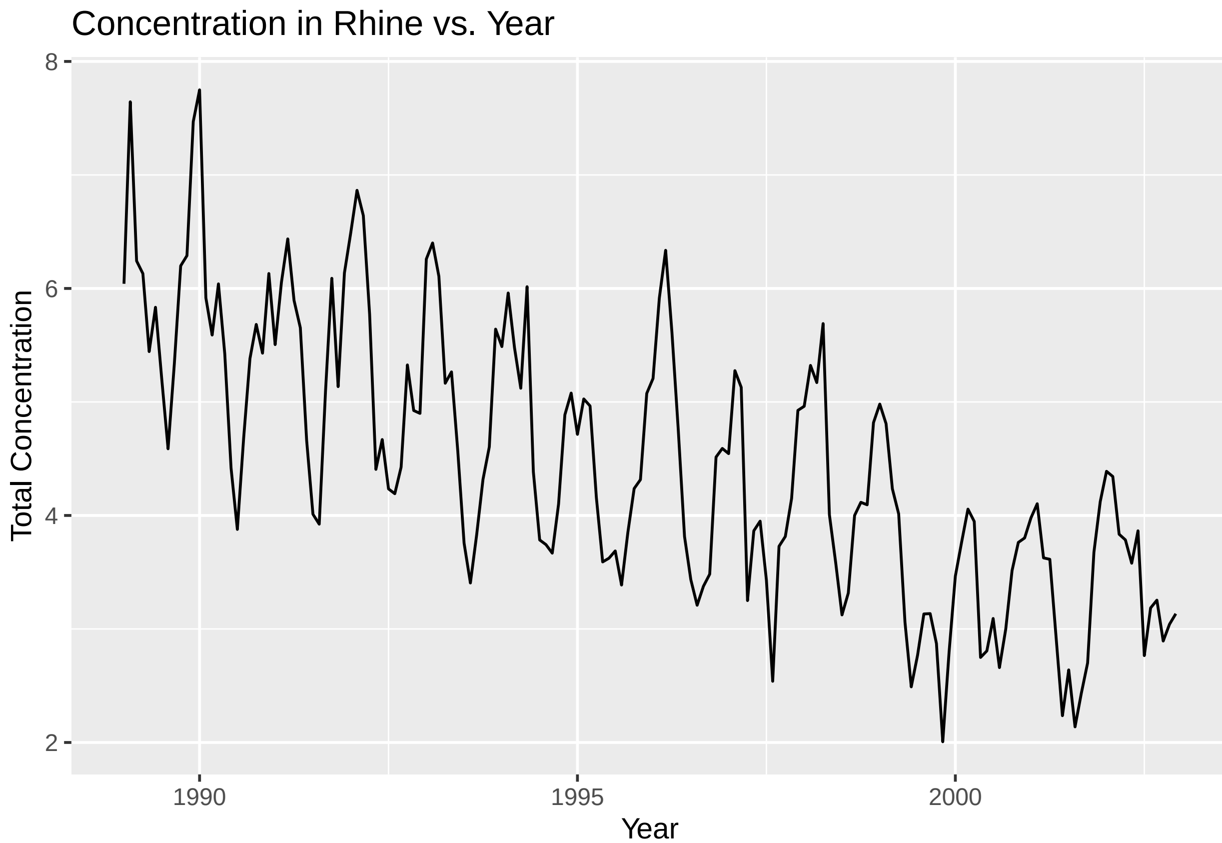

```
lag.plot(rhine_data_ts,lags = 12)
```



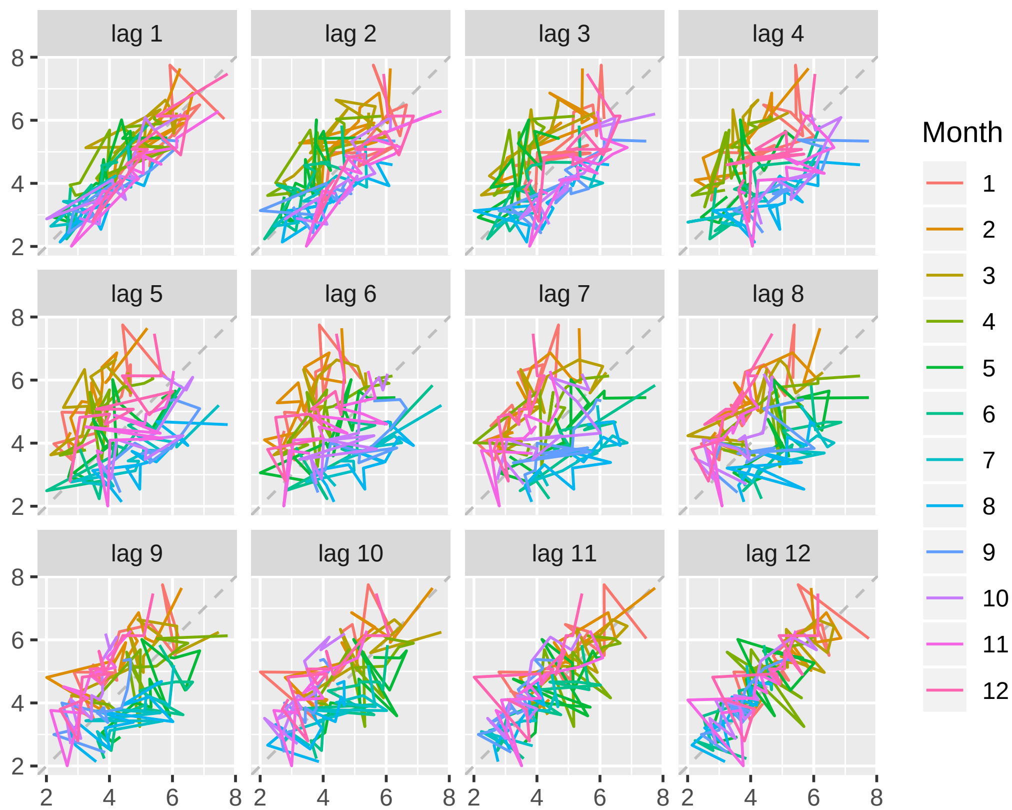

```r
acf(rhine_data_ts)
```



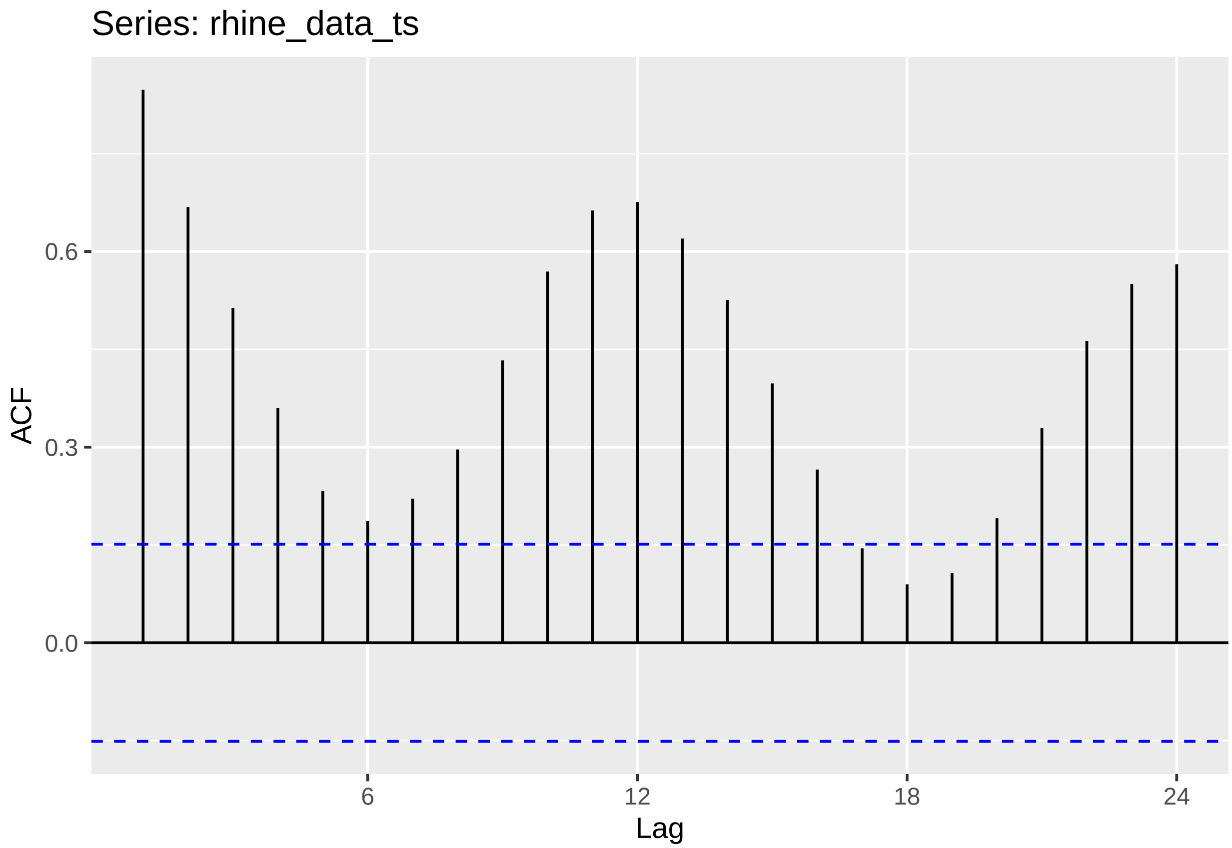

```r
#alternative
autoplot(rhine_data_ts) + ylab("Total Concentration") +xlab("Year") +
  ggtitle("Concentration in Rhine vs. Year")
```




```
gglagplot(rhine_data_ts, lags = 1, set.lags = 1:12, color=FALSE)
```



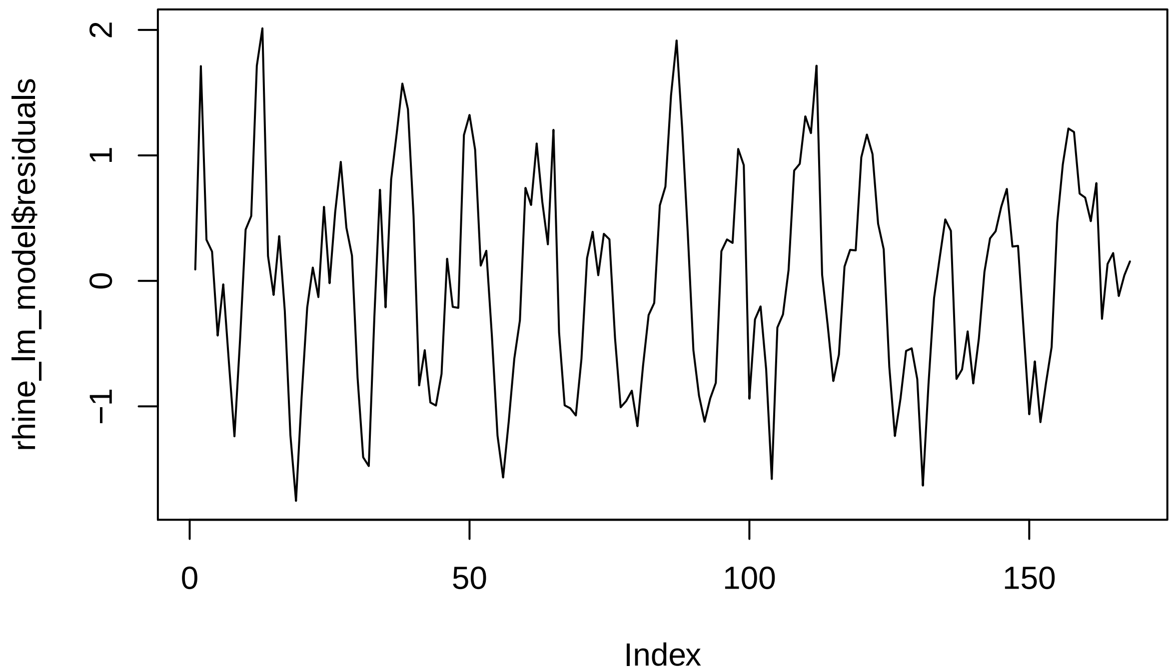

```
ggAcf(rhine_data_ts)
```



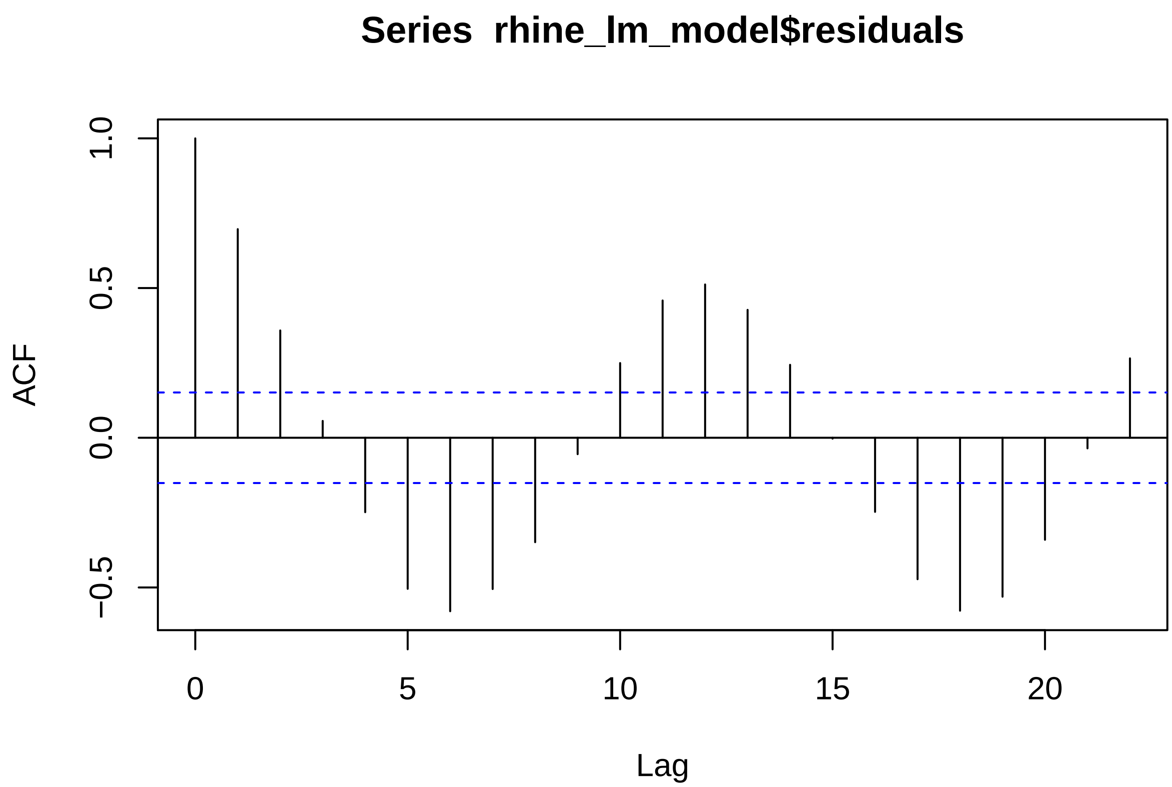
Analysis:

b) Eliminate the trend by fitting a linear model with respect to t to the time series. Is there a significant time trend? Look at the residual pattern and the sample ACF of the residuals and comment how this pattern might be related to seasonality of the series.

```
set.seed(12345)

rhine_lm_model <- lm(TotN_conc~Time, data=rhine_data)
plot(rhine_lm_model$residuals, type = 'l')
```



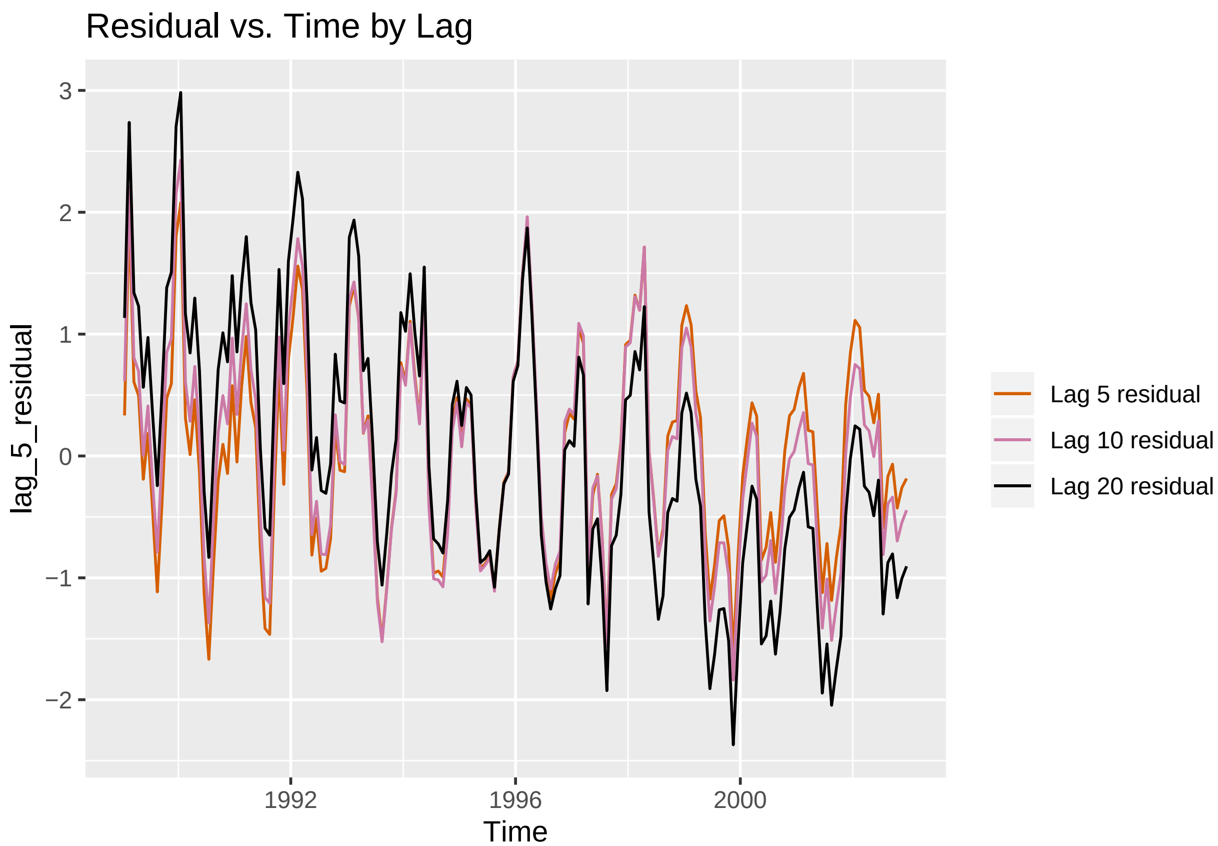

```
acf(rhine_lm_model$residuals)
```



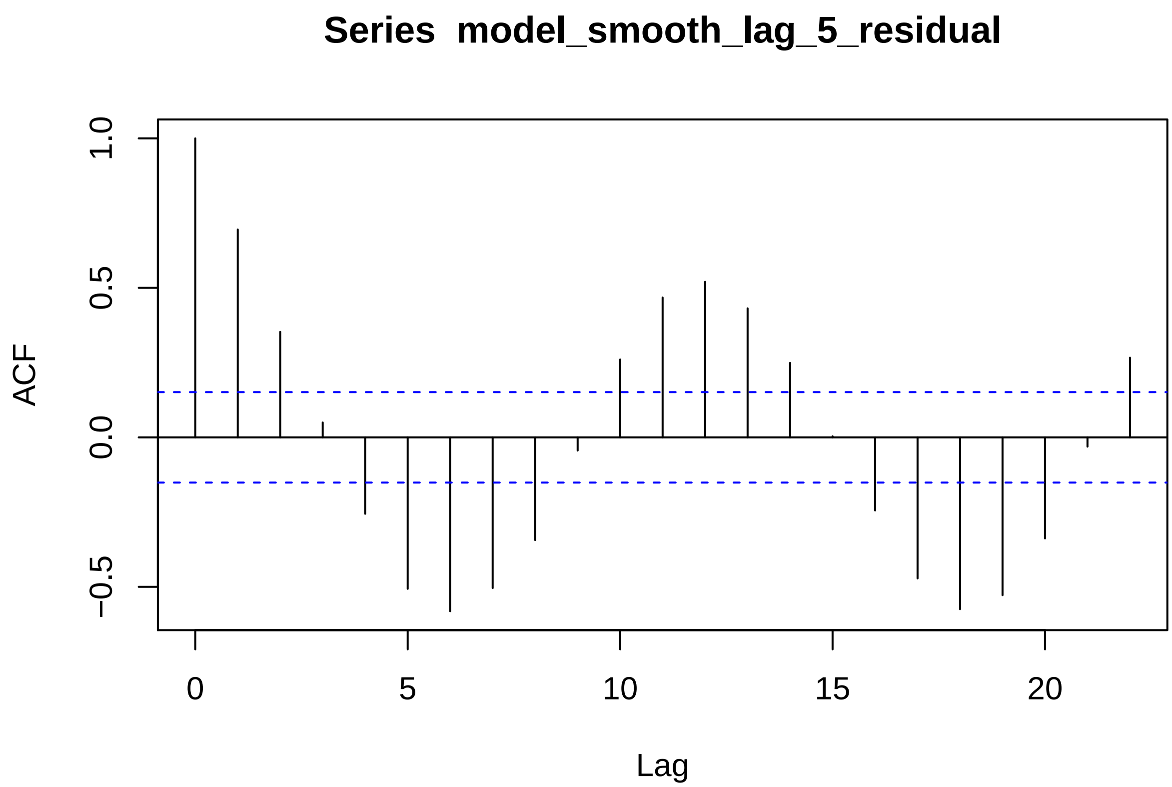
c) Eliminate the trend by fitting a kernel smoother with respect to t to the time series (choose a reasonable bandwidth yourself so the fit looks reasonable). Analyze the residual pattern and the sample ACF of the residuals and compare it to the ACF from step b). Conclusions? Do residuals seem to represent a stationary series?

```
set.seed(12345)

model_smooth_lag_5 <- ksmooth(x = rhine_data$Time, y = rhine_data$TotN_conc, bandwidth=5)
model_smooth_lag_10 <- ksmooth(x = rhine_data$Time, y = rhine_data$TotN_conc, bandwidth=10)
model_smooth_lag_20 <- ksmooth(x = rhine_data$Time, y = rhine_data$TotN_conc, bandwidth=20)

model_smooth_lag_5_residual <- rhine_data$TotN_conc - model_smooth_lag_5$y
model_smooth_lag_10_residual <- rhine_data$TotN_conc - model_smooth_lag_10$y
model_smooth_lag_20_residual <- rhine_data$TotN_conc - model_smooth_lag_20$y

residual_df <- cbind(model_smooth_lag_5_residual, model_smooth_lag_10_residual,
                     model_smooth_lag_20_residual, rhine_data$Time) %>% as.data.frame()

colnames(residual_df) <- c("lag_5_residual", "lag_10_residual", "lag_20_residual", "Time")

ggplot(residual_df, aes(x=Time)) +
  geom_line(aes(y=lag_5_residual, color="Lag 5 residual")) +
  geom_line(aes(y=lag_10_residual, color="Lag 10 residual")) +
  geom_line(aes(y=lag_20_residual, color="Lag 20 residual")) +
  ggtitle("Residual vs. Time by Lag") +
    scale_colour_manual("", breaks = c("Lag 5 residual", "Lag 10 residual", "Lag 20 residual"),
```

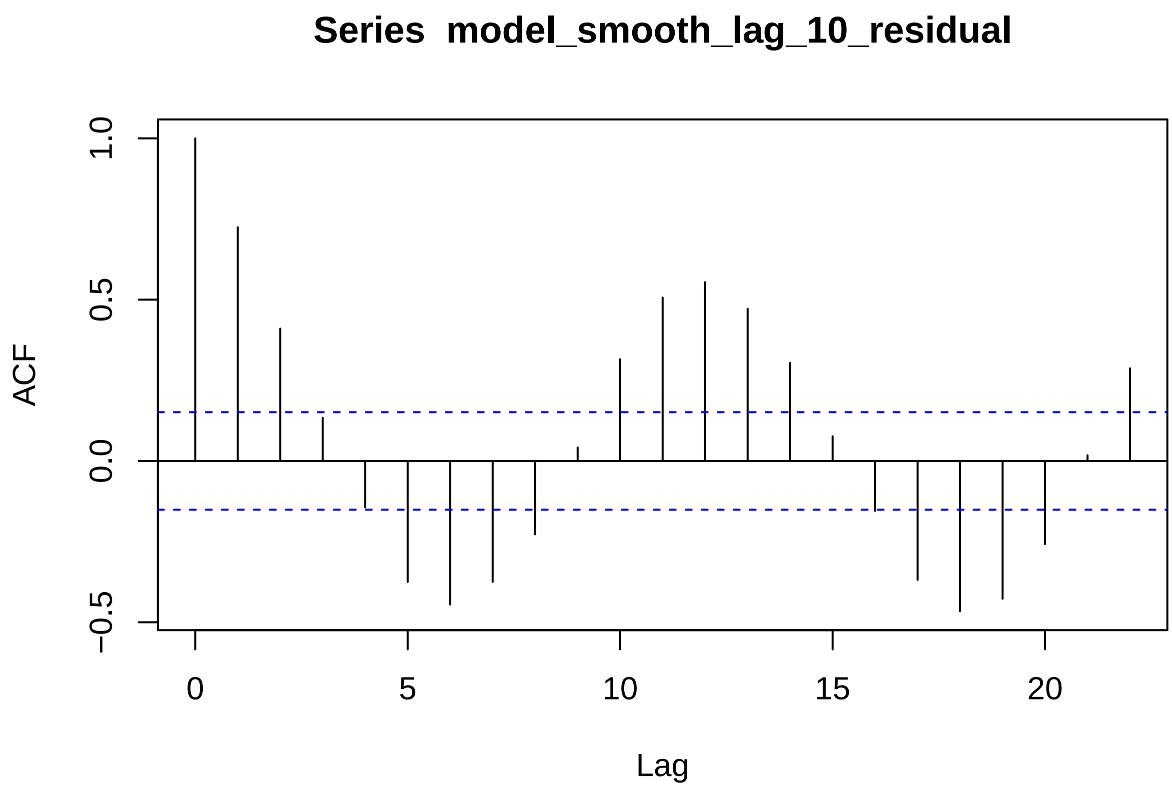
```
values = c("#CC79A7", "#000000", "#D55E00"))
```



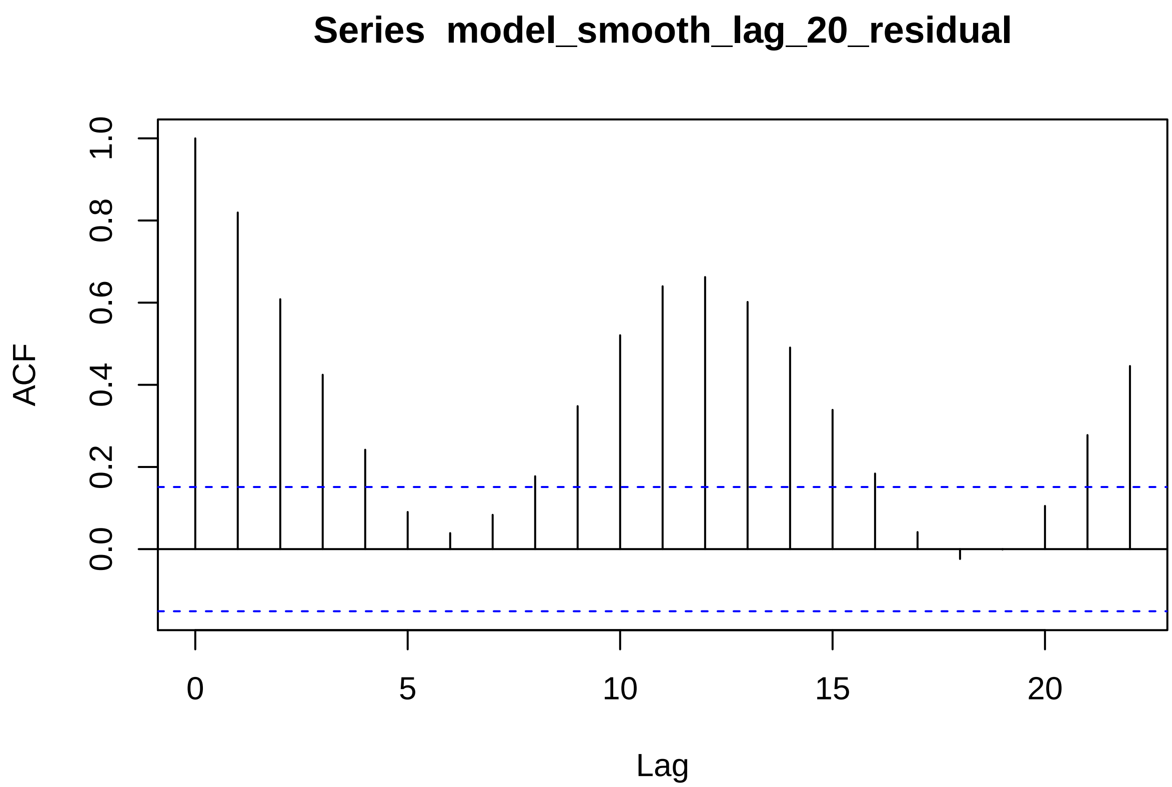

```
acf(model_smooth_lag_5_residual)
```




```r
acf(model_smooth_lag_10_residual)
```

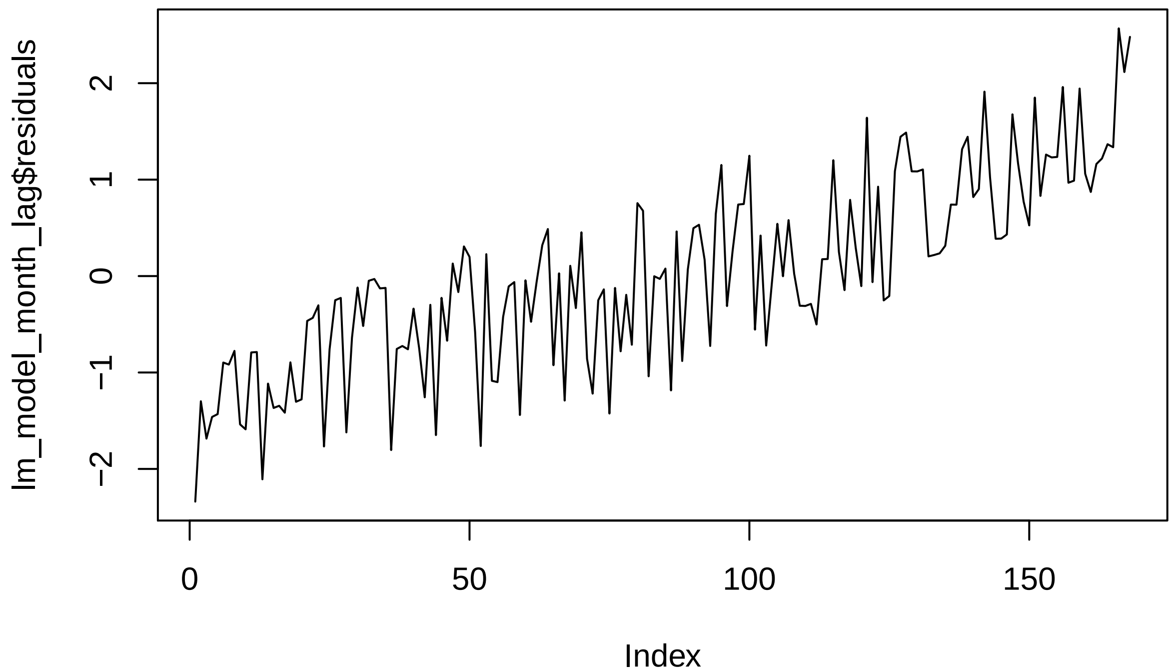



```
acf(model_smooth_lag_20_residual)
```

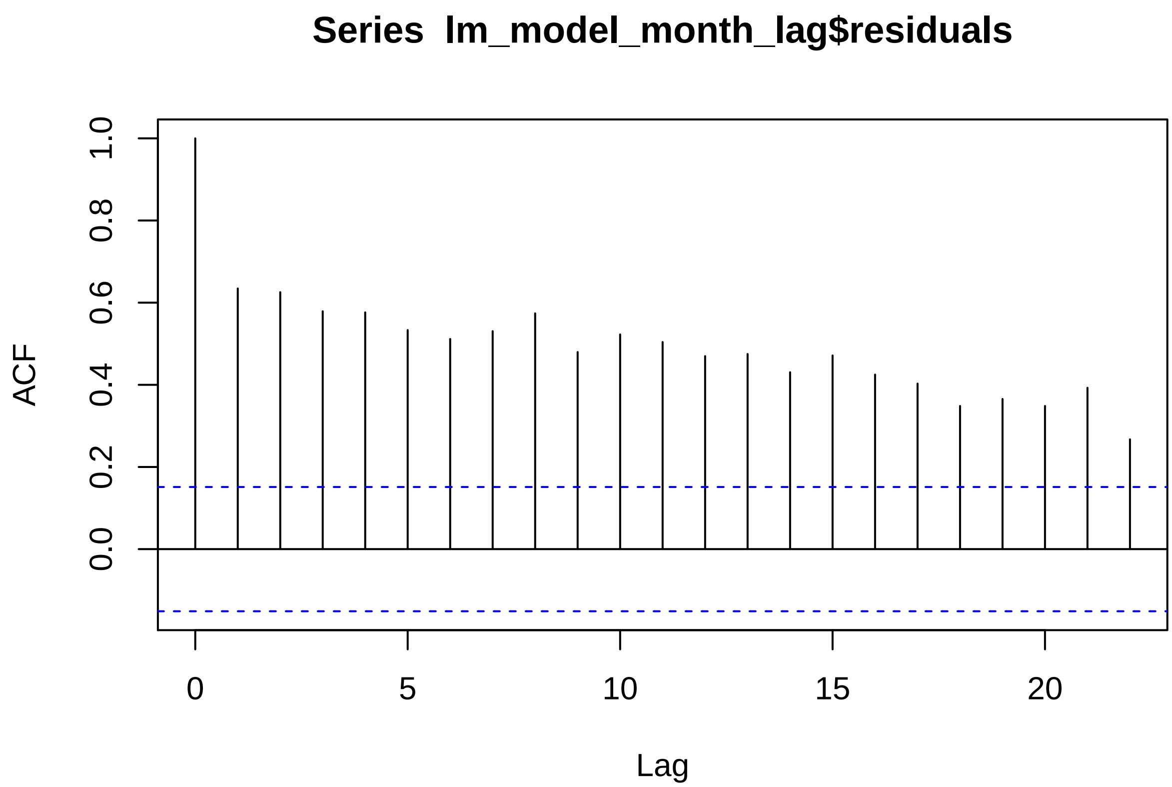


d) Eliminate the trend by fitting the following so-called seasonal means model: $x_t = \alpha_0 + \alpha_1 t + \beta_1 I(month = 2) + ... + \beta_{12} I(month = 12) + w_t$, where I(x)=1 is an identity function. Fitting of this model will require you to augment data with a categorical variable showing the current month, and then fitting a usual linear regression. Analyze the residual pattern and the ACF of residuals.

```
set.seed(12345)

rhine_data_wide <- rhine_data
rhine_data_wide$dummy <- "1"
rhine_data_wide$Month <- paste0("Month_",rhine_data_wide$Month)
rhine_data_wide <- dcast(rhine_data_wide, formula = TotN_conc+Year+Time~Month, value.var = "dummy", fil

lm_model_month_lag <- lm(data=rhine_data_wide,
                    TotN_conc~Month_1+Month_2+Month_3+Month_4+Month_5+Month_6+Month_7+
                      Month_8+Month_9+Month_10+Month_11+Month_12)


plot(lm_model_month_lag$residuals, type = 'l')
```




```r
acf(lm_model_month_lag$residuals)
```



Analysis:

e) Perform stepwise variable selection in model from step d). Which model gives you the lowest AIC value? Which variables are left in the model?

```
set.seed(12345)

lm_model_month_lag_step <- stepAIC(lm_model_month_lag, scope = list(upper = ~Month_1+Month_2+
                                                                   Month_3+Month_4+Month_5+Month_6+Month_
                                                                     Month_8+Month_9+Month_10+Month_11+Mo
                                                                   lower = ~1), trace = TRUE,
                                   direction="backward")
```

```
## Start:  AIC=17.5
## TotN_conc ~ Month_1 + Month_2 + Month_3 + Month_4 + Month_5 +
##     Month_6 + Month_7 + Month_8 + Month_9 + Month_10 + Month_11 +
##     Month_12
##
##
## Step:  AIC=17.5
## TotN_conc ~ Month_1 + Month_2 + Month_3 + Month_4 + Month_5 +
##     Month_6 + Month_7 + Month_8 + Month_9 + Month_10 + Month_11
##
##            Df Sum of Sq    RSS    AIC
## - Month_4   1    0.0064 161.63 15.506
## - Month_1   1    0.9497 162.57 16.484
```

```
## - Month_3   1    0.9775 162.60 16.512
## <none>                 161.62 17.499
## - Month_11  1    2.1676 163.79 17.738
## - Month_5   1    2.2167 163.84 17.788
## - Month_2   1    2.7569 164.38 18.341
## - Month_10  1    3.0705 164.69 18.661
## - Month_6   1    6.7169 168.34 22.340
## - Month_9   1    7.1009 168.72 22.723
## - Month_7   1   12.6417 174.26 28.151
## - Month_8   1   15.0349 176.66 30.443
##
## Step:  AIC=15.51
## TotN_conc ~ Month_1 + Month_2 + Month_3 + Month_5 + Month_6 +
##     Month_7 + Month_8 + Month_9 + Month_10 + Month_11
##
##            Df Sum of Sq    RSS    AIC
## - Month_1   1    1.3720 163.00 14.926
## - Month_3   1    1.4106 163.04 14.966
## <none>                 161.63 15.506
## - Month_11  1    2.7357 164.37 16.326
## - Month_5   1    2.7993 164.43 16.391
## - Month_2   1    3.8546 165.49 17.465
## - Month_10  1    3.9098 165.54 17.521
## - Month_6   1    8.6824 170.31 22.297
## - Month_9   1    9.1865 170.82 22.793
## - Month_7   1   16.4796 178.11 29.817
## - Month_8   1   19.6363 181.27 32.768
##
## Step:  AIC=14.93
## TotN_conc ~ Month_2 + Month_3 + Month_5 + Month_6 + Month_7 +
##     Month_8 + Month_9 + Month_10 + Month_11
##
##            Df Sum of Sq    RSS    AIC
## - Month_3   1    0.7151 163.72 13.661
## <none>                 163.00 14.926
## - Month_2   1    2.7831 165.78 15.770
## - Month_11  1    4.7022 167.70 17.704
## - Month_5   1    4.7906 167.79 17.792
## - Month_10  1    6.3071 169.31 19.304
## - Month_6   1   12.5278 175.53 25.366
## - Month_9   1   13.1690 176.17 25.979
## - Month_7   1   22.2773 185.28 34.447
## - Month_8   1   26.1552 189.16 37.927
##
## Step:  AIC=13.66
## TotN_conc ~ Month_2 + Month_5 + Month_6 + Month_7 + Month_8 +
##     Month_9 + Month_10 + Month_11
##
##            Df Sum of Sq    RSS    AIC
## <none>                 163.72 13.661
## - Month_2   1    2.2640 165.98 13.969
## - Month_11  1    6.0413 169.76 17.749
## - Month_5   1    6.1447 169.86 17.852
## - Month_10  1    7.9079 171.62 19.586
```

```
## - Month_6   1   15.0069 178.72 26.396
## - Month_9   1   15.7313 179.45 27.075
## - Month_7   1   25.9388 189.66 36.370
## - Month_8   1   30.2531 193.97 40.148
```

```r
colnames(lm_model_month_lag_step$model)
```

```
## [1] "TotN_conc" "Month_2"   "Month_5"   "Month_6"   "Month_7"   "Month_8"
## [7] "Month_9"   "Month_10"  "Month_11"
```

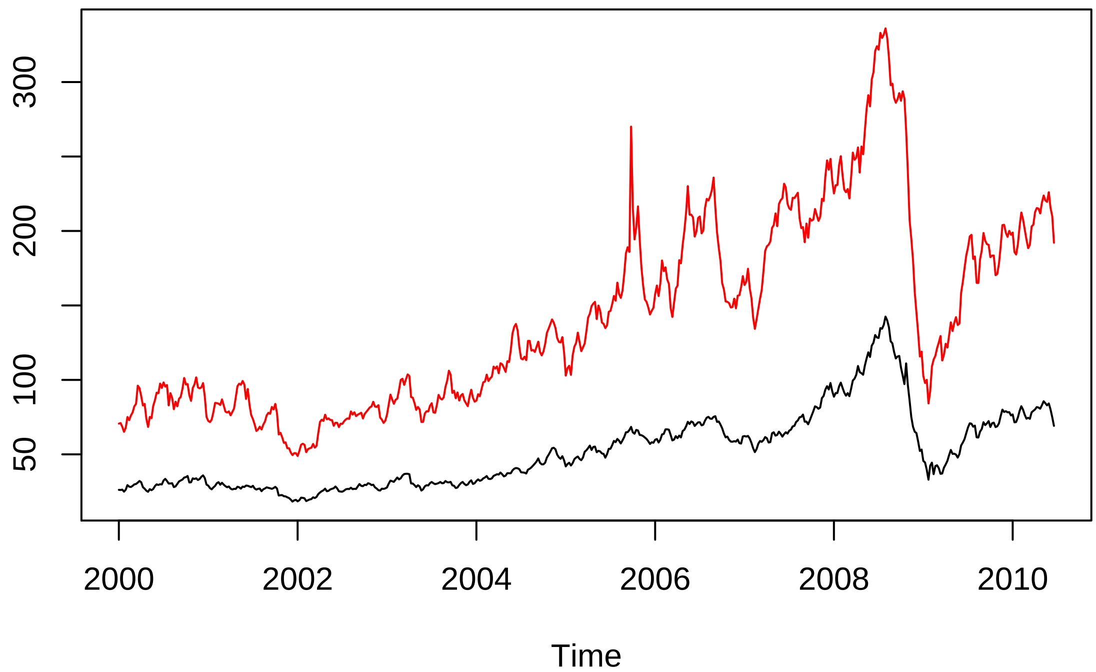
Analysis: The final terms in the model are given above, this model had the least AIC.

Assignment 3. Analysis of oil and gas time series.

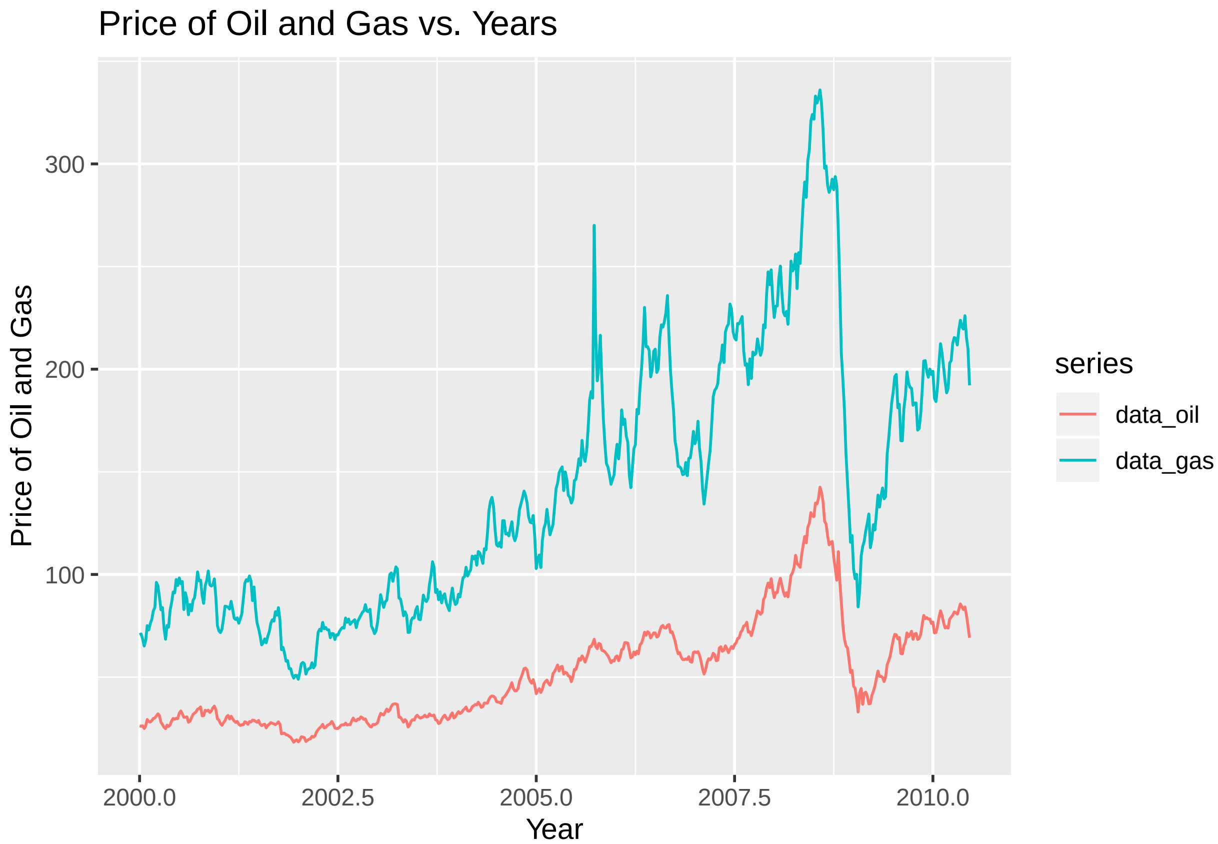
Weekly time series oil and gas present in the package astsa show the oil prices in dollars per barrel and gas prices in cents per dollar.

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? Motivate your answer.

```r
set.seed(12345)

data_oil <- astsa::oil
data_gas <- astsa::gas

ts.plot(data_oil, data_gas, gpars = list(col = c("black", "red")))
```

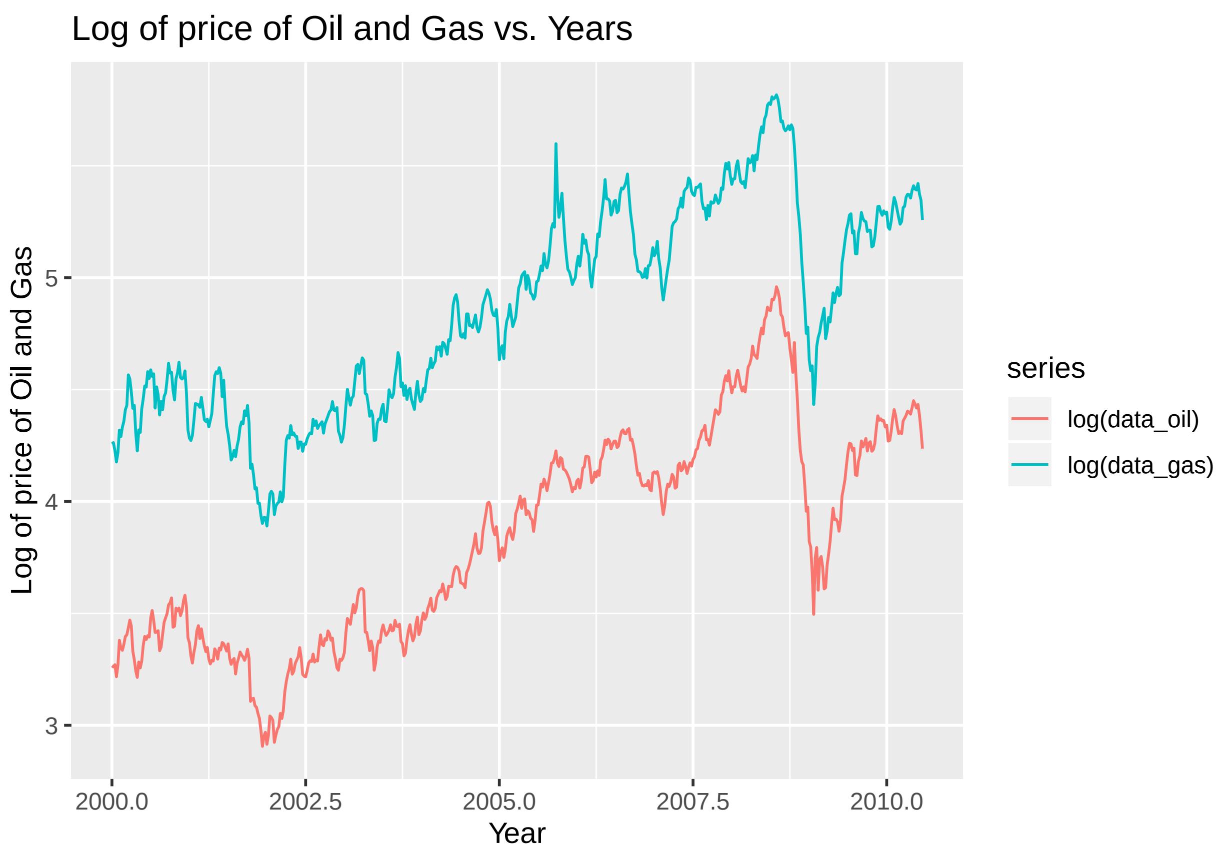



```
#alternative

autoplot(ts(cbind(data_oil, data_gas), start = 2000, frequency = 52)) +
          ylab("Price of Oil and Gas") +xlab("Year") +
          ggtitle("Price of Oil and Gas vs. Years")
```



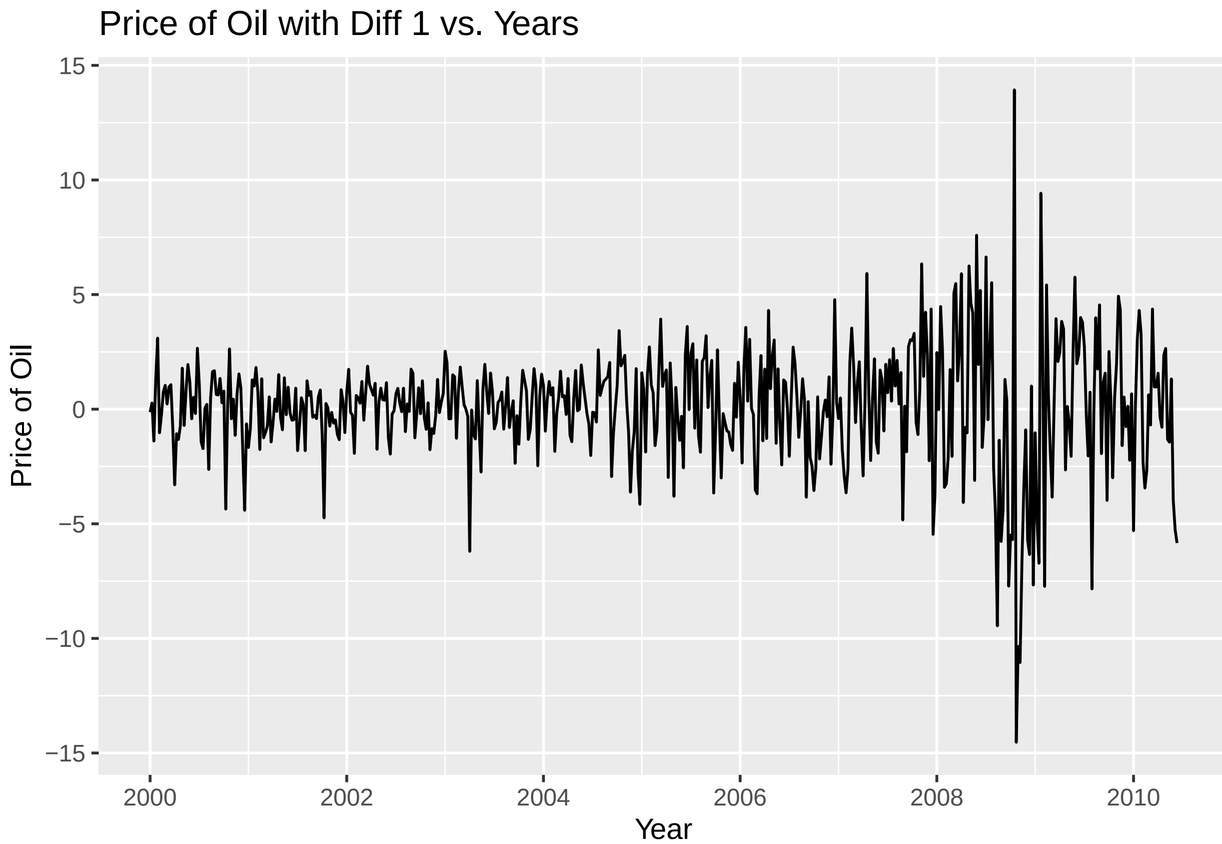
b) Apply log-transform to the time series and plot the transformed data. In what respect did this transformation made the data easier for the analysis?

```
set.seed(12345)

autoplot(ts(cbind(log(data_oil), log(data_gas)), start = 2000, frequency = 52)) +
           ylab("Log of price of Oil and Gas") +xlab("Year") +
           ggtitle("Log of price of Oil and Gas vs. Years")
```

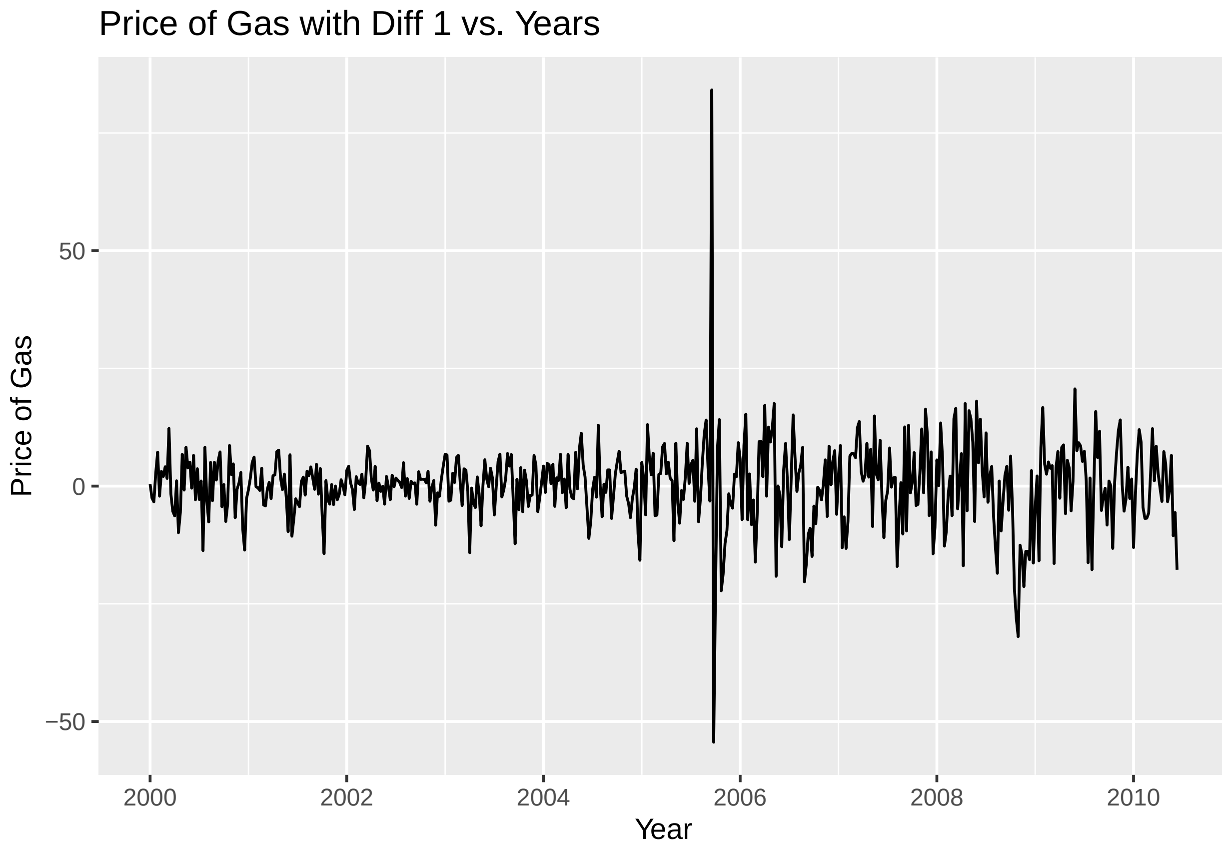


c) To eliminate trend, compute the first difference of the transformed data, plot the detrended series, check their ACFs and analyze the obtained plots. Denote the data obtained here as x_t(oil) and y_t(gas).

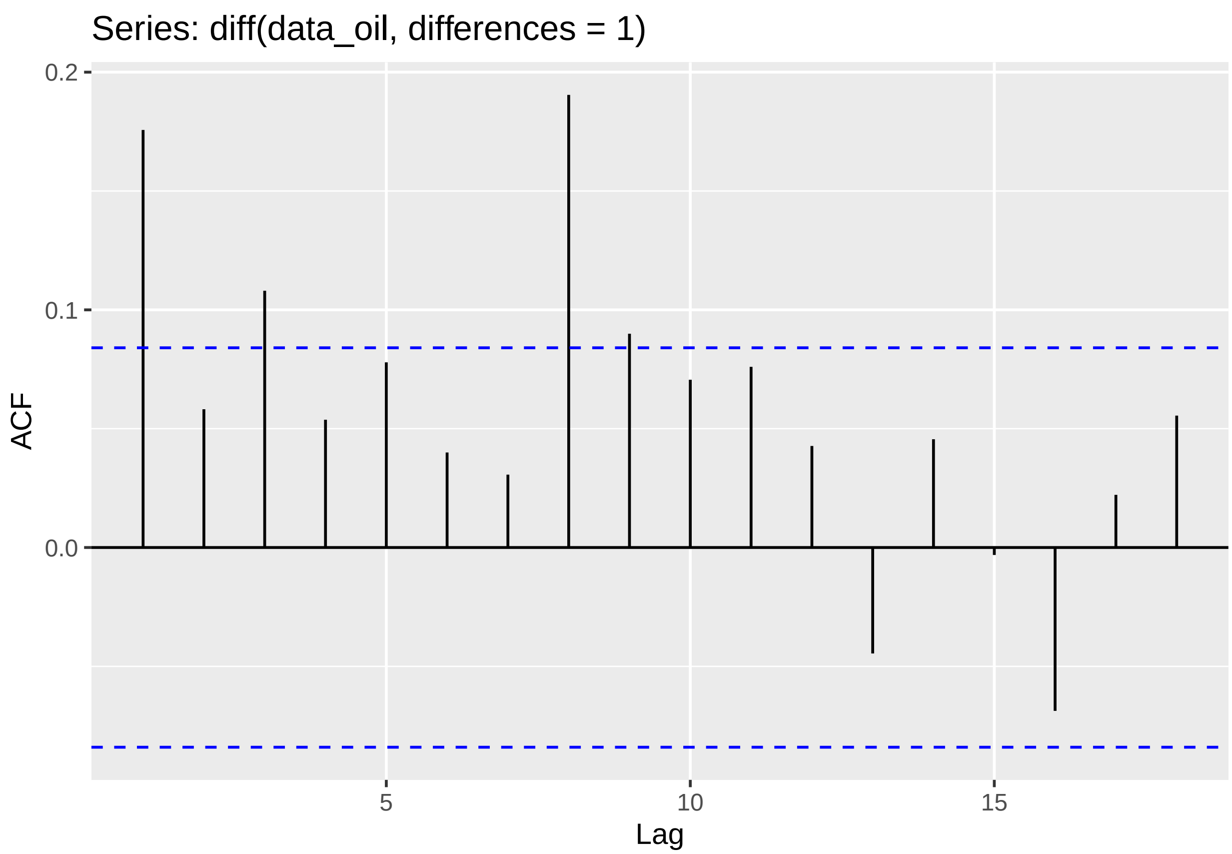
```
set.seed(12345)

autoplot(ts(diff(data_oil, differences = 1), start = 2000, frequency = 52)) +
           ylab("Price of Oil") +xlab("Year") +
           ggtitle("Price of Oil with Diff 1 vs. Years")
```



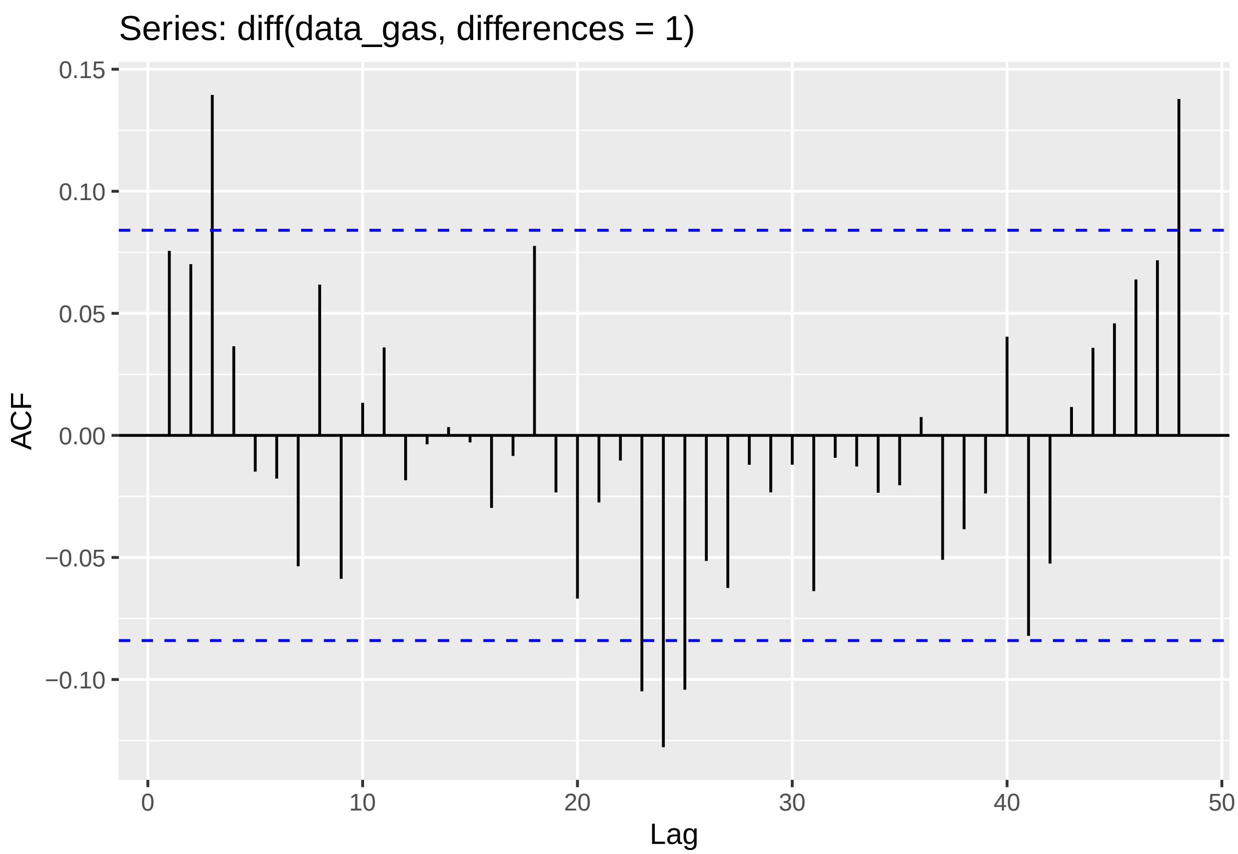

```r
autoplot(ts(diff(data_gas, differences = 1), start = 2000, frequency = 52)) +
           ylab("Price of Gas") +xlab("Year") +
           ggtitle("Price of Gas with Diff 1 vs. Years")
```




```
ggAcf(diff(data_oil, differences = 1), data_oil)
```



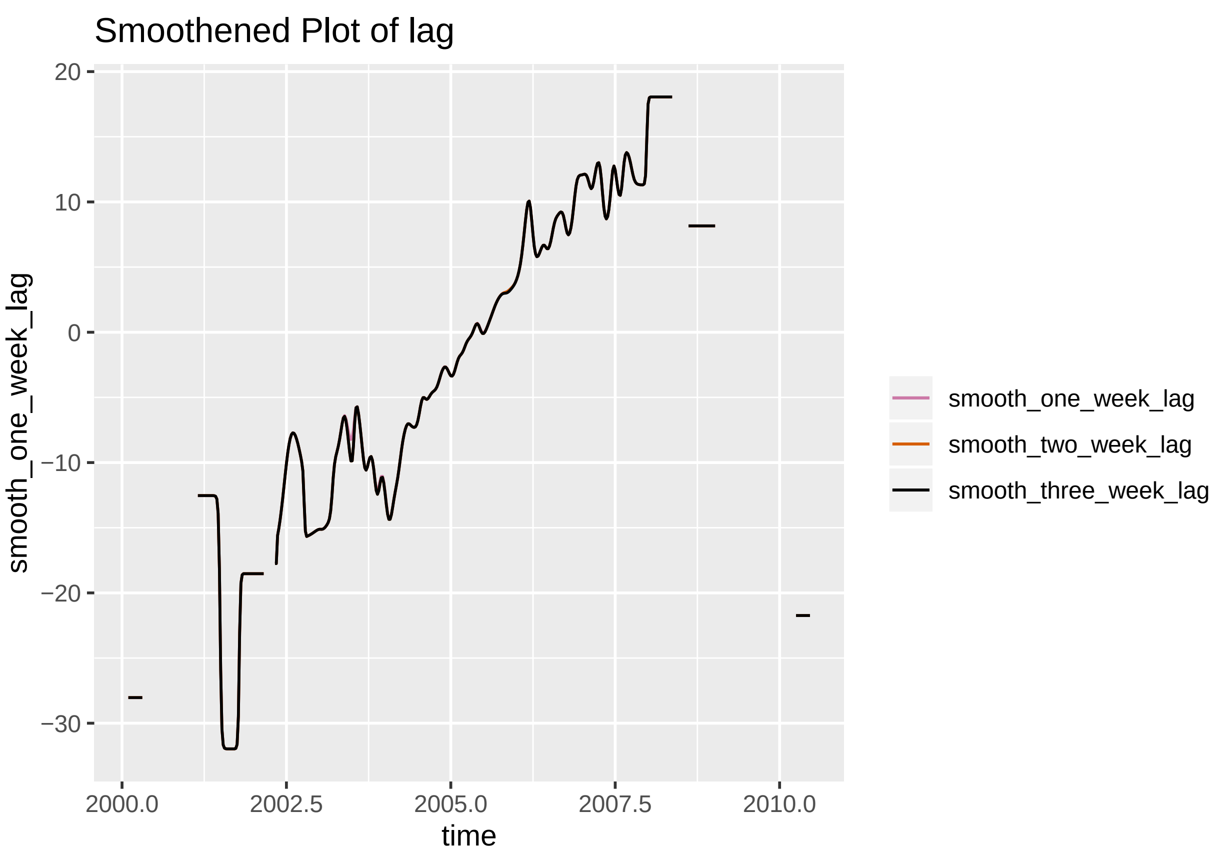

```r
ggAcf(diff(data_gas, differences = 1), data_gas)
```



d) Exhibit scatterplots of x_t and y_t for up to three weeks of lead time of x_t include a nonparametric smoother in each plot and comment the results: are there outliers? Are the relationships linear? Are there changes in the trend?

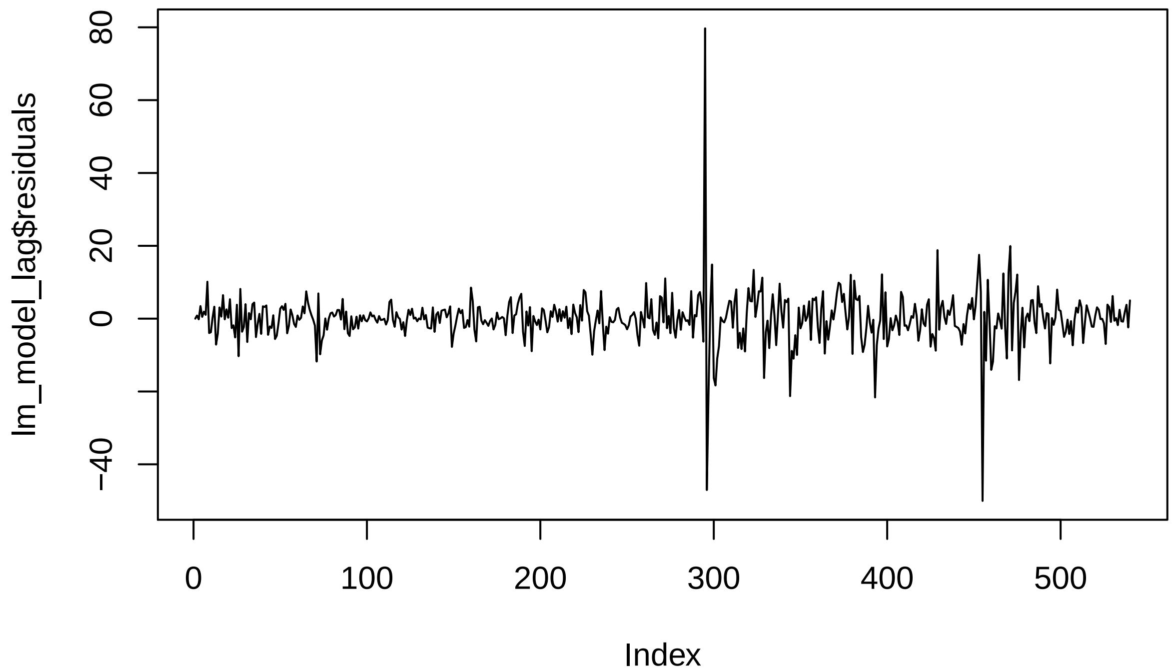
```
set.seed(12345)

oil_price_one_diff <- diff(data_oil, differences = 1)
gas_price_one_diff <- diff(data_gas, differences = 1)

df <- data.frame(oil_price_one_diff=as.matrix(oil_price_one_diff),
           gas_price_one_diff = as.matrix(gas_price_one_diff),
                     time=time(oil_price_one_diff))

df <- na.omit(df)


df$gas_price_one_diff = lag(df$gas_price_one_diff,1)
df$gas_price_two_diff = lag(df$gas_price_one_diff,2)
df$gas_price_three_diff = lag(df$gas_price_one_diff,3)
df$oil_price_one_diff = lag(df$oil_price_one_diff,1)
df$oil_price_two_diff = lag(df$oil_price_one_diff,2)
df$oil_price_three_diff = lag(df$oil_price_one_diff,3)

df <- na.omit(df)

df$smooth_one_week_lag <- ksmooth(x = df$oil_price_one_diff, y = df$gas_price_one_diff, bandwidth = 0.4
```

```
df$smooth_two_week_lag <- ksmooth(x = df$oil_price_two_diff, y = df$gas_price_two_diff, bandwidth = 0.4
df$smooth_three_week_lag <- ksmooth(x = df$oil_price_three_diff, y = df$gas_price_three_diff, bandwidth


ggplot(data=df, aes(x=time)) +
  geom_line(aes(y= smooth_one_week_lag, color= "smooth_one_week_lag")) +
    geom_line(aes(y= smooth_two_week_lag, color= "smooth_two_week_lag")) +
    geom_line(aes(y= smooth_three_week_lag, color= "smooth_three_week_lag")) +
      scale_colour_manual("", breaks = c("smooth_one_week_lag", "smooth_two_week_lag", "smooth_three_we
                           values = c("#CC79A7", "#000000", "#D55E00")) +
  ggtitle("Smoothened Plot of lag")
```

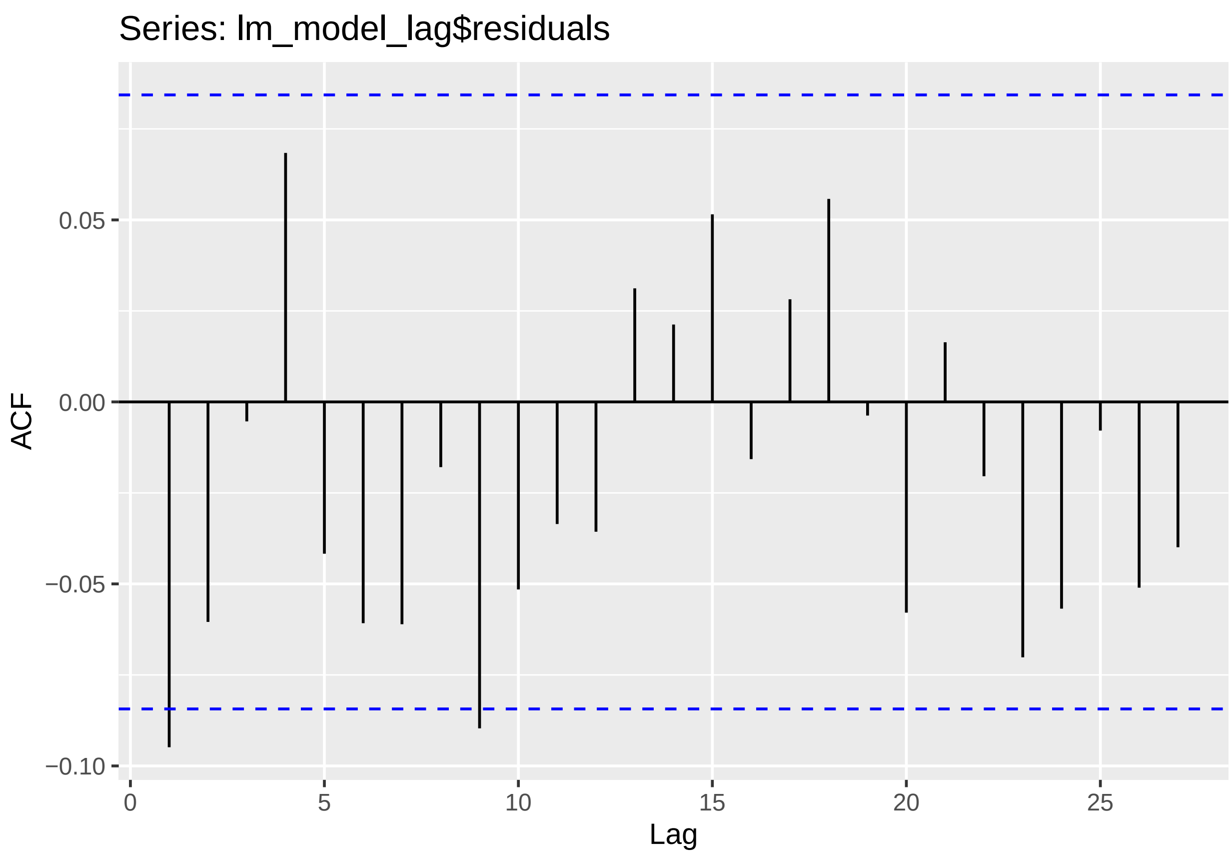


e) Fit the following model: $y_t = \alpha_0 + \alpha_1 I(x_t > 0) + \beta_1 x_1 + \beta_2 x_{t-1} + w_t$ and check which coefficients seem to be significant. How can this be interpreted? Analyze the residual pattern and the ACF of the residuals.

```
set.seed(12345)

df$x_t_more_zero <- ifelse(df$oil_price_one_diff>0,"1","0")
lm_model_lag <- lm(data=df, formula = gas_price_one_diff~oil_price_one_diff+oil_price_two_diff)

plot(lm_model_lag$residuals, type = 'l')
```




```r
ggAcf(lm_model_lag$residuals)
```



Appendix

```
knitr::opts_chunk$set(echo = TRUE)
options(scipen=999)

library("tidyverse") #ggplot and dplyr
library("gridExtra") # combine plots
library("knitr") # for pdf
library("fpp2") #timeseries with autoplot and stuff
library("reshape2") #reshape the data
library("MASS") #StepAIC
library("astsa") #dataset oil and gas is present here
library("zoo") #dataset oil and gas is present here


# The palette with black:
cbbPalette <- c("#000000", "#E69F00", "#56B4E9", "#009E73",
                "#F0E442", "#0072B2", "#D55E00", "#CC79A7")
set.seed(12345)
set.seed(12345)

n = 100
x <- vector(length = n)
```

```r
x2 <- vector(length = n)

x[1] <- 0
x[2] <- 0

#first series generation
for(i in 3:n){
  x[i] <- -0.8 * x[i-2] + rnorm(1,0,1)
}

#second series generation
for(i in 1:n){
  x2[i] <- cos(0.4*pi*i)
}

# smoothing filter function
smoothing_filter <- function(x){
v <- vector(length = length(x))
for(i in 5:length(x)){
  v[i] = 0.2 * (x[i] + x[i-1] + x[i-2] + x[i-3] + x[i-4])
}
return(v)
}

#generate smoothed series
smooth_x <- smoothing_filter(x)
smooth_x2 <- smoothing_filter(x2)

#adding everything to a dataframe
df <- cbind(x,x2,smooth_x,smooth_x2) %>% as.data.frame() %>% mutate(index=1:100)

ggplot(df, aes(x=index)) +
  geom_line(aes(y=x, color="Original Time Series")) +
  geom_line(aes(y=smooth_x, color="Smoothened Time Series")) +
  ggtitle("Plot of 1st time series and its smoothened version") +
    scale_colour_manual("", breaks = c("Original Time Series", "Smoothened Time Series"),
                        values = c("#CC79A7", "#000000"))

ggplot(df, aes(x=index)) +
  geom_line(aes(y=x2, color="Original Time Series")) +
  geom_line(aes(y=smooth_x2, color="Smoothened Time Series")) +
  ggtitle("Plot of 2nd time series and its smoothened version") +
    scale_colour_manual("", breaks = c("Original Time Series", "Smoothened Time Series"),
                        values = c("#CC79A7", "#000000"))

z = c(1,-4,2,0,0,1)
polyroot(z)
any(Mod(polyroot(z))<=1)
z = c(1,0,3,0,1,0,-4)
polyroot(z)
any(Mod(polyroot(z))<=1)
set.seed(54321)
```

```r
series <- arima.sim(n = 100, list(ar = c(-3/4), ma = c(0,-1/9)))

acf(series)
acf(ARMAacf(ar = c(-3/4), ma =  c(0,-1/9), lag.max = 20))
set.seed(12345)
rhine_data <- read.csv2("Rhine.csv")
rhine_data_ts <- ts(data = rhine_data$TotN_conc,
                    start = c(1989,1),
                    frequency = 12)

plot.ts(rhine_data_ts)
lag.plot(rhine_data_ts,lags = 12)
acf(rhine_data_ts)

#alternative
autoplot(rhine_data_ts) + ylab("Total Concentration") +xlab("Year") +
  ggtitle("Concentration in Rhine vs. Year")

gglagplot(rhine_data_ts, lags = 1, set.lags = 1:12, color=FALSE)
ggAcf(rhine_data_ts)

set.seed(12345)

rhine_lm_model <- lm(TotN_conc~Time, data=rhine_data)
plot(rhine_lm_model$residuals, type = 'l')
acf(rhine_lm_model$residuals)

set.seed(12345)

model_smooth_lag_5 <- ksmooth(x = rhine_data$Time, y = rhine_data$TotN_conc, bandwidth=5)
model_smooth_lag_10 <- ksmooth(x = rhine_data$Time, y = rhine_data$TotN_conc, bandwidth=10)
model_smooth_lag_20 <- ksmooth(x = rhine_data$Time, y = rhine_data$TotN_conc, bandwidth=20)

model_smooth_lag_5_residual <- rhine_data$TotN_conc - model_smooth_lag_5$y
model_smooth_lag_10_residual <- rhine_data$TotN_conc - model_smooth_lag_10$y
model_smooth_lag_20_residual <- rhine_data$TotN_conc - model_smooth_lag_20$y

residual_df <- cbind(model_smooth_lag_5_residual, model_smooth_lag_10_residual,
                     model_smooth_lag_20_residual, rhine_data$Time) %>% as.data.frame()

colnames(residual_df) <- c("lag_5_residual", "lag_10_residual", "lag_20_residual", "Time")

ggplot(residual_df, aes(x=Time)) +
  geom_line(aes(y=lag_5_residual, color="Lag 5 residual")) +
  geom_line(aes(y=lag_10_residual, color="Lag 10 residual")) +
  geom_line(aes(y=lag_20_residual, color="Lag 20 residual")) +
  ggtitle("Residual vs. Time by Lag") +
    scale_colour_manual("", breaks = c("Lag 5 residual", "Lag 10 residual", "Lag 20 residual"),
                        values = c("#CC79A7", "#000000", "#D55E00"))


acf(model_smooth_lag_5_residual)
acf(model_smooth_lag_10_residual)
```

```
acf(model_smooth_lag_20_residual)
set.seed(12345)

rhine_data_wide <- rhine_data
rhine_data_wide$dummy <- "1"
rhine_data_wide$Month <- paste0("Month_",rhine_data_wide$Month)
rhine_data_wide <- dcast(rhine_data_wide, formula = TotN_conc+Year+Time~Month, value.var = "dummy", fil

lm_model_month_lag <- lm(data=rhine_data_wide,
                    TotN_conc~Month_1+Month_2+Month_3+Month_4+Month_5+Month_6+Month_7+
                      Month_8+Month_9+Month_10+Month_11+Month_12)


plot(lm_model_month_lag$residuals, type = 'l')
acf(lm_model_month_lag$residuals)

set.seed(12345)

lm_model_month_lag_step <- stepAIC(lm_model_month_lag, scope = list(upper = ~Month_1+Month_2+
                                                                  Month_3+Month_4+Month_5+Month_6+Month
                                                                    Month_8+Month_9+Month_10+Month_11+Mo
                                                                  lower = ~1), trace = TRUE,
                                   direction="backward")

colnames(lm_model_month_lag_step$model)
set.seed(12345)

data_oil <- astsa::oil
data_gas <- astsa::gas

ts.plot(data_oil, data_gas, gpars = list(col = c("black", "red")))

#alternative

autoplot(ts(cbind(data_oil, data_gas), start = 2000, frequency = 52)) +
           ylab("Price of Oil and Gas") +xlab("Year") +
           ggtitle("Price of Oil and Gas vs. Years")
set.seed(12345)

autoplot(ts(cbind(log(data_oil), log(data_gas)), start = 2000, frequency = 52)) +
           ylab("Log of price of Oil and Gas") +xlab("Year") +
           ggtitle("Log of price of Oil and Gas vs. Years")
set.seed(12345)

autoplot(ts(diff(data_oil, differences = 1), start = 2000, frequency = 52)) +
           ylab("Price of Oil") +xlab("Year") +
           ggtitle("Price of Oil with Diff 1 vs. Years")

autoplot(ts(diff(data_gas, differences = 1), start = 2000, frequency = 52)) +
           ylab("Price of Gas") +xlab("Year") +
           ggtitle("Price of Gas with Diff 1 vs. Years")

ggAcf(diff(data_oil, differences = 1), data_oil)
```

```
ggAcf(diff(data_gas, differences = 1), data_gas)

set.seed(12345)

oil_price_one_diff <- diff(data_oil, differences = 1)
gas_price_one_diff <- diff(data_gas, differences = 1)

df <- data.frame(oil_price_one_diff=as.matrix(oil_price_one_diff),
           gas_price_one_diff = as.matrix(gas_price_one_diff),
                      time=time(oil_price_one_diff))

df <- na.omit(df)


df$gas_price_one_diff = lag(df$gas_price_one_diff,1)
df$gas_price_two_diff = lag(df$gas_price_one_diff,2)
df$gas_price_three_diff = lag(df$gas_price_one_diff,3)
df$oil_price_one_diff = lag(df$oil_price_one_diff,1)
df$oil_price_two_diff = lag(df$oil_price_one_diff,2)
df$oil_price_three_diff = lag(df$oil_price_one_diff,3)

df <- na.omit(df)

df$smooth_one_week_lag <- ksmooth(x = df$oil_price_one_diff, y = df$gas_price_one_diff, bandwidth = 0.4
df$smooth_two_week_lag <- ksmooth(x = df$oil_price_two_diff, y = df$gas_price_two_diff, bandwidth = 0.4
df$smooth_three_week_lag <- ksmooth(x = df$oil_price_three_diff, y = df$gas_price_three_diff, bandwidth


ggplot(data=df, aes(x=time)) +
  geom_line(aes(y= smooth_one_week_lag, color= "smooth_one_week_lag")) +
    geom_line(aes(y= smooth_two_week_lag, color= "smooth_two_week_lag")) +
    geom_line(aes(y= smooth_three_week_lag, color= "smooth_three_week_lag")) +
      scale_colour_manual("", breaks = c("smooth_one_week_lag", "smooth_two_week_lag", "smooth_three_we
                        values = c("#CC79A7", "#000000", "#D55E00")) +
  ggtitle("Smoothened Plot of lag")

set.seed(12345)

df$x_t_more_zero <- ifelse(df$oil_price_one_diff>0,"1","0")
lm_model_lag <- lm(data=df, formula = gas_price_one_diff~oil_price_one_diff+oil_price_two_diff)

plot(lm_model_lag$residuals, type = 'l')
ggAcf(lm_model_lag$residuals)
```