

Time Series (732A62) Lab1

Anubhav Dikshit(anudi287) and Maximilian Pfundstein(maxpf364)

11 September, 2019

Contents

Assignment 1. Computations with simulated data	2
Assignment 2. Visualization, detrending and residual analysis of Rhine data.	6
Assignment 3. Analysis of oil and gas time series.	21
Appendix	33

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)
}

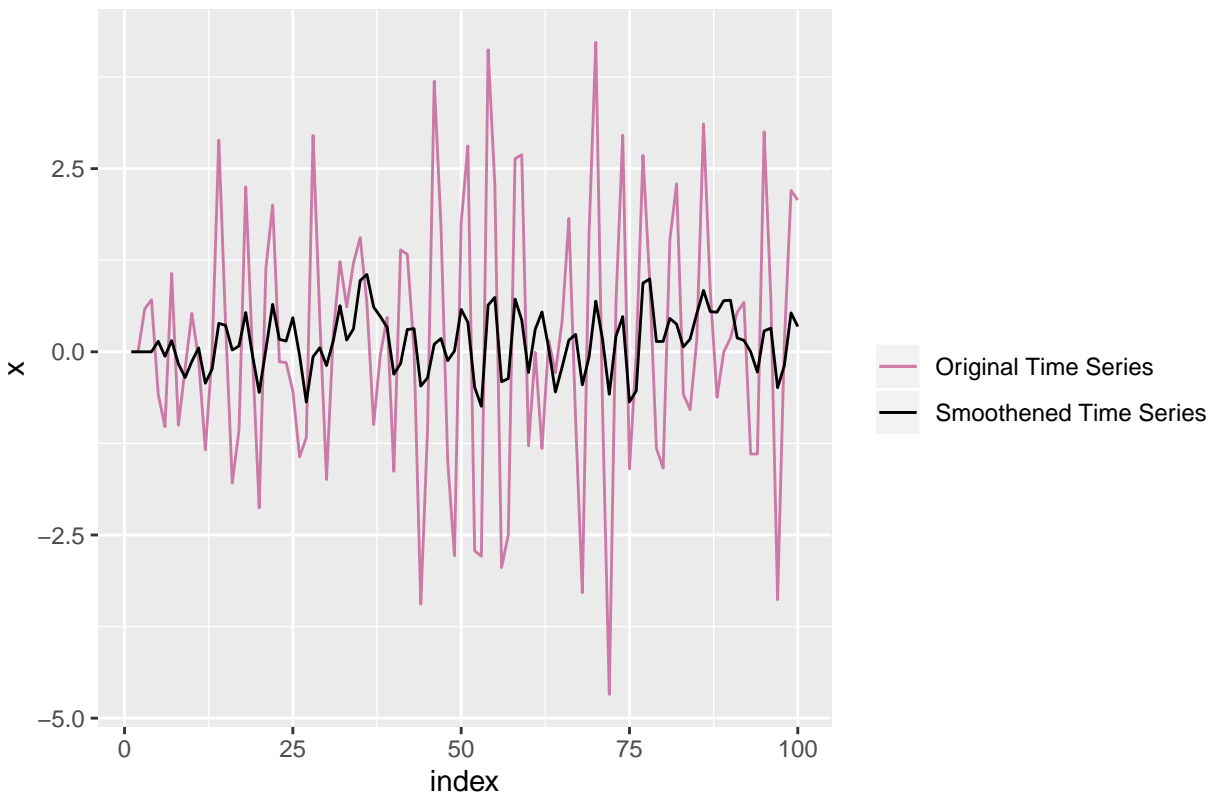
# smoothing filter function
smoothing_filter <- function(x){
  v <- vector(length = length(x))
  for(i in 5:length(x)){
    v[i] = 0.2 * (x[i] + x[i-1] + x[i-2] + x[i-3] + x[i-4])
  }
  return(v)
}

#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)

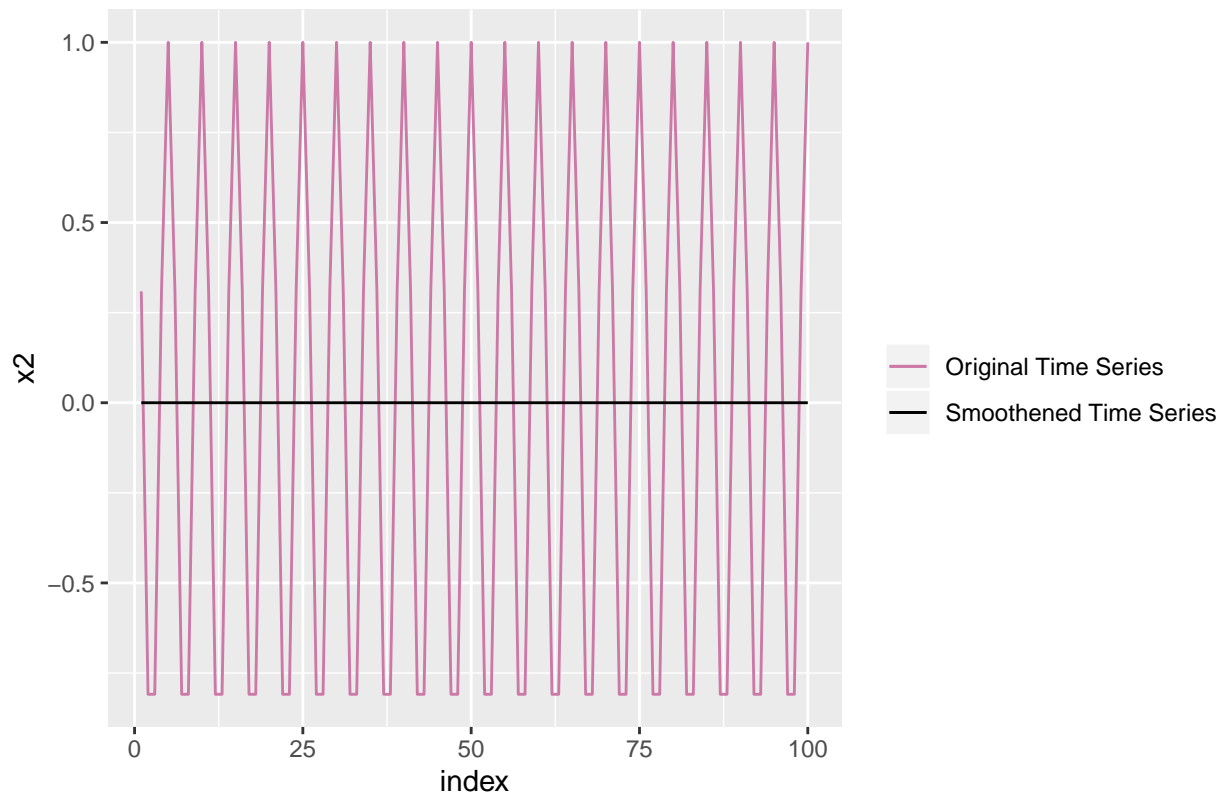
ggplot(df, aes(x=index)) +
  geom_line(aes(y=x, color="Original Time Series")) +
  geom_line(aes(y=smooth_x, color="Smoothened Time Series")) +
  ggtitle("Plot of 1st time series and its smoothened version") +
  scale_colour_manual("", breaks = c("Original Time Series", "Smoothened Time Series"),
  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

Plot of 1st time series and its smoothed version



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

Plot of 2nd time series and its smoothened version



Analysis: For the first time series the smoothing really helped the variation in the vertical axis(y-axis) has decreased, which is an expected effect of the time series smoothing. In the cosine case the smoothing results in a horizontal straight line, this is due to the fact that cosine is a periodic function and odd number of lags will always lead to a horizontal line.

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 $\rightarrow 1 - 0.4B + 0.3B^2$

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
```

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

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

Analysis: Baring one of the roots all are inside the unit circle. Thus the time series is neither invertiable or causal.

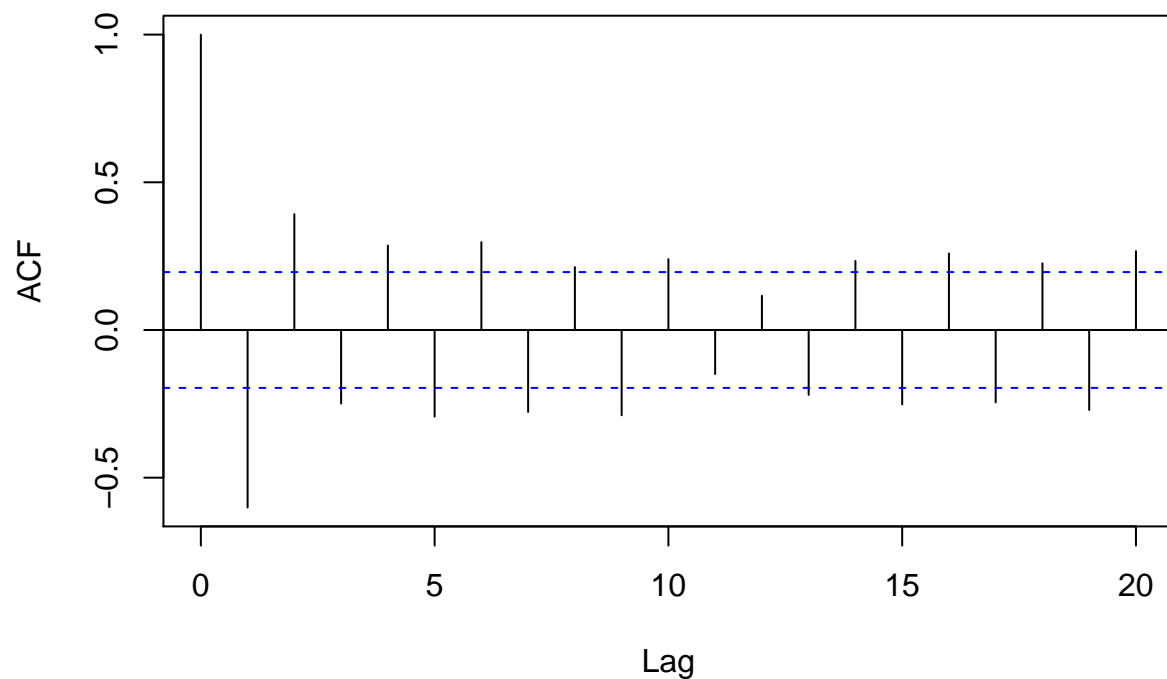
c) Use built-in R functions to simulate 100 observations from the process $x_t + \frac{3}{4}x_{t-1} = w_t - \frac{1}{9}w_{t-2}$ compute sample ACF and theoretical ACF, use seed 54321. Compare the ACF plots.

```
set.seed(54321)

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

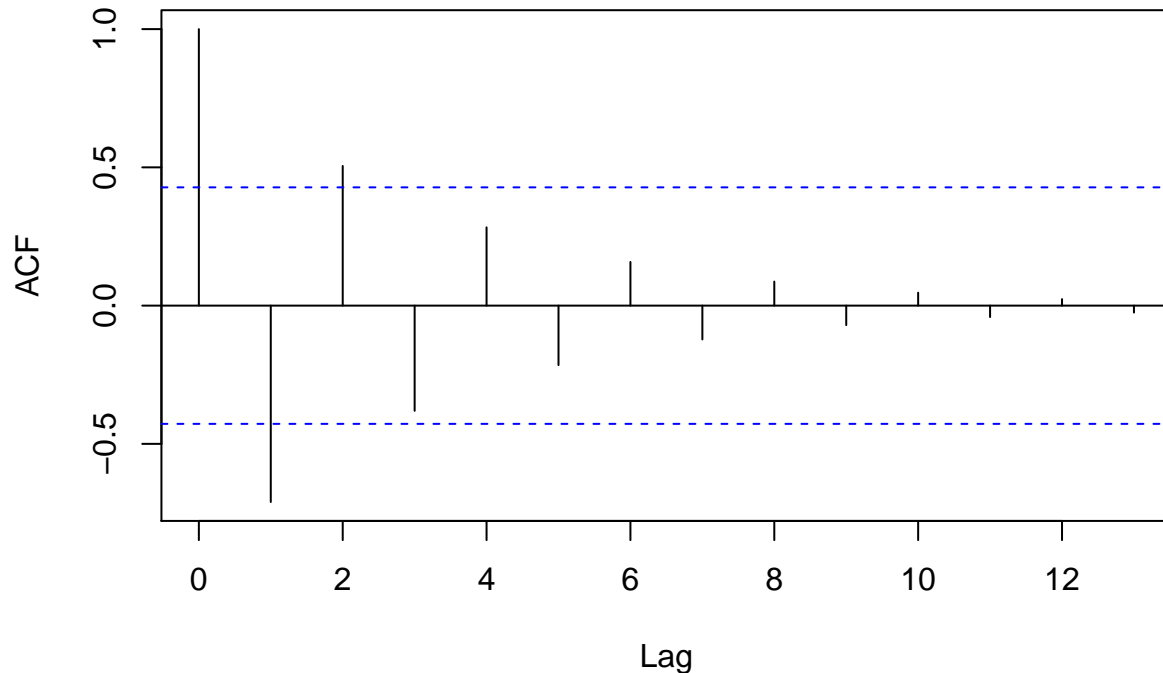
acf(series)
```

Series series



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

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



Analysis: In the theoretical 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.

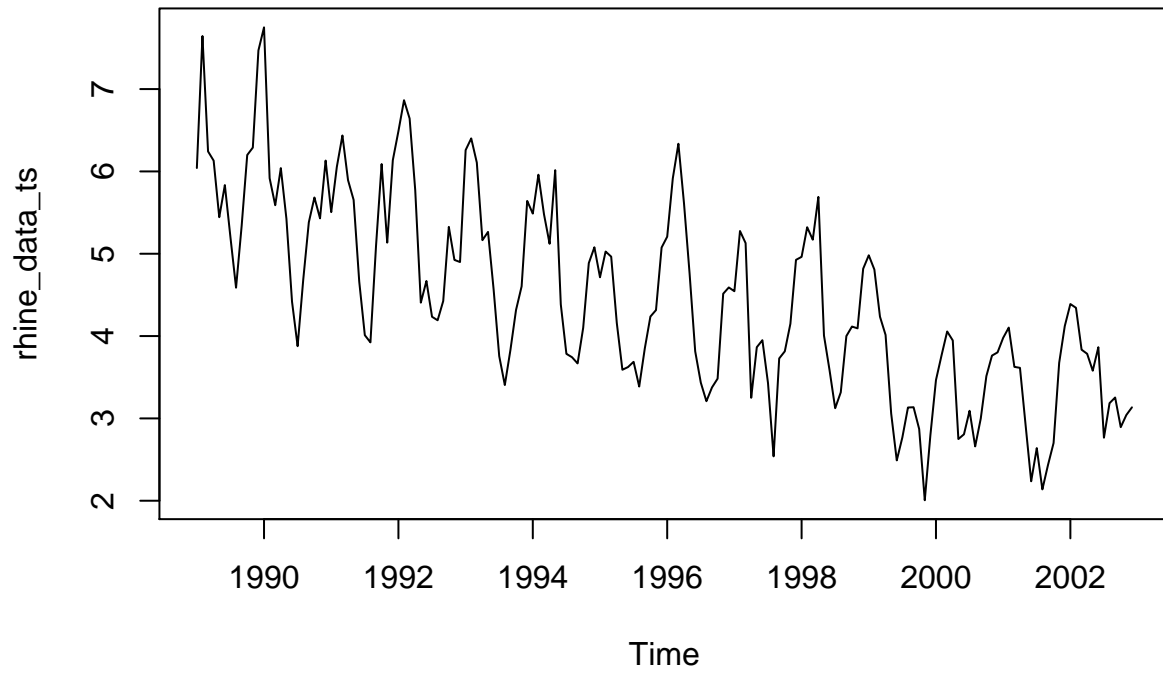
The data set Rhine.csv contains monthly concentrations of total nitrogen in the Rhine River in the period 1989-2002.

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}, \dots, 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?

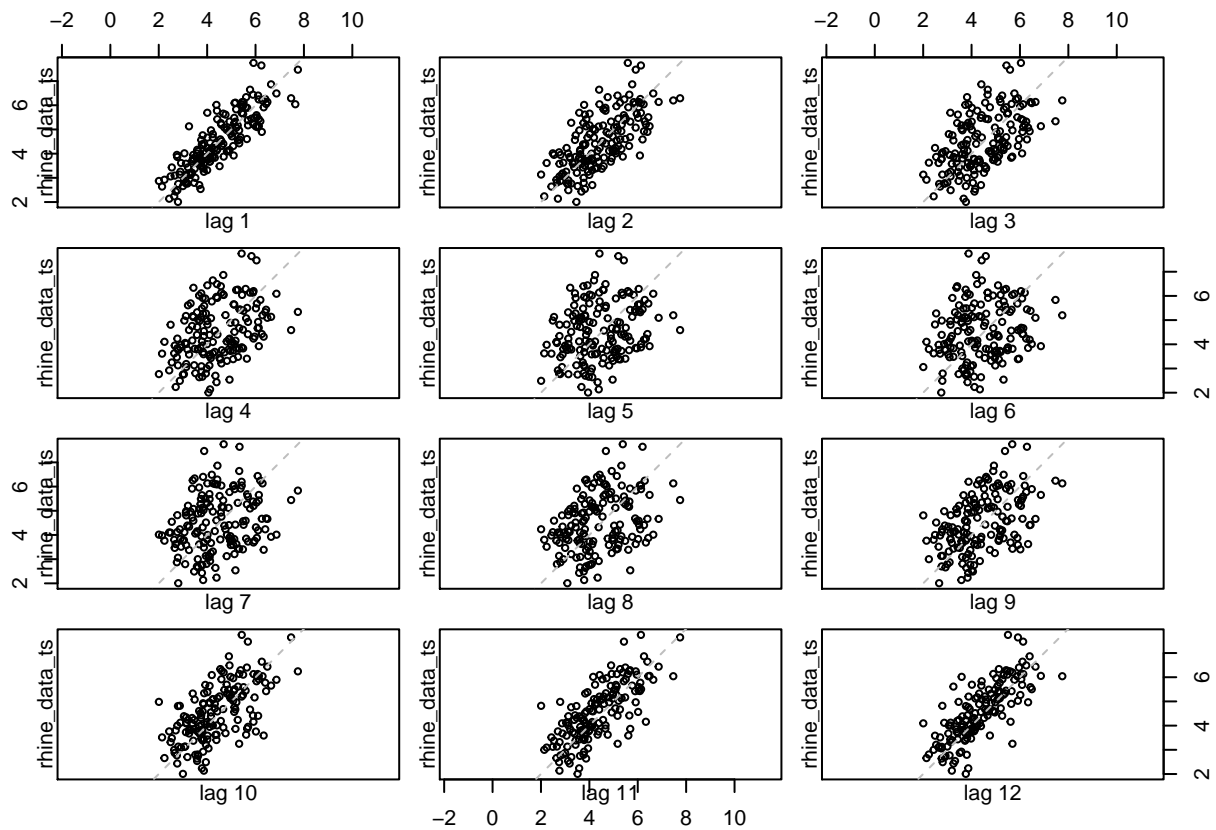
```
set.seed(12345)
rhine_data <- read.csv2("Rhine.csv")
rhine_data_ts <- ts(data = rhine_data$TotN_conc,
                    start = c(1989,1),
                    frequency = 12)

plot.ts(rhine_data_ts, main="Time Series of Nitrogen Concentration in Rhine")
```

Time Series of Nitrogen Concentration in Rhine

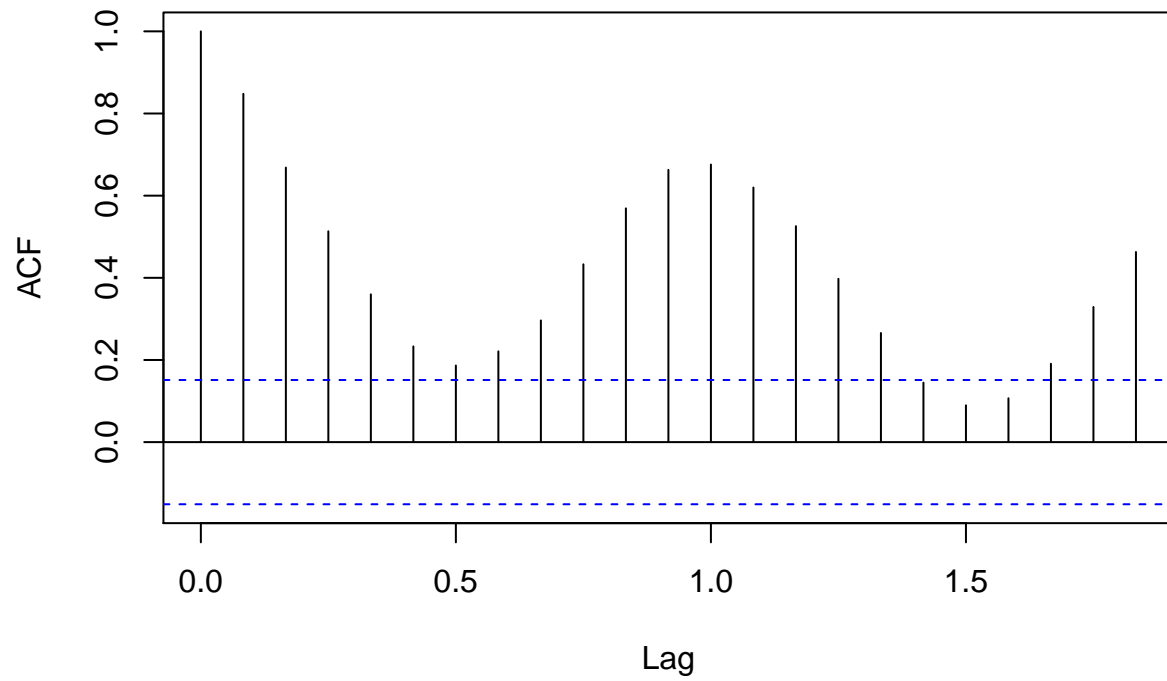


```
lag.plot(rhine_data_ts,lags = 12)
```

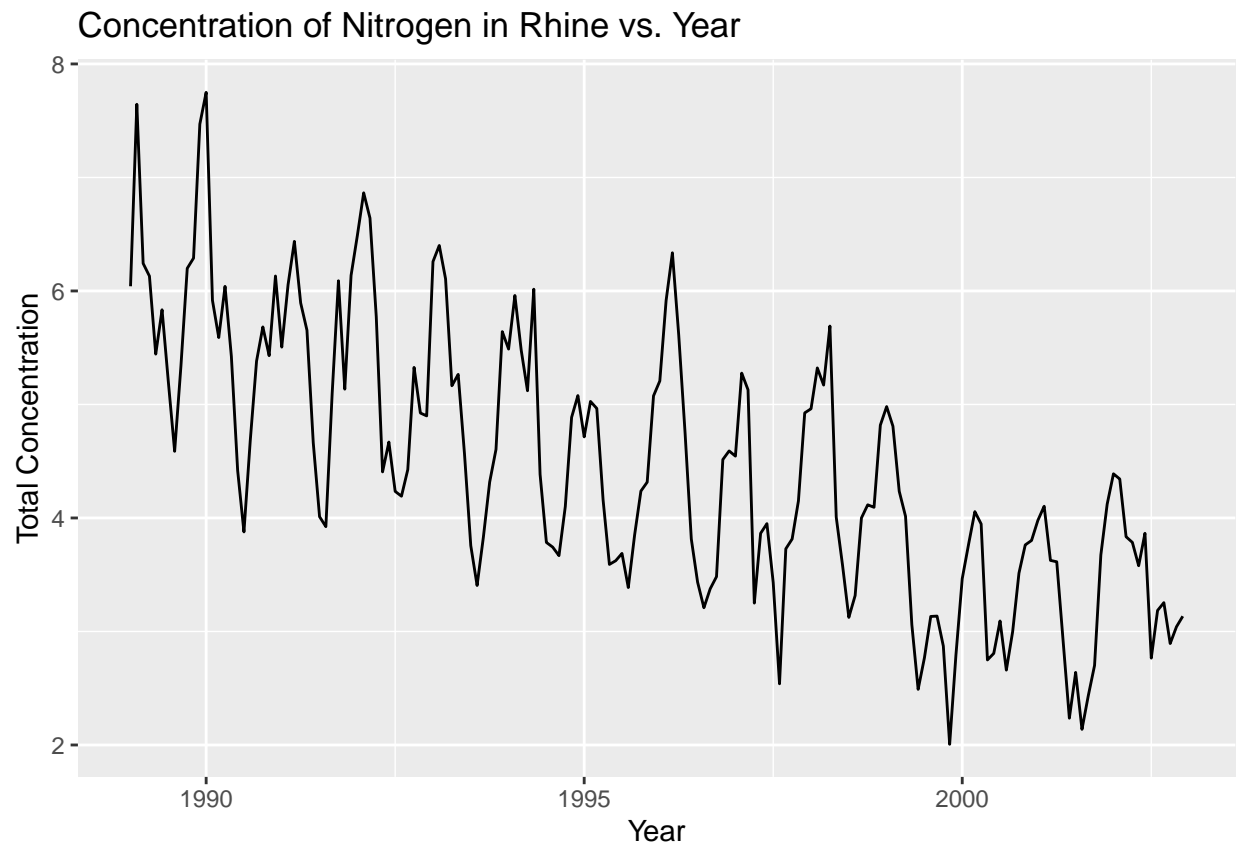


```
acf(rhine_data_ts)
```

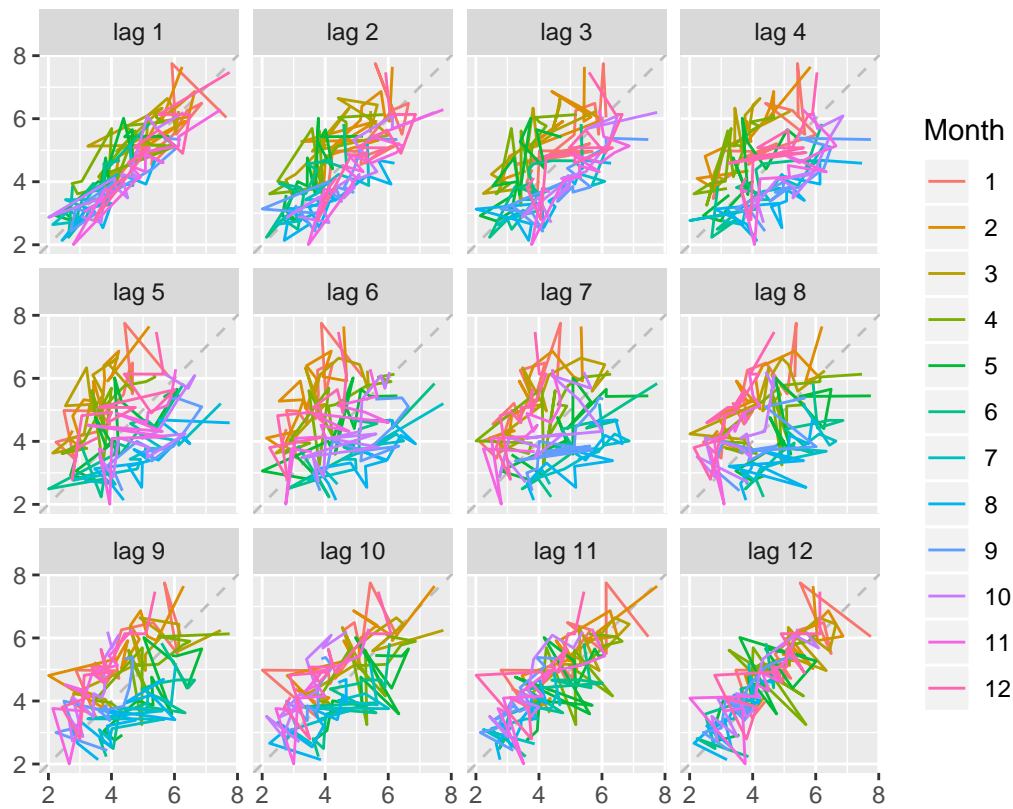

Series rhine_data_ts



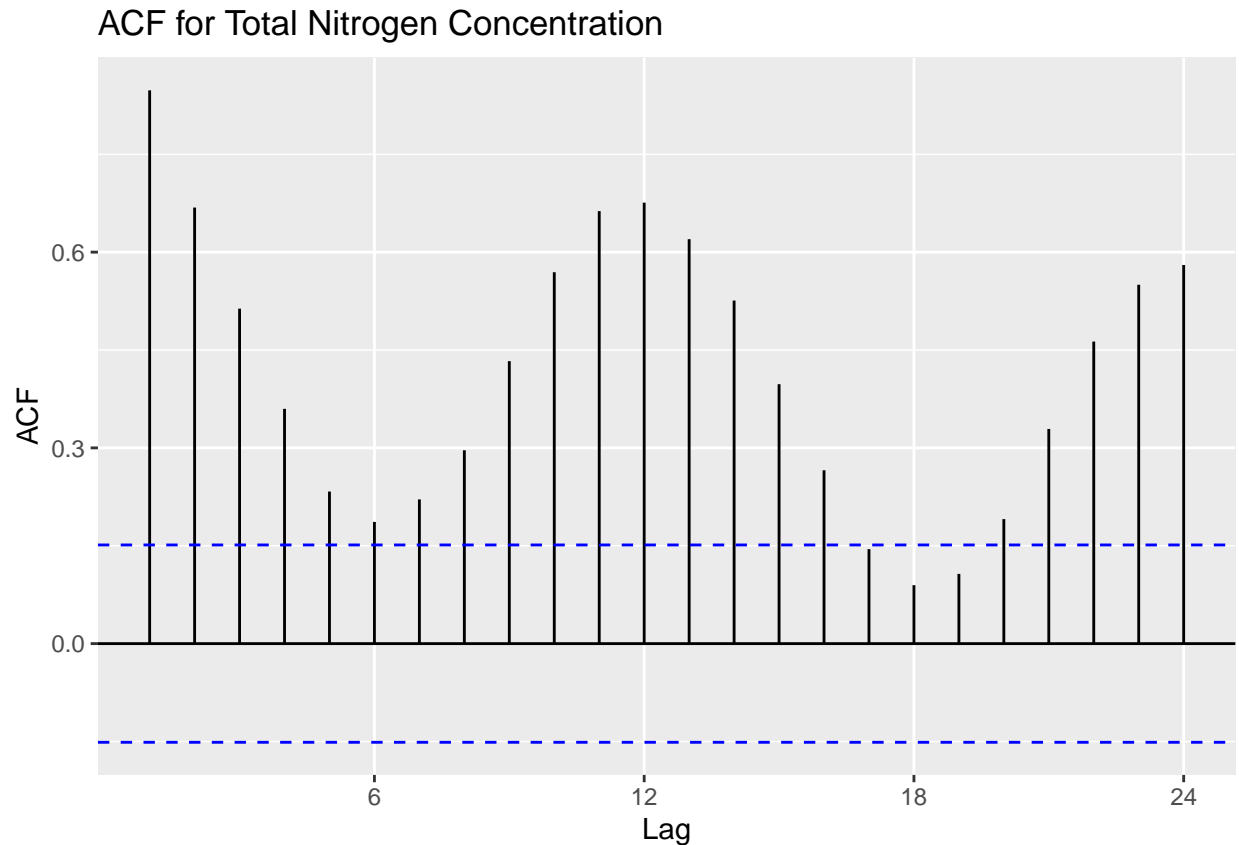
```
#alternative  
autoplot(rhine_data_ts) + ylab("Total Concentration") + xlab("Year") +  
  ggtitle("Concentration of Nitrogen in Rhine vs. Year")
```



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



```
ggAcf(rhine_data_ts) + ggtitle("ACF for Total Nitrogen Concentration")
```



Analysis: Its evident from the time series plot that there is a decreasing trend and there is a seasonality to the concentration of nitrogen.

From the scatter plots of lag concentration vs. time we can conclude that correlation between concentration and lag is least in the middle months of the years than the beginning and end months.

As from above we from ACF plots we can also conclude that the components of the end of the year and beginning of the year have more impact than the middle year components.

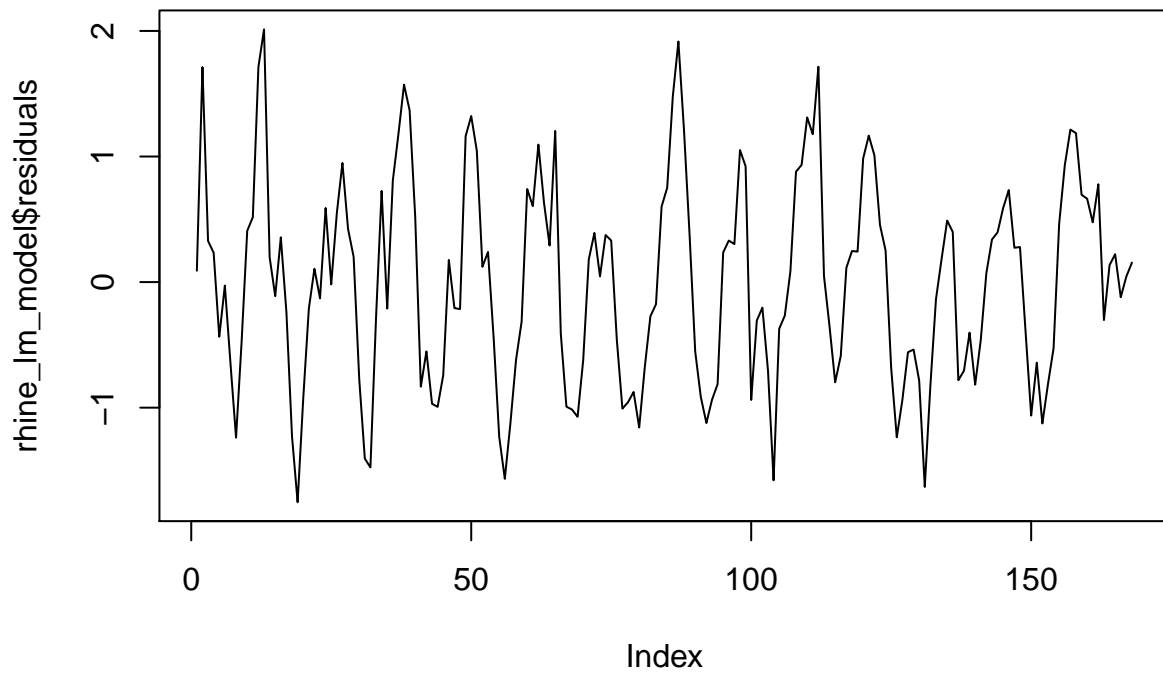
b) Eliminate the trend by fitting a linear model with respect to t to the time series. Is there a significant time trend? Look at the residual pattern and the sample ACF of the residuals and comment how this pattern might be related to seasonality of the series.

```
set.seed(12345)
```

```
rhine_lm_model <- lm(TotN_conc~Time, data=rhine_data)
```

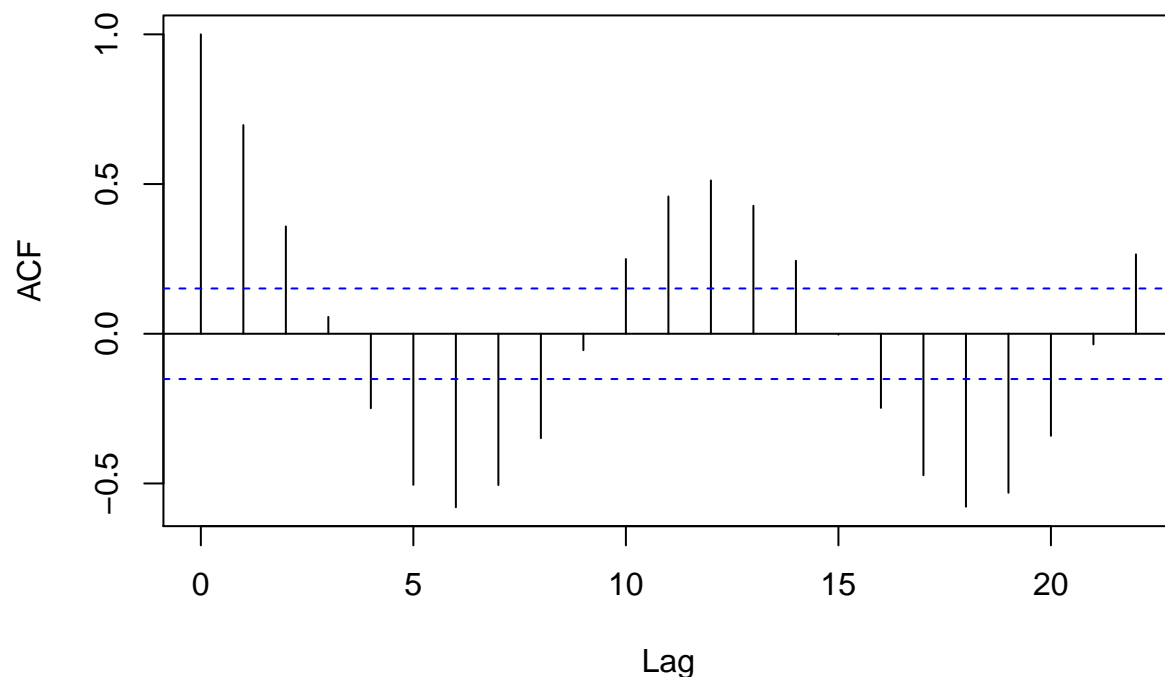
```
plot(rhine_lm_model$residuals, type = 'l', main="Plot of Residual from the linear model of Nitrogen Con
```

Plot of Residual from the linear model of Nitrogen Concentration



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

Series rhine_lm_model\$residuals



Analysis: The trend from the time series has been eliminated and the seasonality variation is limited to 1/2 units. From the ACF plot we also see that few components are needed to describe this compared to our previous approach.

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)

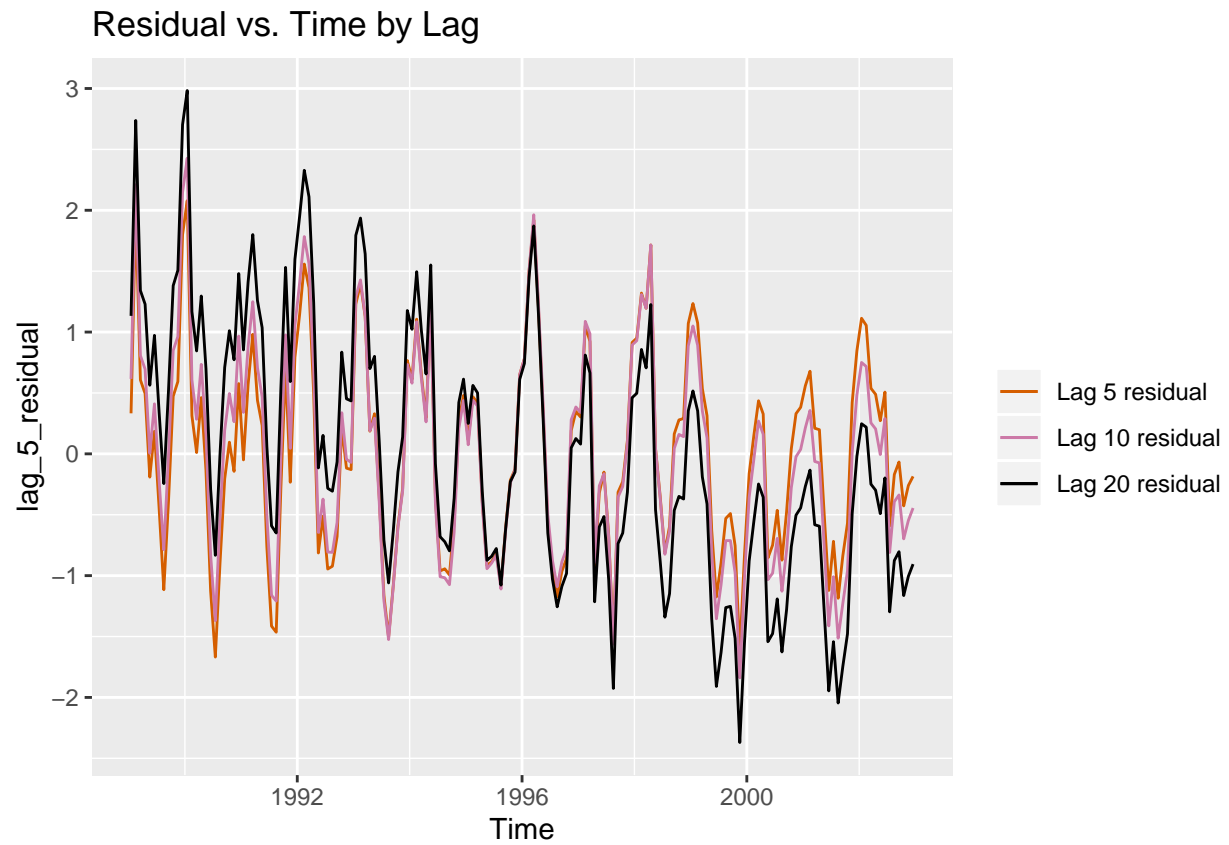
model_smooth_lag_5_resid <- rhine_data$TotN_conc - model_smooth_lag_5$y
model_smooth_lag_10_resid <- rhine_data$TotN_conc - model_smooth_lag_10$y
model_smooth_lag_20_resid <- rhine_data$TotN_conc - model_smooth_lag_20$y

residual_df <- cbind(model_smooth_lag_5_resid, model_smooth_lag_10_resid,
                    model_smooth_lag_20_resid, rhine_data$Time) %>% as.data.frame()

colnames(residual_df) <- c("lag_5_resid", "lag_10_resid", "lag_20_resid", "Time")

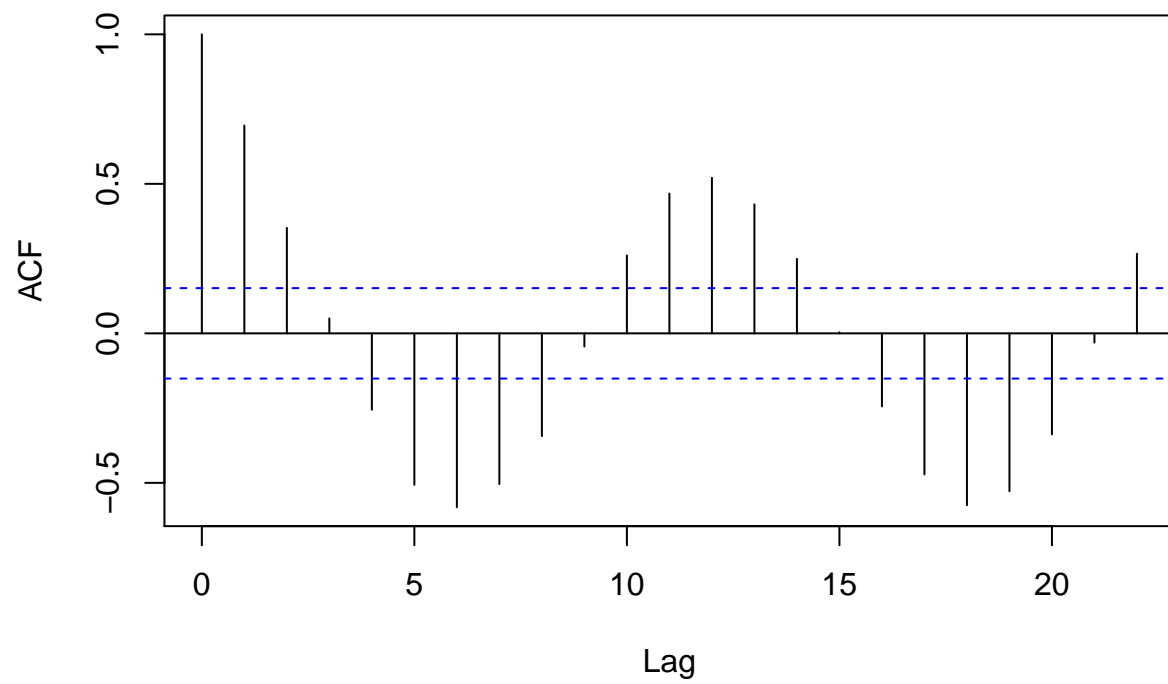
ggplot(residual_df, aes(x=Time)) +
  geom_line(aes(y=lag_5_resid, color="Lag 5 residual")) +
  geom_line(aes(y=lag_10_resid, color="Lag 10 residual")) +
```

```
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"))
```



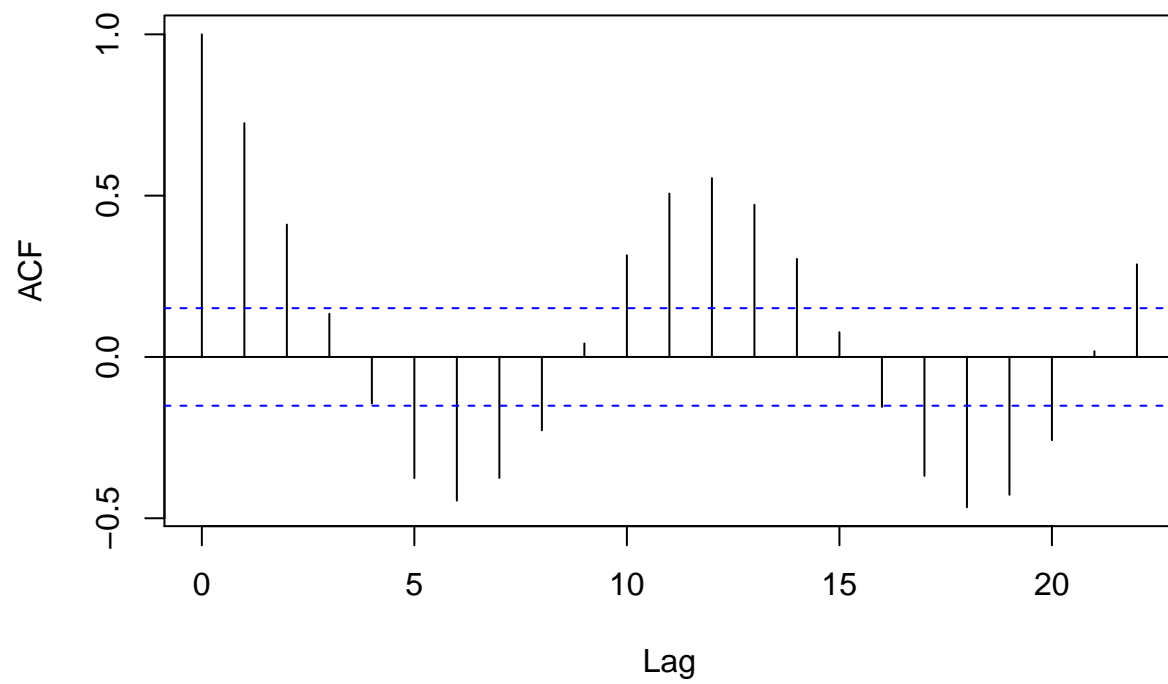
```
acf(model_smooth_lag_5_residual)
```

Series model_smooth_lag_5_residual



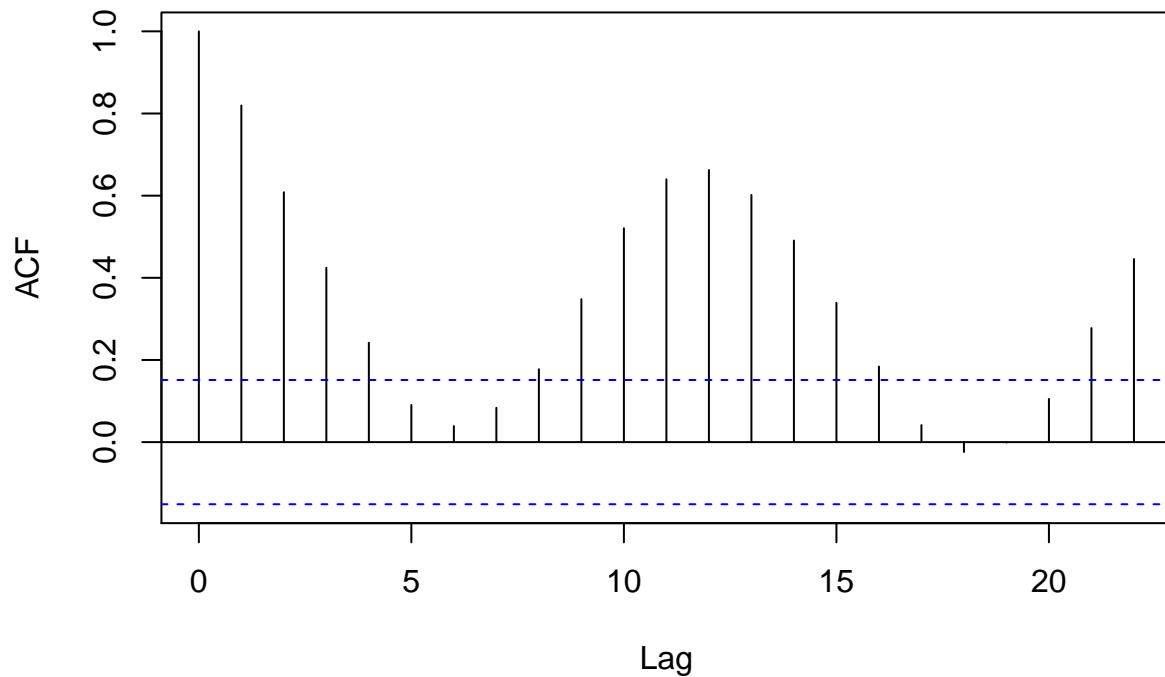
```
acf(model_smooth_lag_10_residual)
```


Series model_smooth_lag_10_residual



```
acf(model_smooth_lag_20_residual)
```

Series model_smooth_lag_20_residual



Analysis: In the above steps we have compared the residual pattern from the linear model, kernel smoother with bandwidth 4, 10 and bandwidth 20. Increasing the bandwidth does not seem to have much impact on the residual pattern and it is evident that it is not stationary, there is a large seasonality effect and a downwards trend still present in the plot.

d) Eliminate the trend by fitting the following so-called seasonal means model: $x_t = \alpha_0 + \alpha_1 t + \beta_1 I(\text{month} = 2) + \dots + \beta_{12} I(\text{month} = 12) + w_t$, 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.

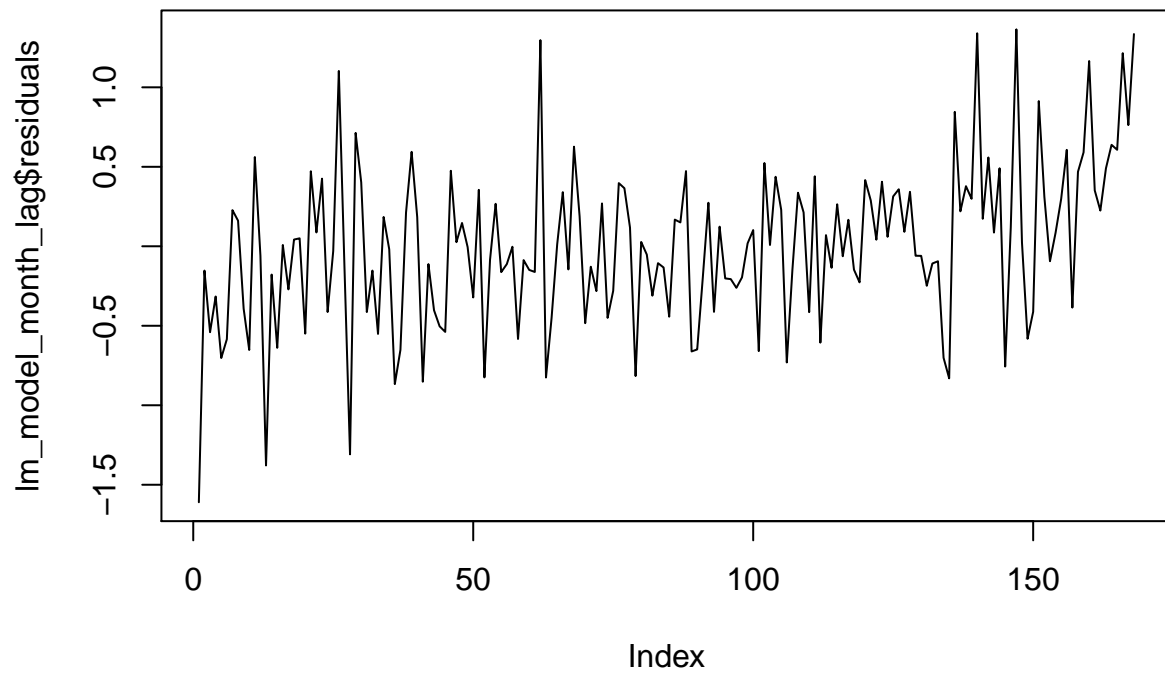
```
set.seed(12345)

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", fill = "0")

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

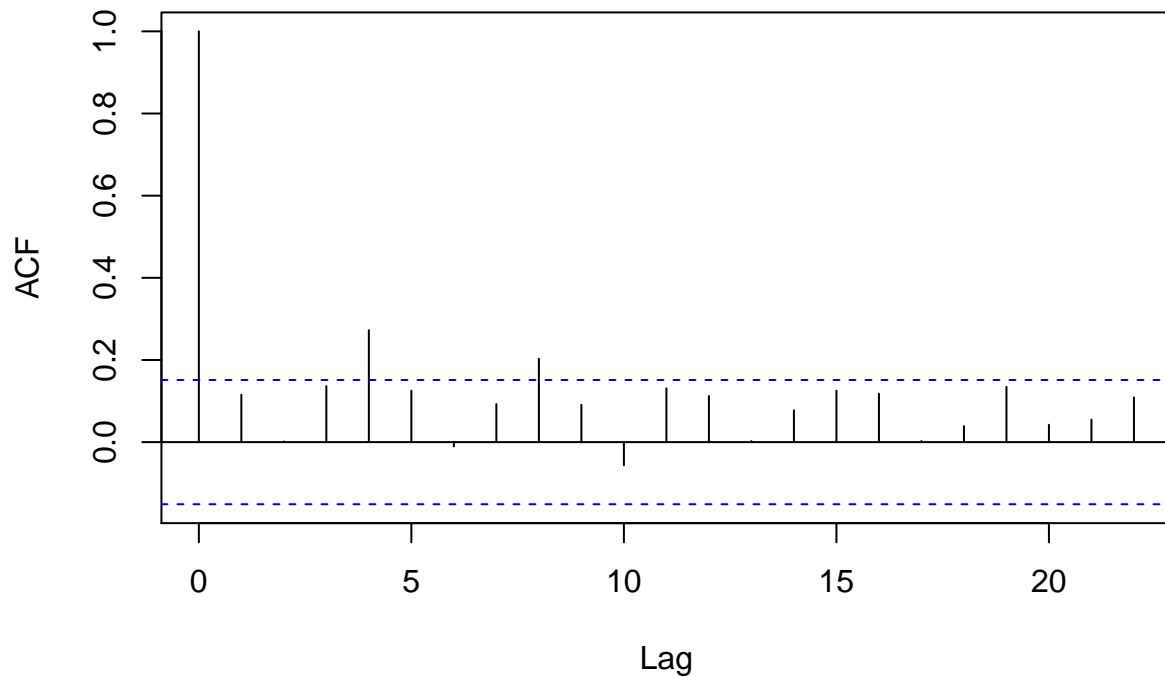
plot(lm_model_month_lag$residuals, type = 'l', main="Plot of the Residuals vs. Time")
```

Plot of the Residuals vs. Time



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

Series lm_model_month_lag\$residuals



Analysis: From the ACF plot we can see that the time series is not dependent on the components (barring maybe 1) and thus is stationary. From the residual plot we can see that there is a small trend, we can use some variable elimination to refine the model.

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 = ~Time+Month_1+Month_2+
                                                Month_3+Month_4+Month_5+Month_6+Month_7+
                                                Month_8+Month_9+Month_10+Month_11+Month_12,
                                                lower = ~1), trace = TRUE,
                                   direction="backward")

## Start:  AIC=-202.02
## TotN_conc ~ Time + 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=-202.02
## TotN_conc ~ Time + 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 0.200 43.436 -203.249
## - Month_1 1 0.220 43.456 -203.170
## - Month_3 1 0.331 43.567 -202.743
## <none> 43.237 -202.023
## - Month_2 1 1.440 44.677 -198.517
## - Month_11 1 2.305 45.541 -195.297
## - Month_5 1 3.274 46.511 -191.760
## - Month_10 1 3.401 46.637 -191.303
## - Month_9 1 7.853 51.089 -175.986
## - Month_6 1 8.215 51.452 -174.797
## - Month_7 1 14.321 57.557 -155.959
## - Month_8 1 16.488 59.725 -149.749
## - Time 1 118.387 161.624 17.499
##
## Step: AIC=-203.25
## TotN_conc ~ Time + 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
## <none> 43.436 -203.249
## - Month_1 1 0.640 44.077 -202.790
## - Month_3 1 0.851 44.288 -201.988
## - Month_11 1 2.235 45.671 -196.819
## - Month_2 1 2.706 46.142 -195.096
## - Month_5 1 3.355 46.791 -192.748
## - Month_10 1 3.502 46.938 -192.223
## - Month_9 1 8.868 52.304 -174.036
## - Month_6 1 9.317 52.753 -172.602
## - Month_7 1 16.912 60.348 -150.004
## - Month_8 1 19.636 63.072 -142.586
## - Time 1 118.194 161.630 15.506
```

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

```
## [1] "TotN_conc" "Time" "Month_1" "Month_2" "Month_3"
## [6] "Month_5" "Month_6" "Month_7" "Month_8" "Month_9"
## [11] "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.

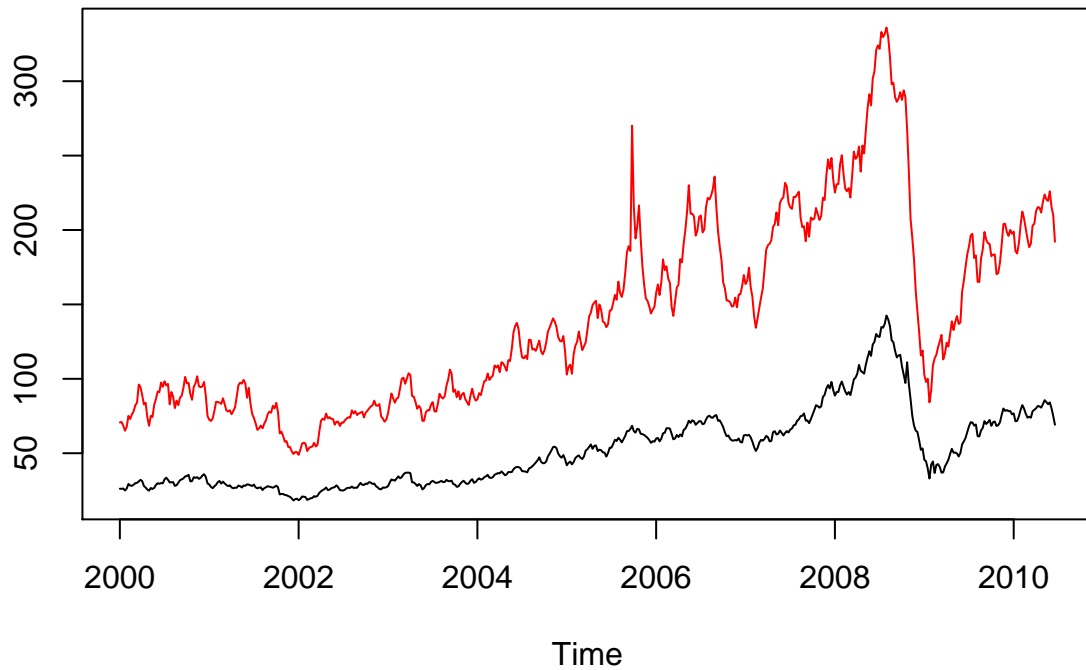
Weekly time series oil and gas present in the package `astsa` show the oil prices in dollars per barrel and gas prices in cents per dollar.

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.

```
set.seed(12345)

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

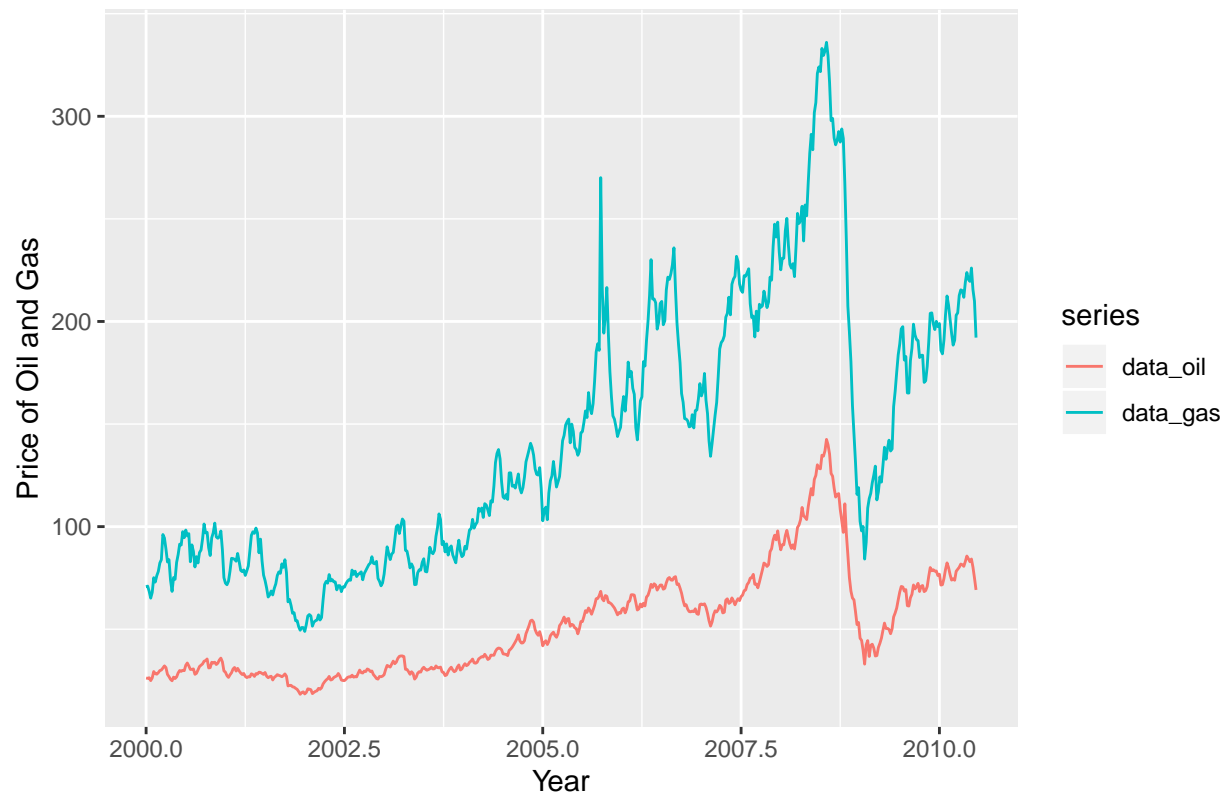
```
ts.plot(data_oil, data_gas, gpars = list(col = c("black", "red")))
```



#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")
```

Price of Oil and Gas vs. Years

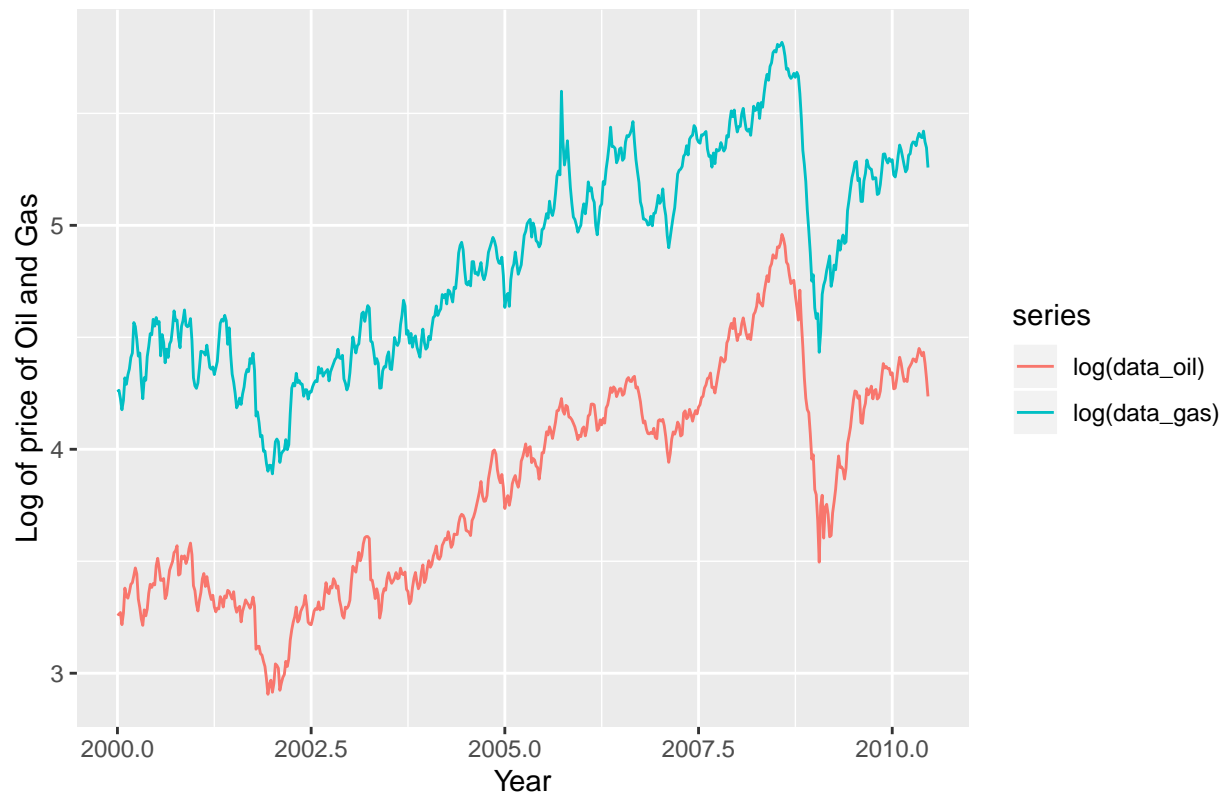


b) Apply log-transform to the time series and plot the transformed data. In what respect did this transformation made the data easier for the analysis?

```
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")
```

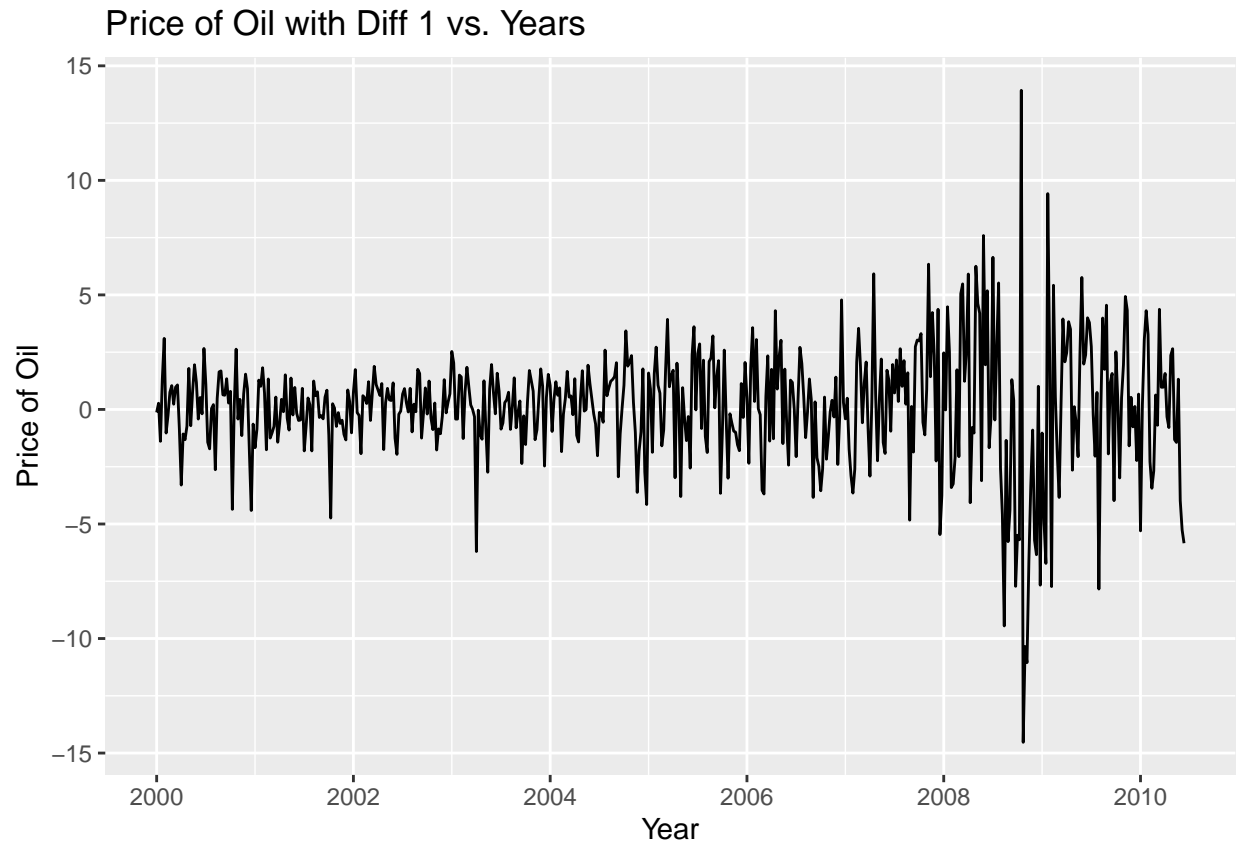
Log of price of Oil and Gas vs. Years



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 $x_t(\text{oil})$ and $y_t(\text{gas})$.

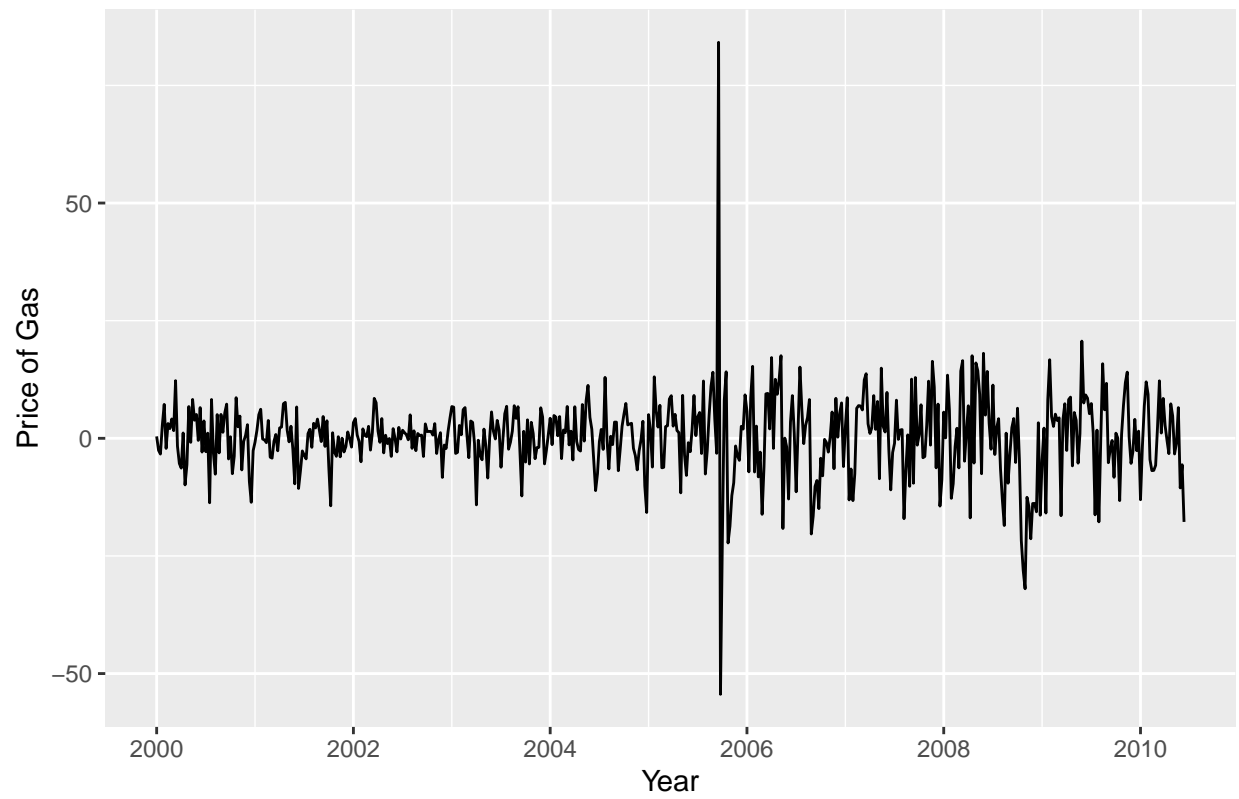
```
set.seed(12345)

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")
```

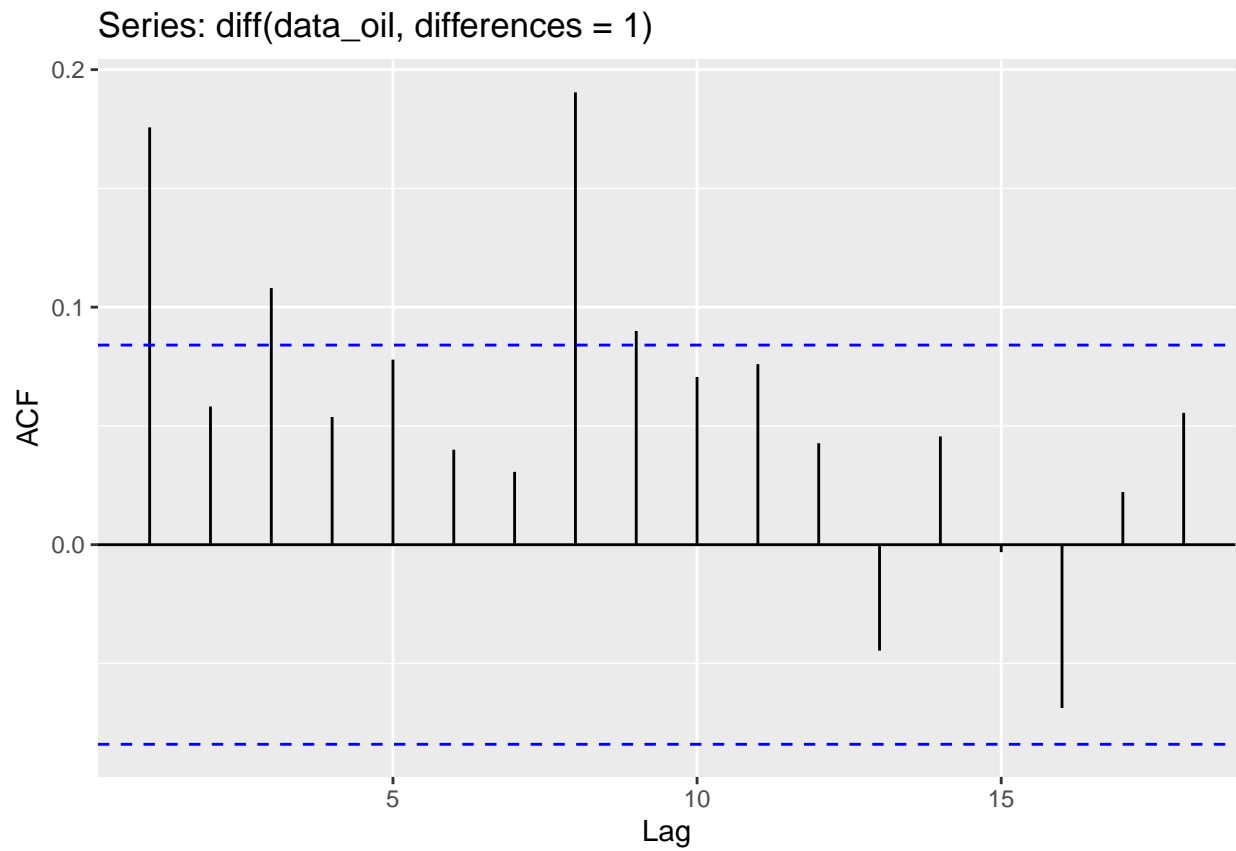



```
autoplot(ts(diff(data_gas, differences = 1), start = 2000, frequency = 52)) +  
  ylab("Price of Gas") + xlab("Year") +  
  ggtitle("Price of Gas with Diff 1 vs. Years")
```

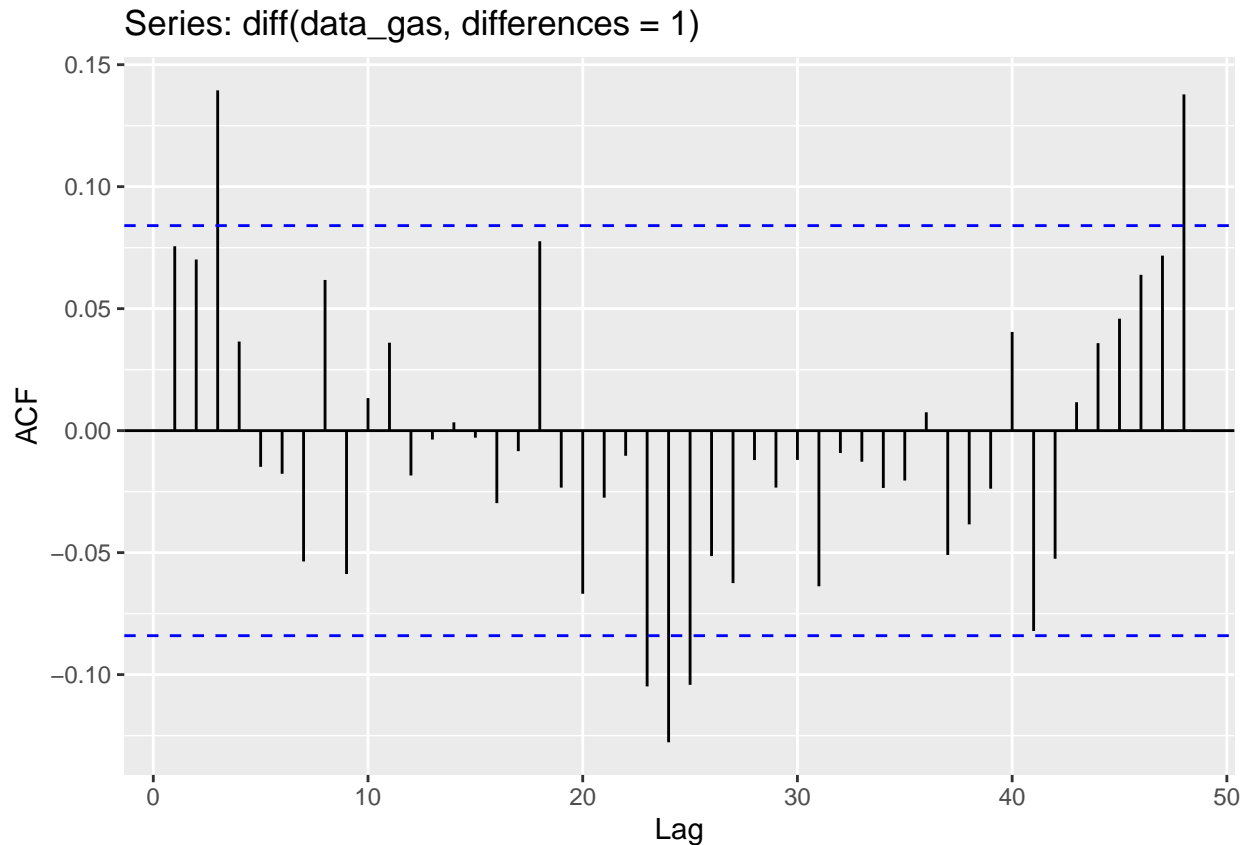
Price of Gas with Diff 1 vs. Years



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



```
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)
```

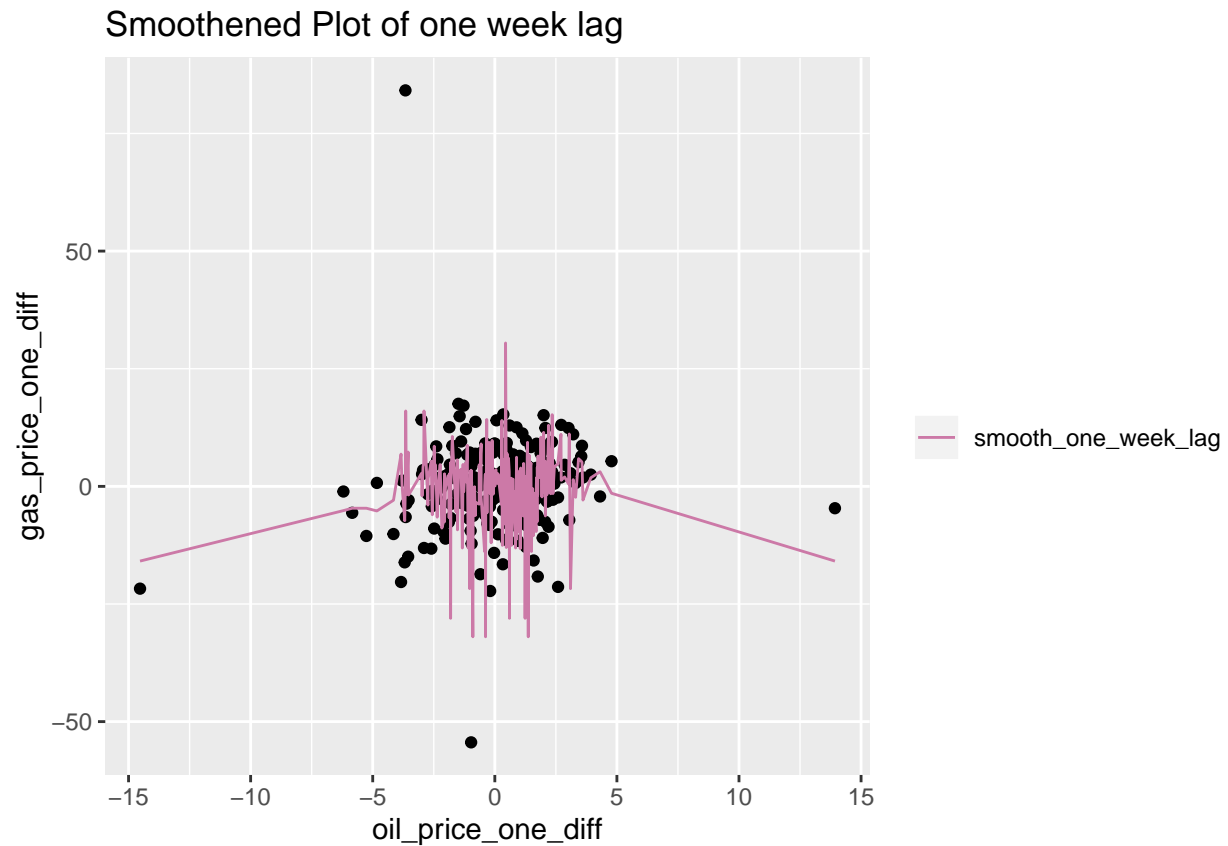
```
df$gas_price_one_diff = lag(df$gas_price_one_diff,1)
df$gas_price_two_diff = lag(df$gas_price_one_diff,2)
df$gas_price_three_diff = lag(df$gas_price_one_diff,3)
```

```
df <- na.omit(df)
```

```
df$smooth_one_week_lag <- ksmooth(x = df$oil_price_one_diff, y = df$gas_price_one_diff, bandwidth = 0.05)
df$smooth_two_week_lag <- ksmooth(x = df$oil_price_one_diff, y = df$gas_price_two_diff, bandwidth = 0.05)
df$smooth_three_week_lag <- ksmooth(x = df$oil_price_one_diff, y = df$gas_price_three_diff, bandwidth = 0.05)
```

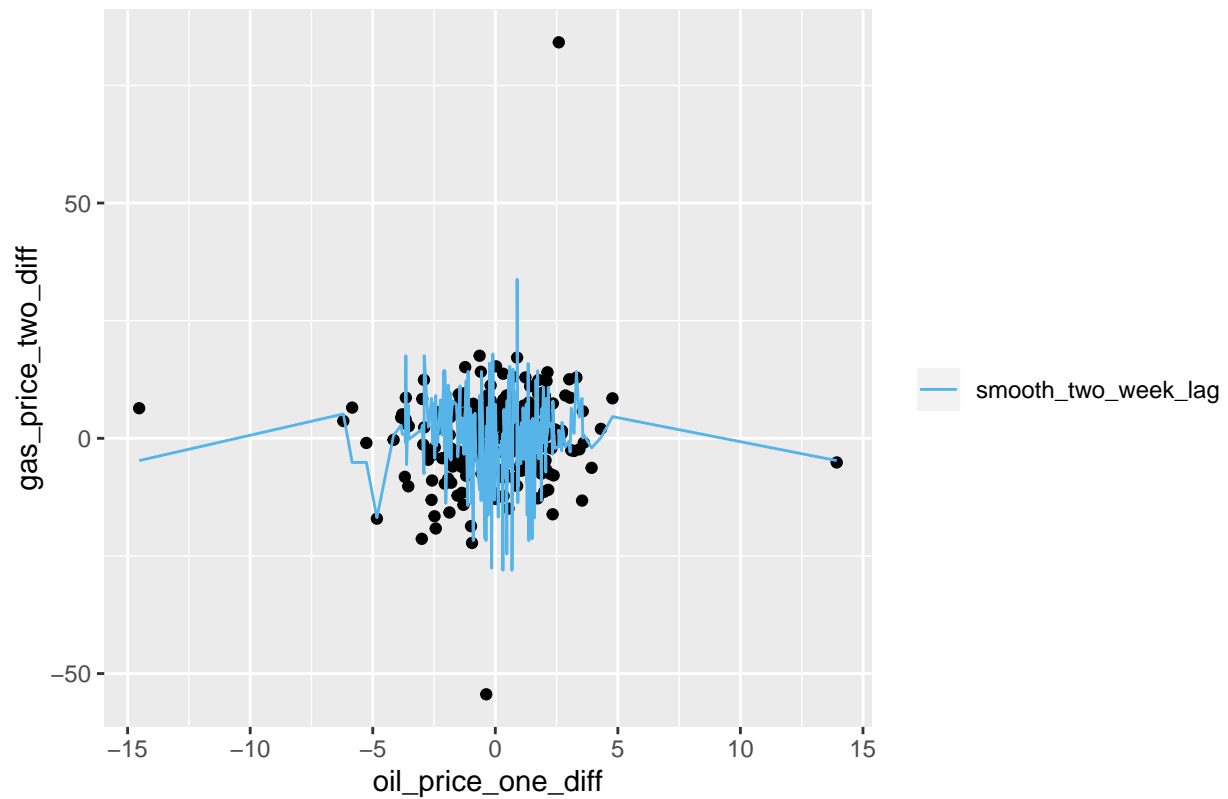
```
df <- na.omit(df)

ggplot(data=df, aes(x=oil_price_one_diff, y = gas_price_one_diff)) + geom_point() +
  geom_line(aes(y= smooth_one_week_lag, color= "smooth_one_week_lag")) +
  scale_colour_manual("", breaks = c("smooth_one_week_lag"),
    values = c("#CC79A7")) +
  ggtitle("Smoothened Plot of one week lag")
```



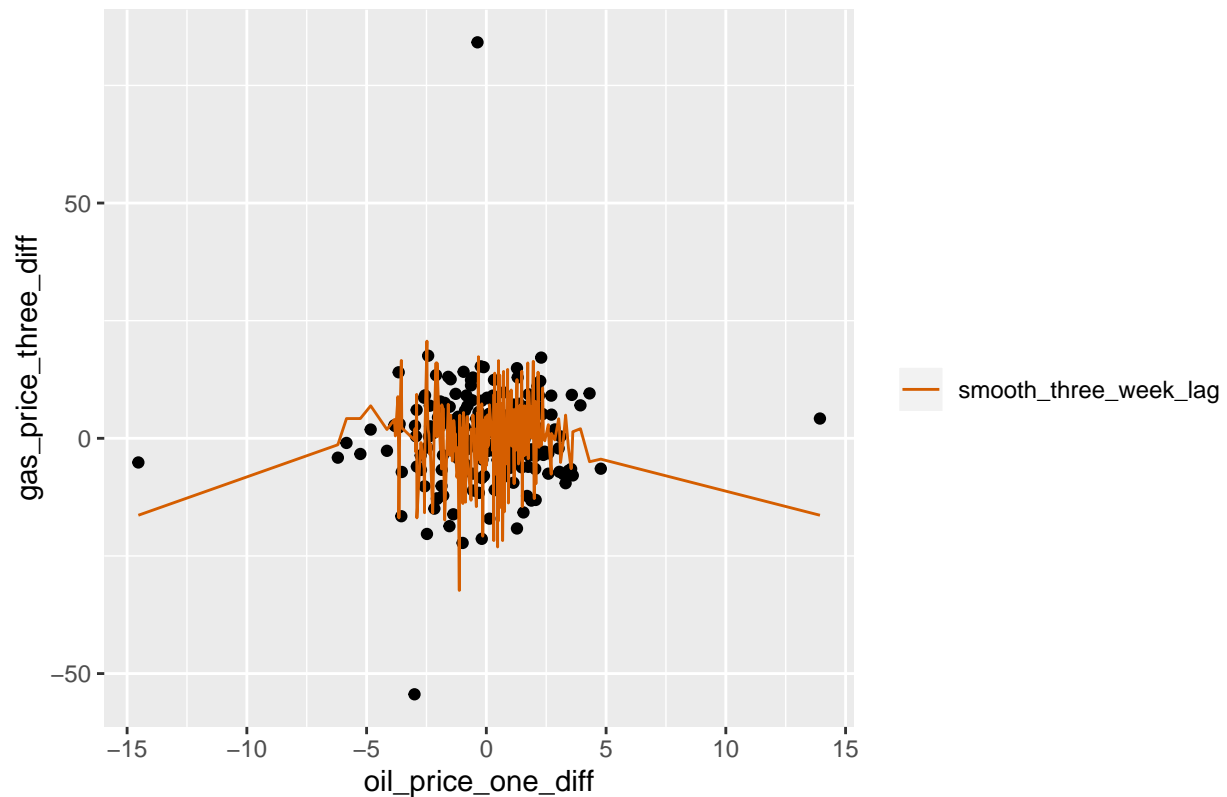
```
ggplot(data=df, aes(x=oil_price_one_diff, y = gas_price_two_diff)) + geom_point() +
  geom_line(aes(y= smooth_two_week_lag, color= "smooth_two_week_lag")) +
  scale_colour_manual("", breaks = c("smooth_two_week_lag"),
    values = c("#56B4E9")) +
  ggtitle("Smoothened Plot of two week lag")
```

Smoothened Plot of two week lag



```
ggplot(data=df, aes(x=oil_price_one_diff, y = gas_price_three_diff)) + geom_point() +
  geom_line(aes(y= smooth_three_week_lag, color= "smooth_three_week_lag")) +
  scale_colour_manual("", breaks = c("smooth_three_week_lag"),
    values = c("#D55E00")) +
  ggtitle("Smoothened Plot of three week lag")
```

Smoothened Plot of three week lag

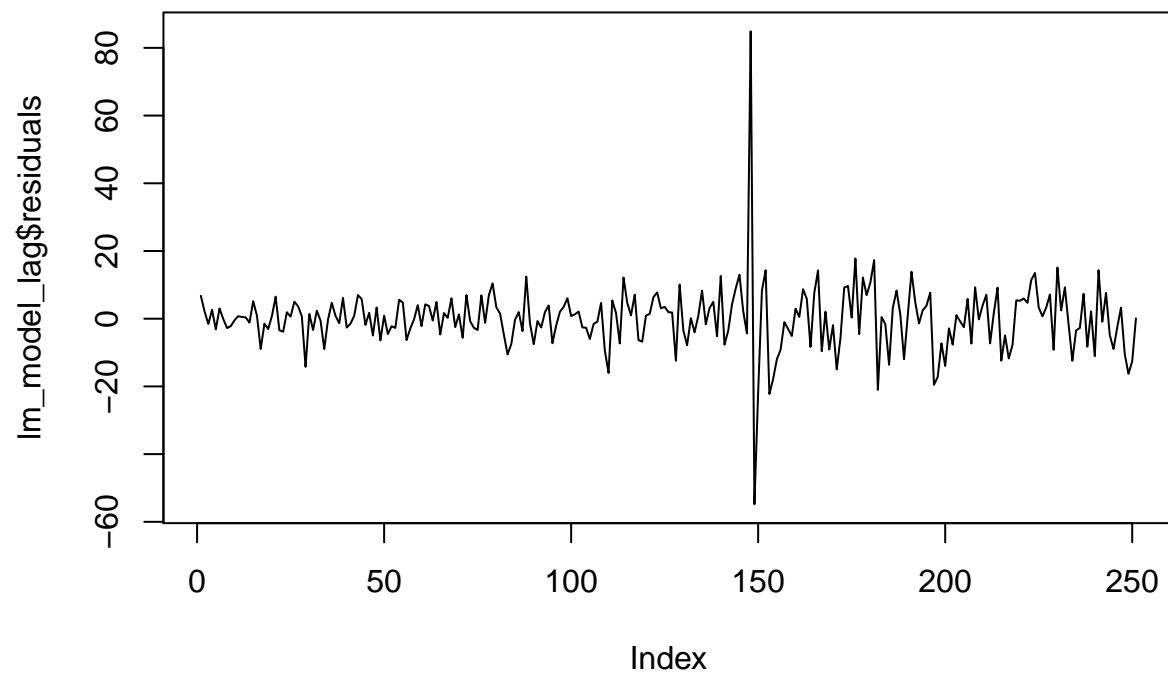


e) Fit the following model: $y_t = \alpha_0 + \alpha_1 I(x_t > 0) + \beta_1 x_t + \beta_2 x_{t-1} + w_t$ and check which coefficients seem to be significant. How can this be interpreted? Analyze the residual pattern and the ACF of the residuals.

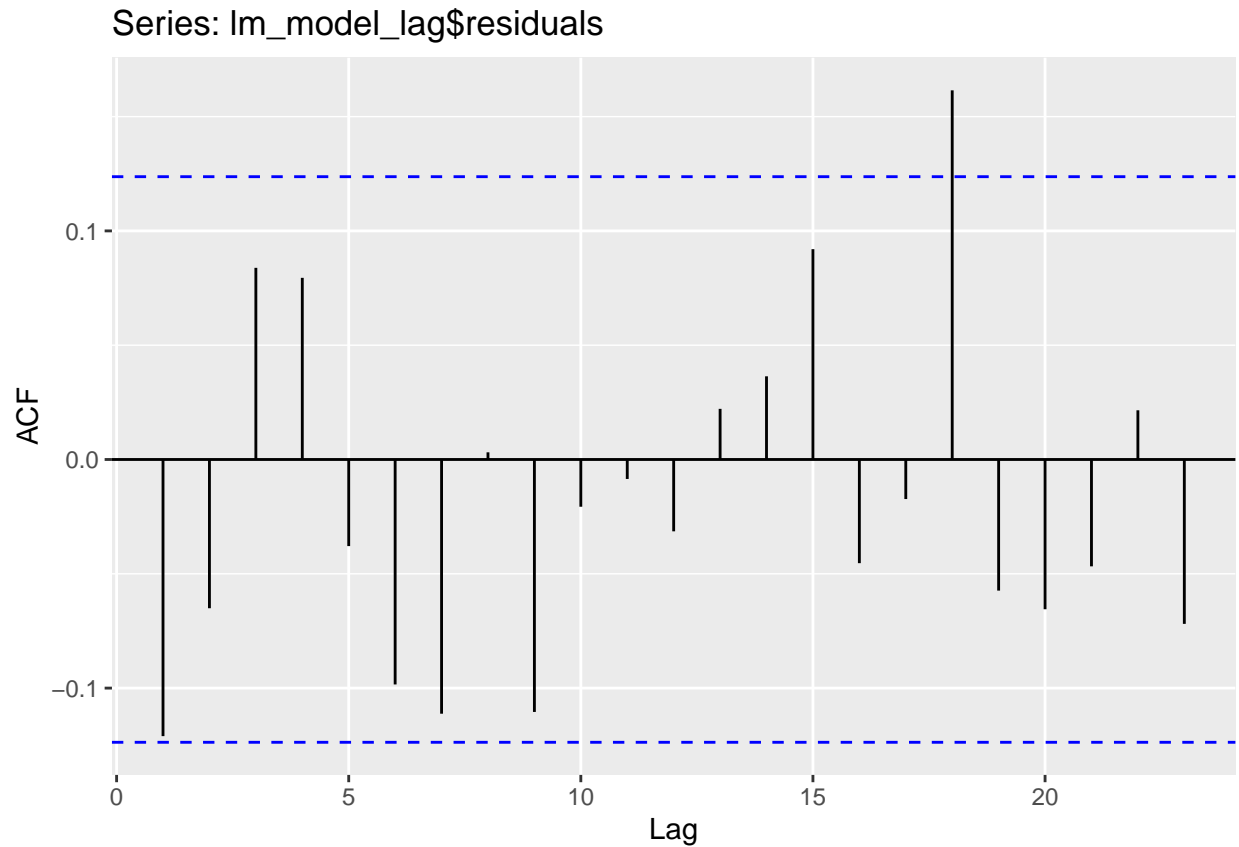
```
set.seed(12345)

df$oil_price_two_diff = lag(df$oil_price_one_diff,2)
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~x_t_more_zero+oil_price_one_diff+oil_price_two_diff)

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



```
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)
```

```

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)

```

```

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, main="Time Series of Nitrogen Concentration in Rhine")
lag.plot(rhine_data_ts, lags = 12)
acf(rhine_data_ts)

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

gglagplot(rhine_data_ts, lags = 1, set.lags = 1:12, color=FALSE)
ggAcf(rhine_data_ts) + ggtitle("ACF for Total Nitrogen Concentration")

set.seed(12345)

rhine_lm_model <- lm(TotN_conc~Time, data=rhine_data)
plot(rhine_lm_model$residuals, type = 'l', main="Plot of Residual from the linear model of Nitrogen Conc
acf(rhine_lm_model$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", fill = "0")

lm_model_month_lag <- lm(data=rhine_data_wide,
                        TotN_conc~Time+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$residuals, type = 'l', main="Plot of the Residuals vs. Time")
acf(lm_model_month_lag$residuals)

set.seed(12345)

lm_model_month_lag_step <- stepAIC(lm_model_month_lag,
                                scope = list(upper = ~Time+Month_1+Month_2+
                                                Month_3+Month_4+Month_5+Month_6+Month_7+
                                                Month_8+Month_9+Month_10+Month_11+Month_12,
                                                lower = ~1), trace = TRUE,
                                direction="backward")

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

data_oil <- asts::oil
data_gas <- asts::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 <- na.omit(df)

df$smooth_one_week_lag <- ksmooth(x = df$oil_price_one_diff, y = df$gas_price_one_diff, bandwidth = 0.05)
df$smooth_two_week_lag <- ksmooth(x = df$oil_price_one_diff, y = df$gas_price_two_diff, bandwidth = 0.05)
df$smooth_three_week_lag <- ksmooth(x = df$oil_price_one_diff, y = df$gas_price_three_diff, bandwidth = 0.05)

df <- na.omit(df)

ggplot(data=df, aes(x=oil_price_one_diff, y = gas_price_one_diff)) + geom_point() +
  geom_line(aes(y= smooth_one_week_lag, color= "smooth_one_week_lag")) +
  scale_colour_manual("", breaks = c("smooth_one_week_lag"),
                      values = c("#CC79A7")) +
  ggtitle("Smoothened Plot of one week lag")

ggplot(data=df, aes(x=oil_price_one_diff, y = gas_price_two_diff)) + geom_point() +
  geom_line(aes(y= smooth_two_week_lag, color= "smooth_two_week_lag")) +
  scale_colour_manual("", breaks = c("smooth_two_week_lag"),
                      values = c("#56B4E9")) +
  ggtitle("Smoothened Plot of two week lag")

ggplot(data=df, aes(x=oil_price_one_diff, y = gas_price_three_diff)) + geom_point() +
  geom_line(aes(y= smooth_three_week_lag, color= "smooth_three_week_lag")) +
  scale_colour_manual("", breaks = c("smooth_three_week_lag"),
                      values = c("#D55E00")) +
  ggtitle("Smoothened Plot of three week lag")

set.seed(12345)

df$oil_price_two_diff = lag(df$oil_price_one_diff,2)
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~x_t_more_zero+oil_price_one_diff+oil_price_two_diff)

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

```

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