732A62: Lab 1
Time Series Analysis

Joshua Burrata, Carles Sans Fuentes

October 6, 2017

Assignment 1- Computations with simulated data

a

In this question we are asked to generate two time series with 100 observations being $x_0 = x_1 = 0$ such that

$$x_t = -0.8 * x_{t-2} + w_t$$

called x_{t1} in the code and

$$x_t = cos(2 * \pi * t/5)$$

called x_{t2} in the code. Then, a a smoothing filter is applied which is

$$v_t = 0.2 * (x_t + x_{t-1} + x_{t-2} + x_{t-3} + x_{t-4})$$

in order to see whether it gets affected or not. The results are plotted in figure 1 and 2 for each function with and without filter.

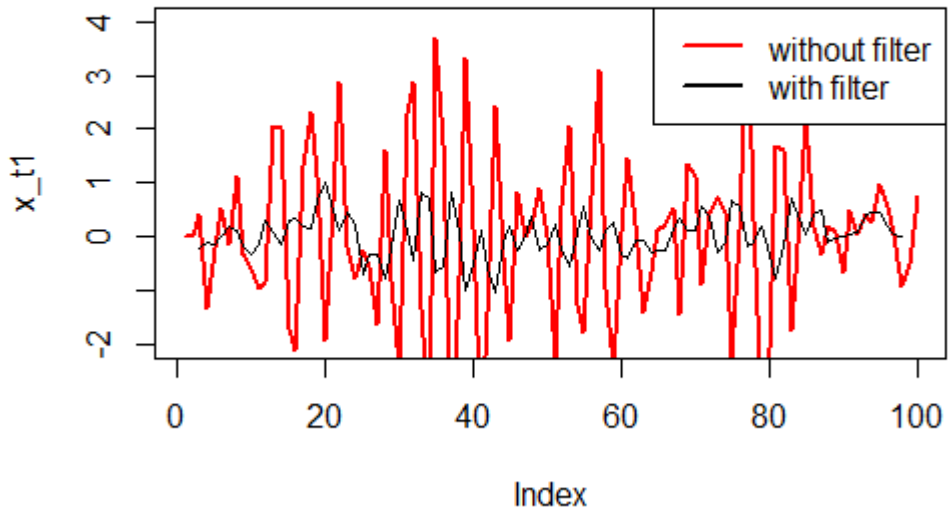

Figure 1: Histogram of 100 observations for x_{t1} with and without the filter

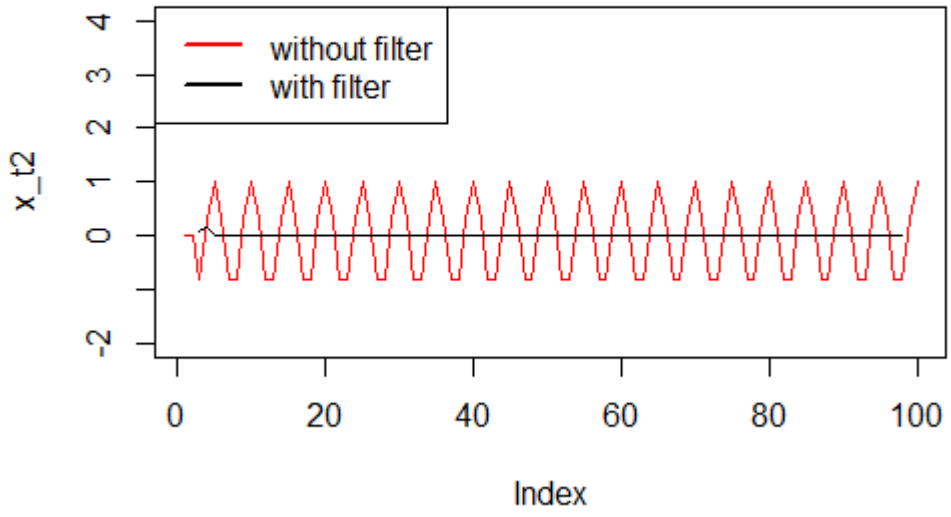

Figure 2: Histogram of 100 observations for x_{t2} with and without the filter

The filter on the first time series dampens the amplitude of the data whereas in the second case the filter removes the whole noise. This is due to the fact that we average over the last 5 points of the cosine distribution, letting the filter being almost 0 and so getting the same result.

b

Now we consider the series

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

In order to investigate if an ARMA process is causal, we must check the AR part, i.e. the left hand side of the equation. If the roots of the AR polynomial are contained between -1 and 1 then we can say that this time series is NOT causal. For the other cases, it is causal. In our case, this is not true given that the roots of this polynomial were:

```
> causal1
[1] 0.2936658 1.6793817 1.0000000 1.4239626 1.4239626
```

For invertibility, we proceed in the same way as for causality but this time by checking the roots of the right hand side polynomial, the MA part of the time series. A time series is invertible only if these roots are not between -1 and 1. The result got is the following:

```
> causal2
[1] 0.6874372 0.6874372 0.6874372 0.6874372 1.0580446 1.0580446
```

All in all, we can see that the time series is not either casual nor invertible.

c

We use the arima.sim() function from R now to simulate 100 observations from the process

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

. Recall that since we need the arima model to be specified in the way of

$$x_t = \phi_1 * x_{t-1} + ... + \phi_p * x_{t-p} + w_t + \theta_1 * w_{t-1} + ... + \theta_q * w_{t-q}$$

, only the coefficients with θ or ϕ must be specified.

Below, both the theoretical (using ARMAacf()) and sample acf() plots will be shown:

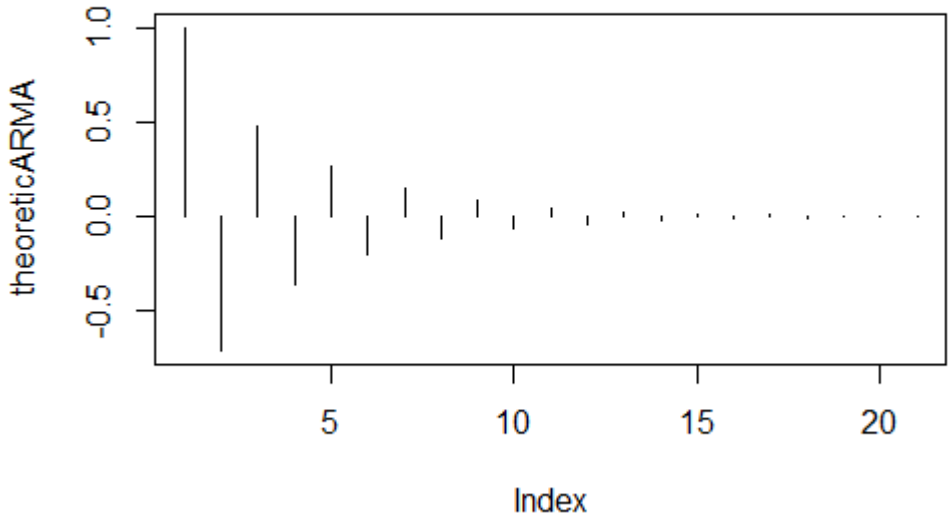

Figure 3: Theoretical acf

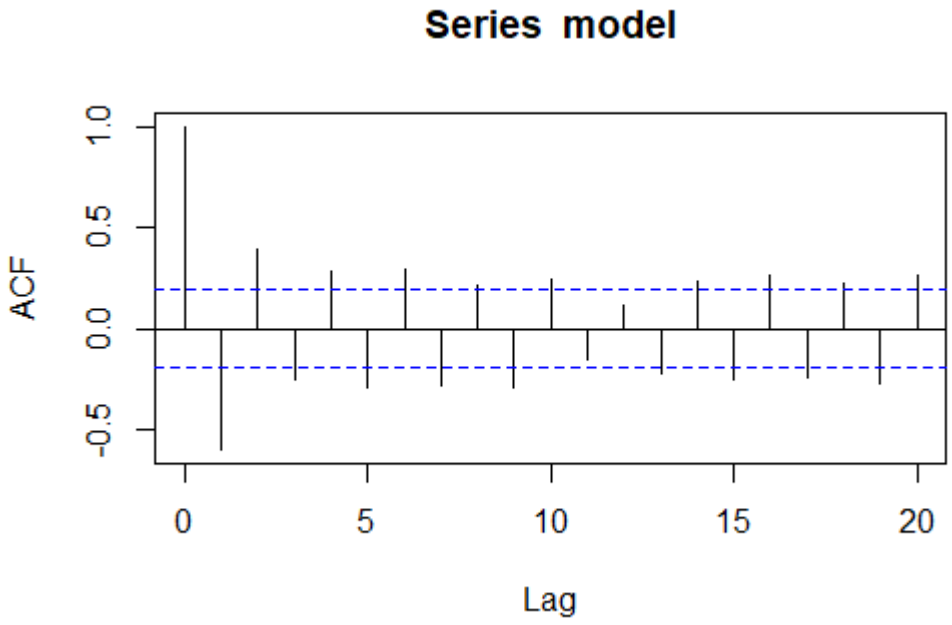

Figure 4: sample acf

It can be seen that in both acfs look quite different: autocorrelations tails down smoothly for the

theoretical acf whereas on the sample acf it goes down more abruptly, which is probably down to the extra noise taken into account on the sample.

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

a

In this part we have imported the rhine.csv data and made it a time series object using the ts() function, which stands for time series function in R. We have plotted the time series and x_t against $x_{t-1}...x_{t-12}$ below.

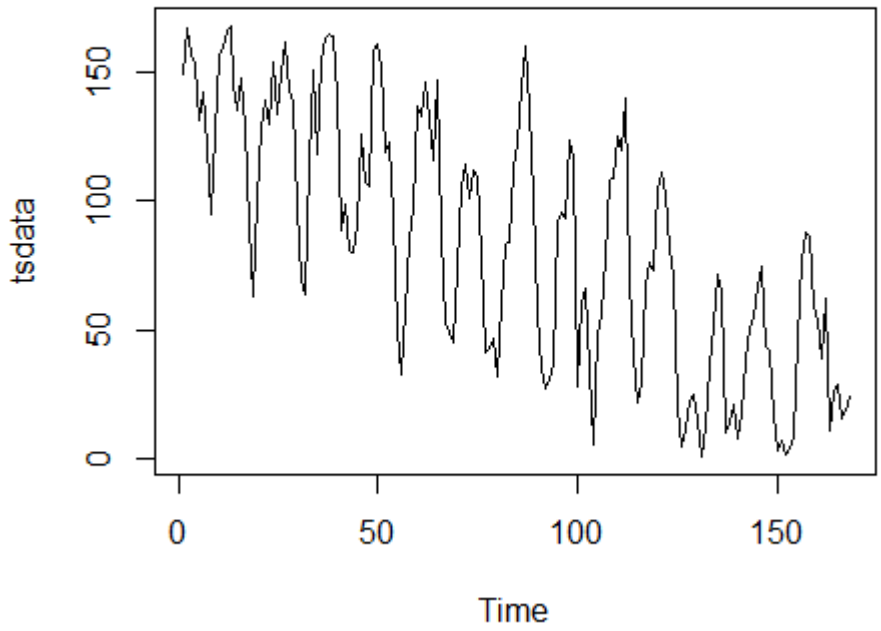

Figure 5: Time series plot of our rhine data

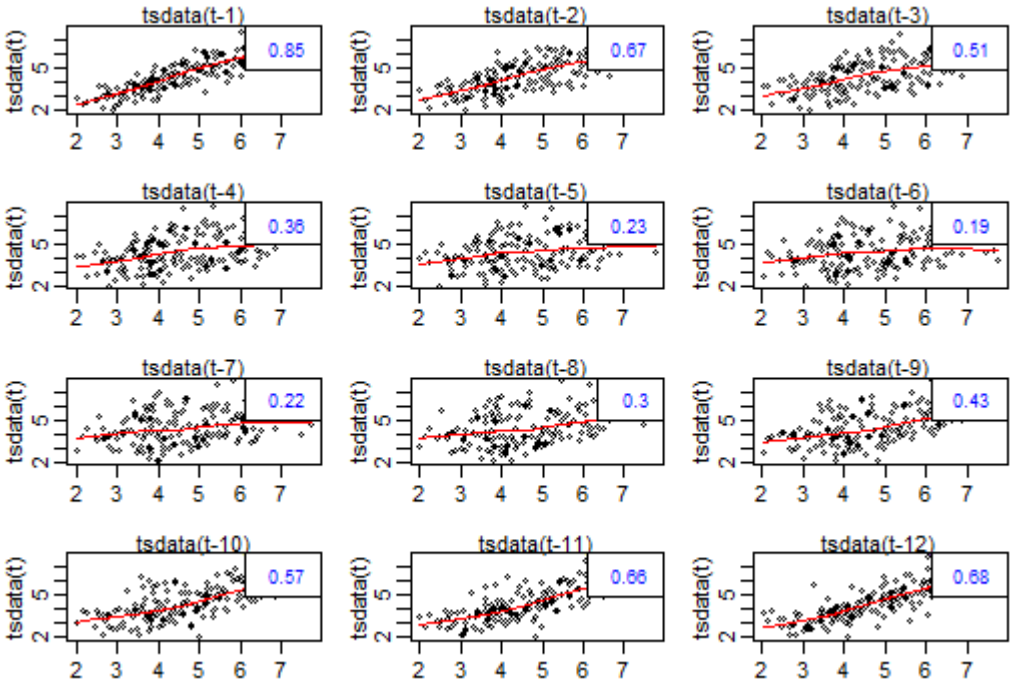

Figure 6: Scatterplots of our rhine data with its lags from 1-12

In the time series plots, there is a downward-linear trend with possible seasonal trends during the same year around this trend due to the periodic appearance of the peaks. From the scatterplots we can see that the lag 1,2, 11 and 12 plots are quite similar, which would further point towards a seasonal trend as these lags would fall into the same period of the year.

b

In order to see it better and get stationarity on the data, we try to eliminate the trend by running a linear model regression. The result of the linear model is

```
> summary(fit)

Call:
lm(formula = y ~ X, data = mydata)

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 ***
X           -0.017796   0.001305  -13.63   <2e-16 ***
---
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
```

There is clearly a downward-linear trend. Both coefficients are found to be significant. In figures 7 and 8 we plotted the residual pattern and the sample ACF.

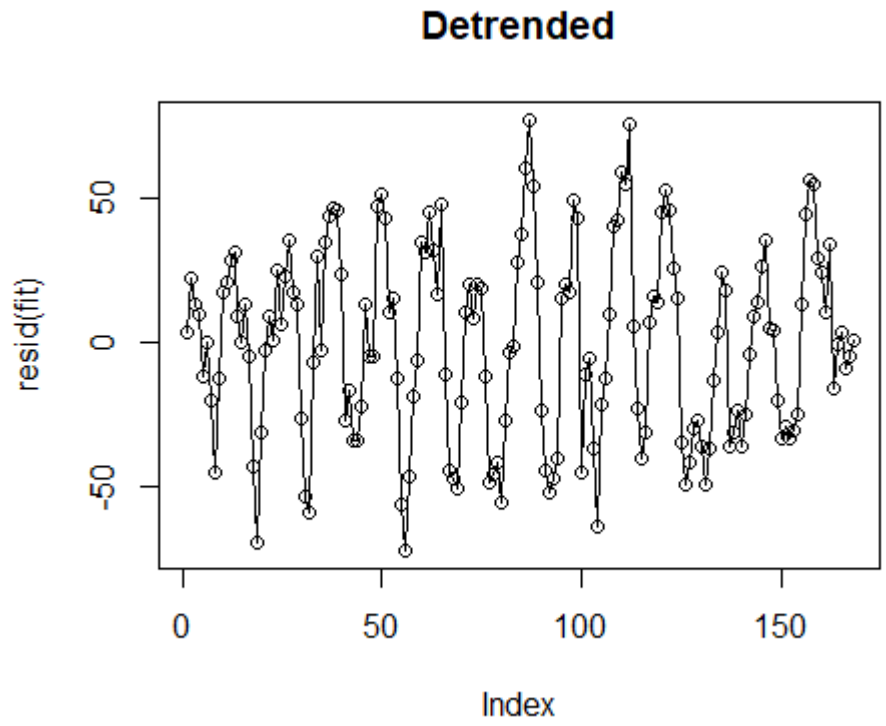

Figure 7: Plotting of the residuals of the lm regression model

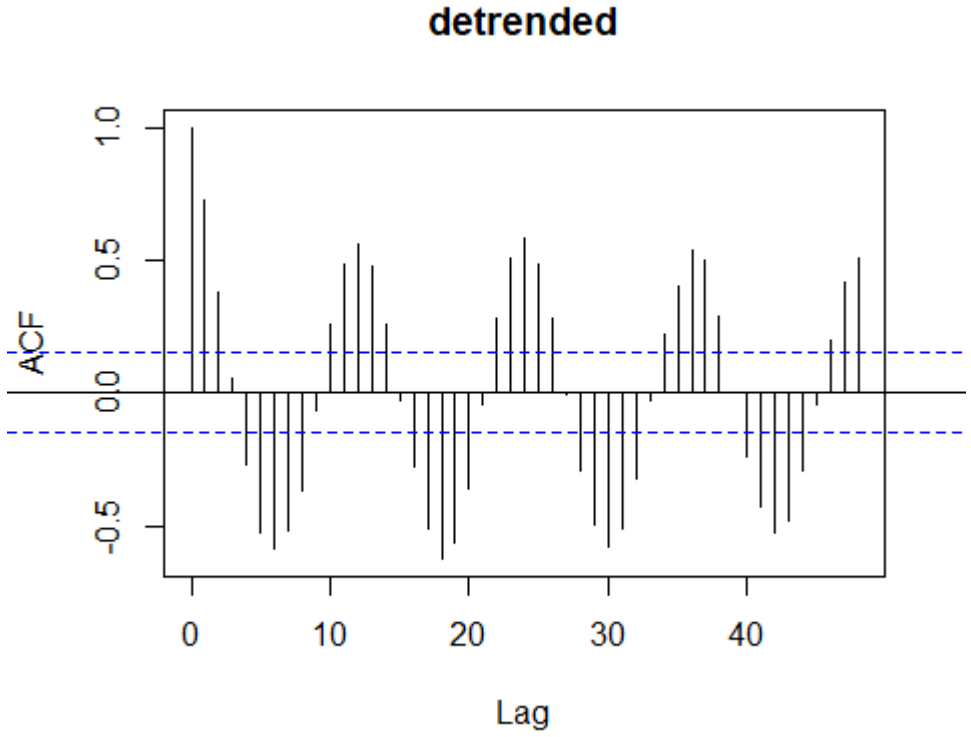

Figure 8: acf of the data

It can be seen that our residuals have a seasonal pattern and are stationary around 0. Thus, the model does not account for the seasonal variance. Also, since the residuals are out of the bands, we can say that the model is not good enough since it does not capture this seasonality neither it decreases with time.

c

In this part we fit a kernel smoother model to eliminate the trend. The kernel bandwith is set to be 2 and 10 respectively. The plots of the residuals as well as its acf are below:

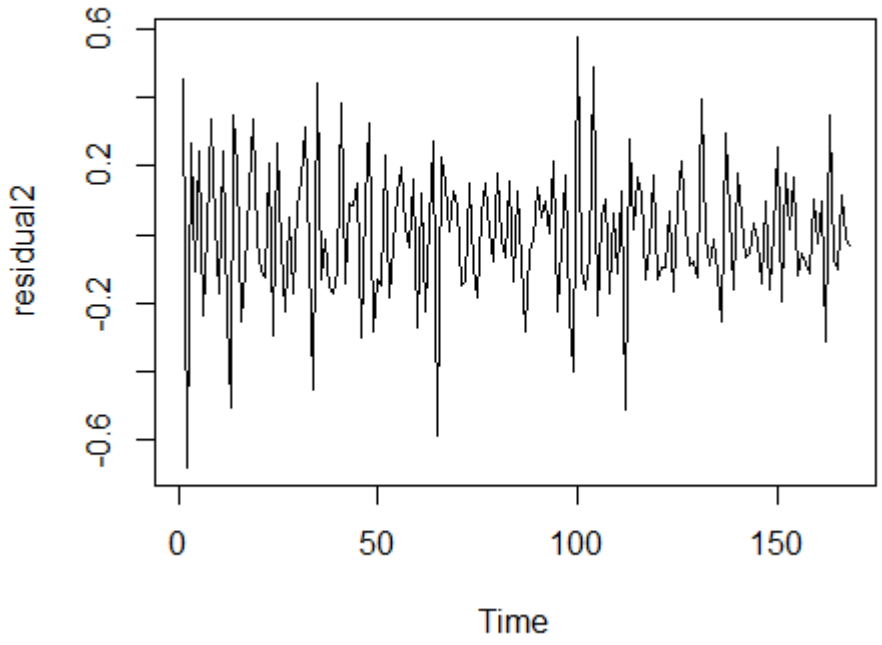

Figure 9: plot of the residuals with bandwith 2

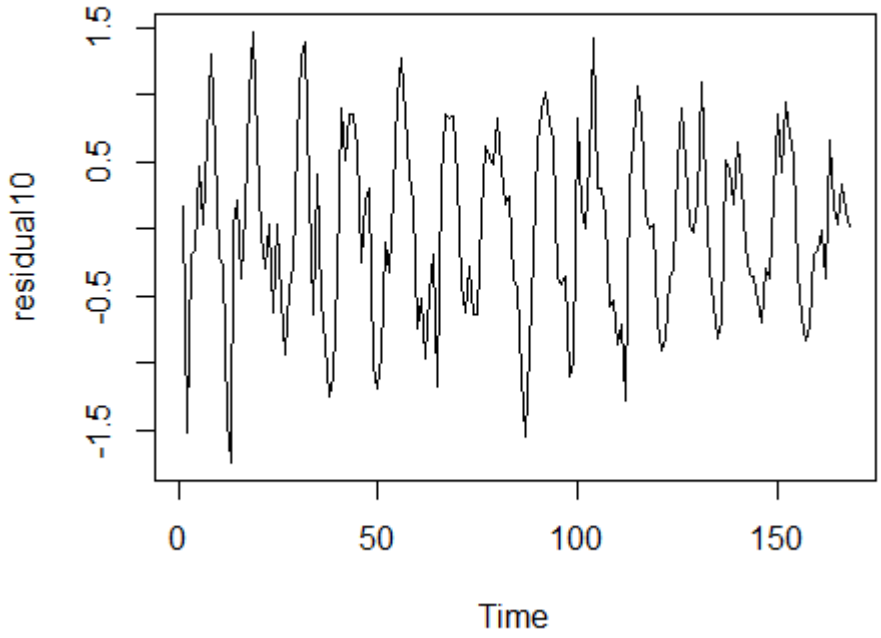

Figure 10: plot of the residuals with bandwith 10

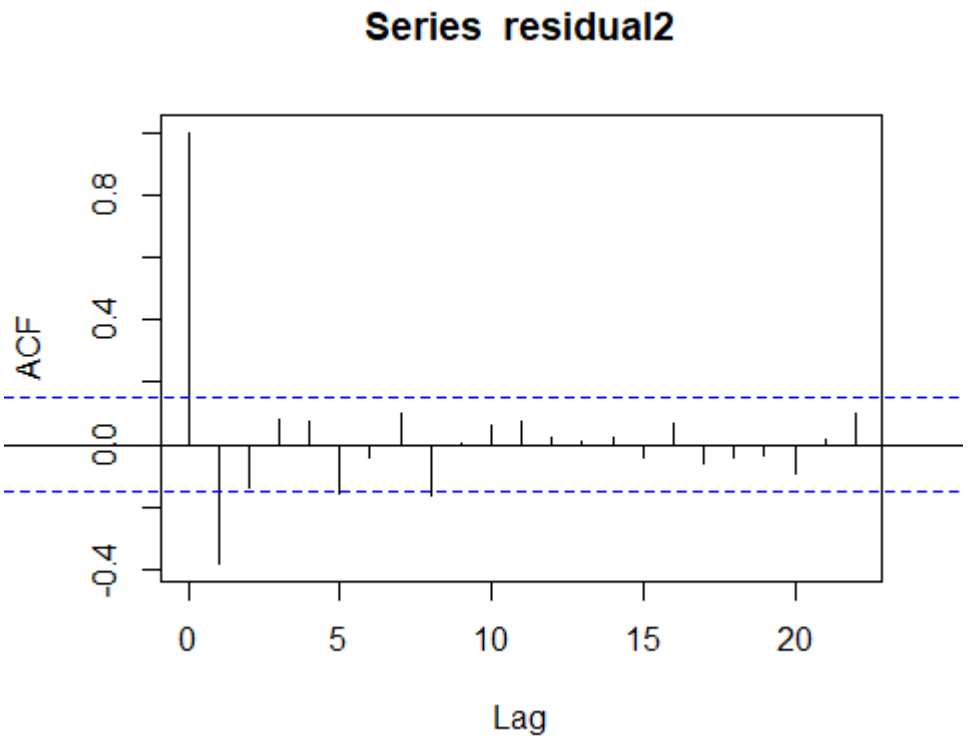

Figure 11: acf of the residuals with bandwith 2

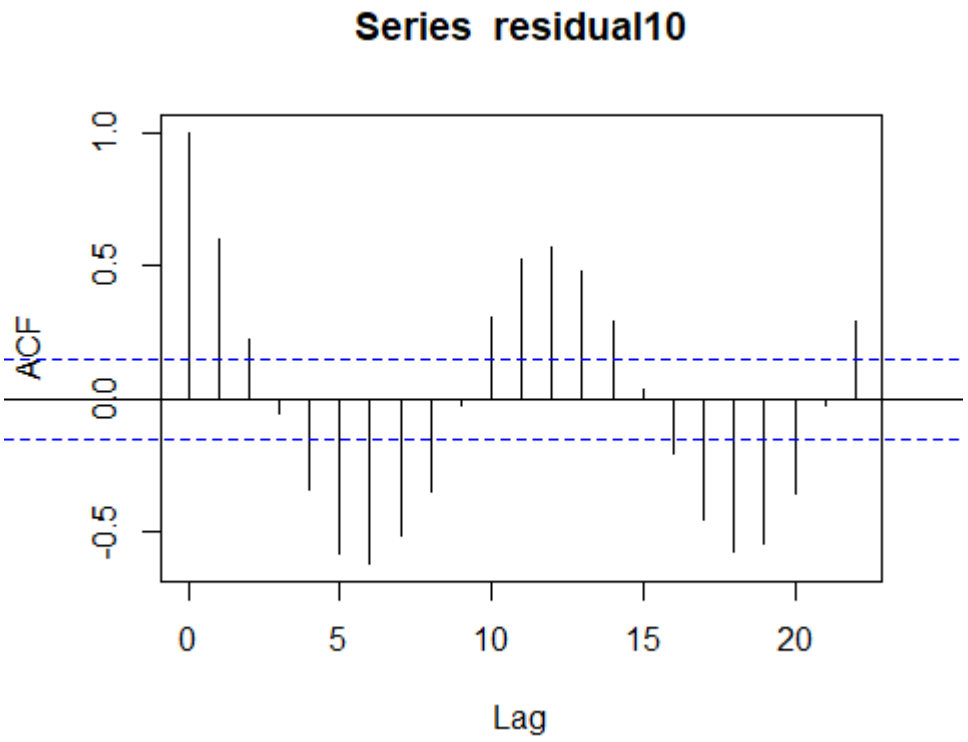

Figure 12: acf of the residuals with bandwith 10

The higher the bandwidth, the more smooth our model gets, capturing less variance on the model and getting higher value of the residuals getting them not to be white noise. In the case of bandwidth equals to 10 the results show that the trend is eliminated BUT the residuals seem to still have some seasonal pattern which is not eliminated (residual 10). This seems to be similar for b. However, in the case for bandwidth equals to 2, the range of the residuals is still quite small around 0, appearing to be around 0 without autocorrelation (just white noise).

For that, the model that eliminates the variance on the residuals is better. In our case, this is when bandwidth equals to 2.

d

We next used a seasonal means model to eliminate the trend, given by the following equation:

$$x_t = \alpha_0 + \alpha_1 t + I(month = 2) + ... + I(month = 12) + \omega_t$$

where $I(x) = 1$ if x is true, 0 otherwise.

Below are the plots for the model residuals (Fig 13) and finally the ACF of these residuals (Fig 14).

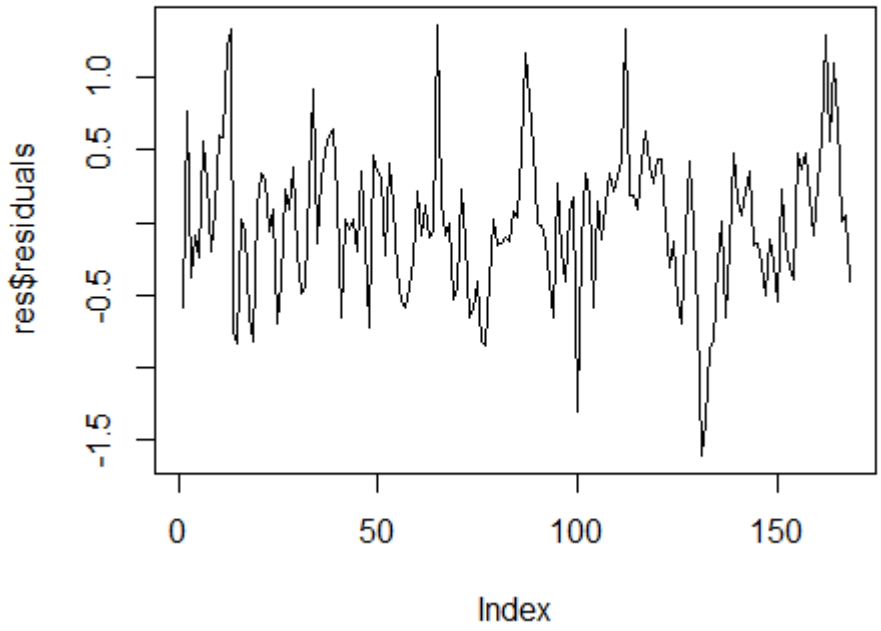

Figure 13: Seasonal Means model fitted to the data

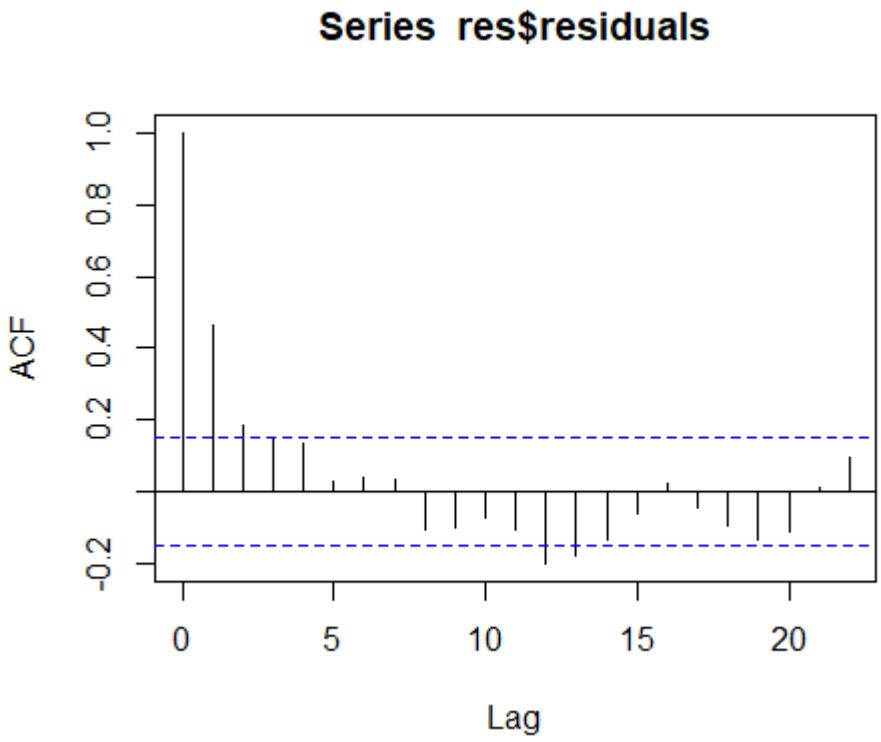

Figure 14: ACF of the residuals of the model

From the plots, we can see that the model fits the data rather well, the seasonal means model allowing for more sharp changes from one period to the next compared to the kernel smoother. The residuals are therefore quite small, similar to the kernel smoother with bandwidth 2. The ACF however shows in Fig 14 that the residuals are equally or even less autocorrelated than in previous models, and may have been even been reduced to white noise: the lines are almost entirely contained within the dotted blue band.

e

Finally we performed stepwise variable selection on the seasonal means model obtained in the previous question. The model with the lowest AIC had an AIC value of -204.57. Below it can be found the summary of the different models, seing at the end the one with AIC value of -204.57:

```
> step<-step(res,direction="both")
Start:  AIC=-86.98
y ~ x + month

         Df Sum of Sq      RSS      AIC
<none>                 96.596 -86.975
- month  1    15.165 111.761 -64.477
- x      1   118.387 214.983  45.428
```

The best model is the one without removing variables having an AIC equal to -86.975

Assignment 3- Analysis of oil and gas time series

In this activity we use Weekly time series oil and gas present in the package astsa which show the oil prices in dollars per barrel and gas prices in cents per dollar.

a

Below, it can be seen plotted the given time series in the same graph. They can look like stationary series if trend is removed and they seem somewhat related as the peaks of both are the same, only at a larger scale with the gas series compared to the oil series.

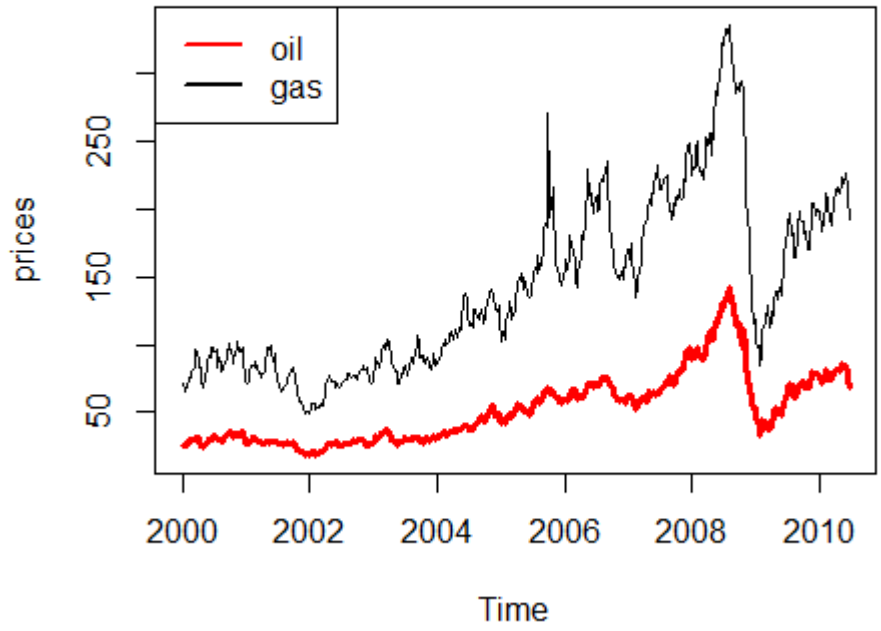

Figure 15: Plot of the oil and gas prices

b

A log-transformation was then applied to the data and the resulting time series plotted (see 16). The variance and the relation with this transformation makes both variables look as if there really does exist a clear relation between them.

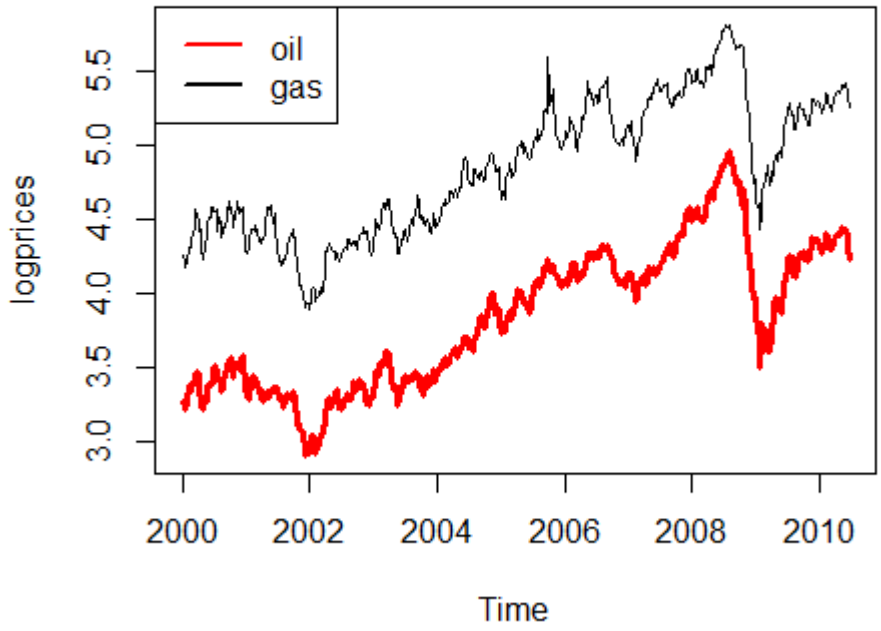

Figure 16: Plot of the log(oil) and log(gas) prices

c

Next the trend in the data was eliminated by computing the first difference of the transformed data; plots of the detrended series can be found in figures 17 and 18 and their respective ACFs in figures 19 and 20.

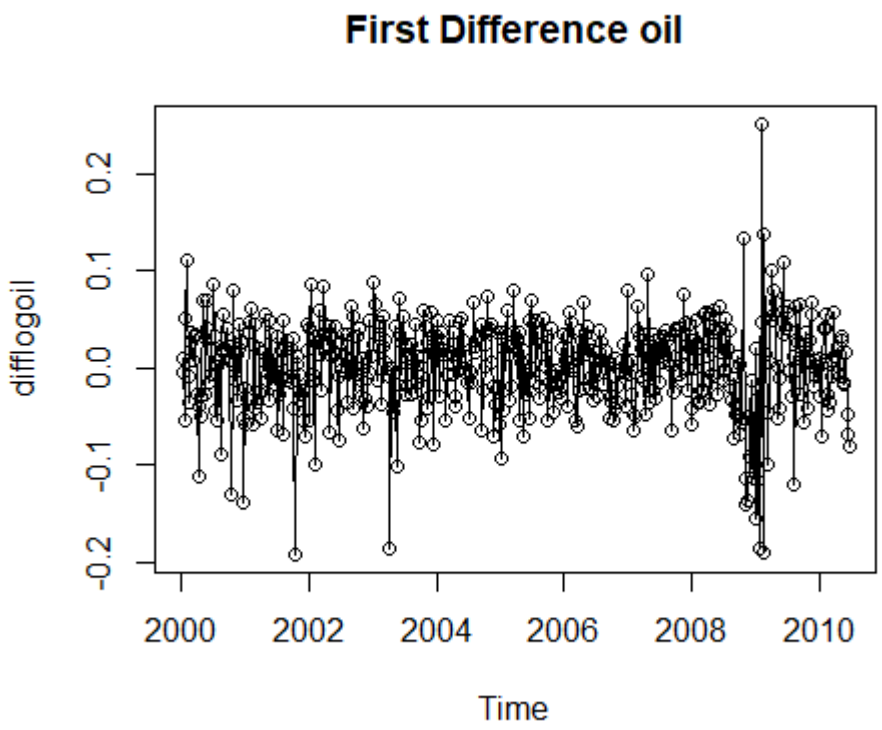

Figure 17: Plot of the log(oil) and log(gas) prices

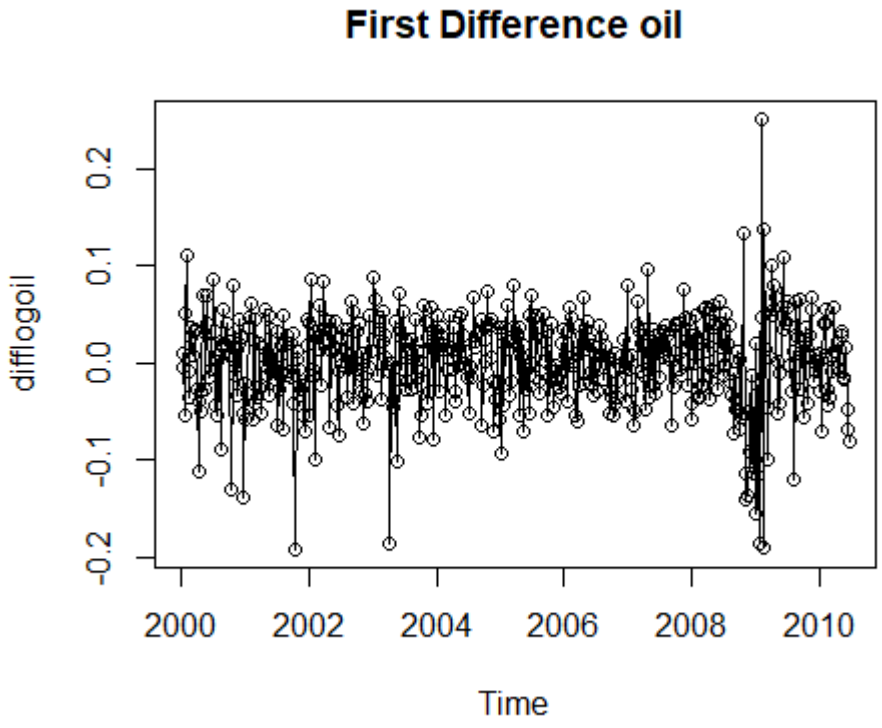

Figure 18: Plot of the diff of the log(oil) prices

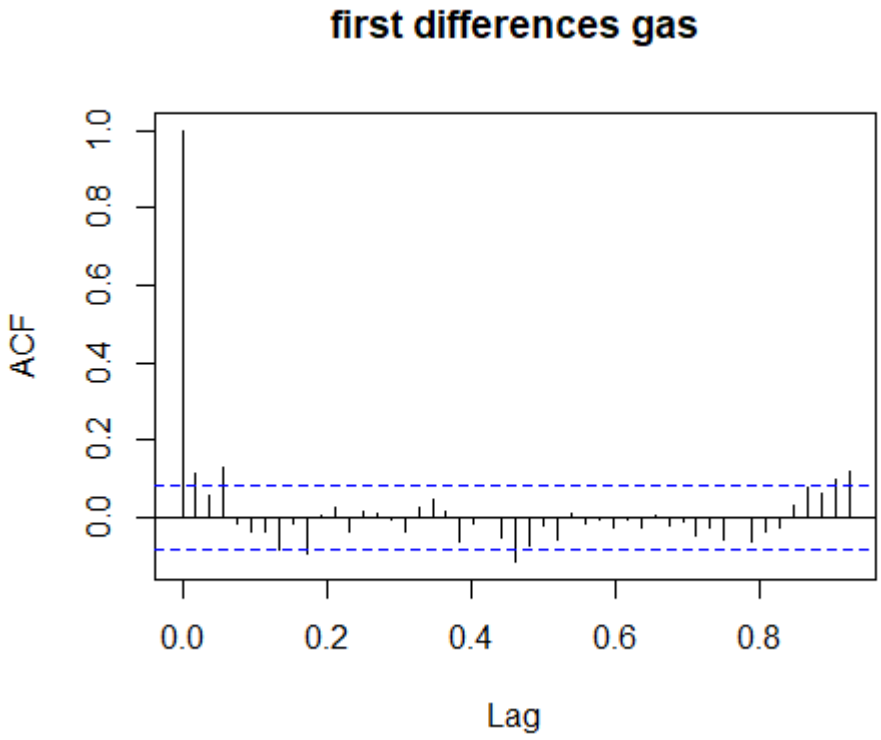

Figure 19: ACF of the diff of the log(gas) prices

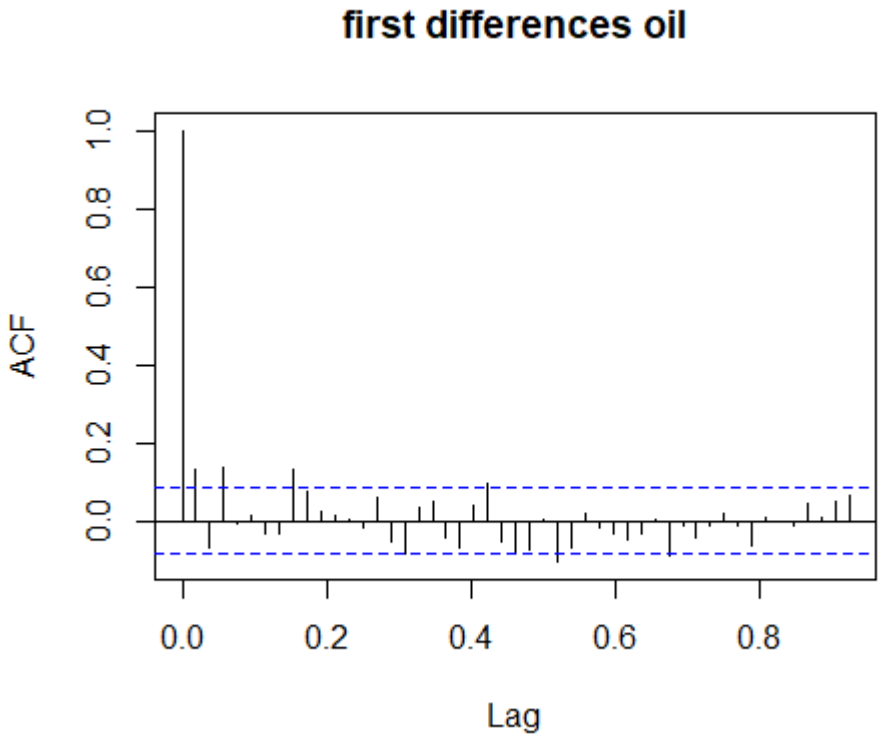

Figure 20: ACF of the diff of the log(oil) prices

We can see from the residual plots that the mean of these residuals for both gas and oil is around 0. However, there still remains larger peaks of variance in certain points of time (possibly at times of crisis). Nonetheless, the ACF plots would indicate that the residuals are not inside the bands. Hence the model is not able to capture well enough of the variance and so the detrended show some correlation, with the values being outside the blue band.

d

Next we produced the scatterplots of y_t with respect to up to 3 weeks lead time of x_t, as shown below.

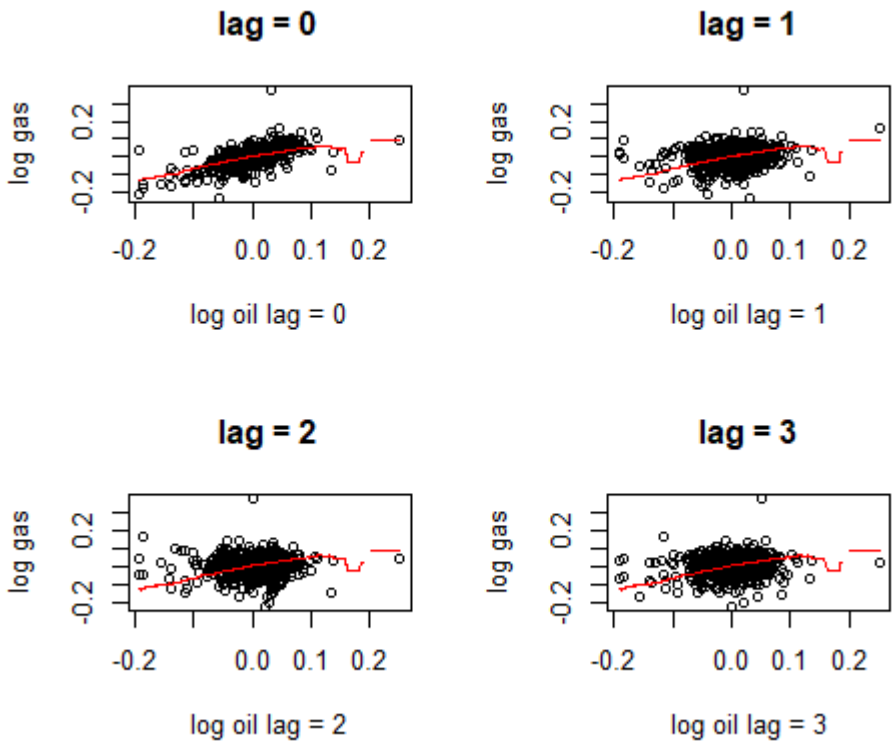

Figure 21: Plot of the log distribution of oil gas for leads 0-3 on oil

From the plots we can see definite outliers, with data points far from the distribution at all leads. The relationships seem linear as the kernel smoother we applied to each set of data (as shown in red in the plots) are roughly linear. In addition the trends displayed by kernel smoothing seems almost identical, even thought the data points become more and more centered around 0. The relationship is close to a linear model with slope 1, being quite a strong relation.

e

Now the model $y_t = \alpha_0 + \alpha_1 * I(x_t > 0) + \beta_1 * x_t + B_2 * x_{t-1} + w_t$ has been fitted. The coefficients got are the following ones:

```
1
2 > summary(lmmodel)
3
4 Call:
5 lm(formula = y ~ . + rnorm(length(x_t), 0, 1), data = myframe)
6
7 Residuals:
8       Min        1Q    Median        3Q       Max
9 -0.18223  -0.02056   0.00120   0.02129   0.34441
```

```
10
11 Coefficients:
12                          Estimate Std. Error t value Pr(>|t|)
13 (Intercept)              -0.006175   0.003468  -1.780   0.0756 .
14 x_t                       0.692673   0.058878  11.765   <2e-16 ***
15 x_{t-1}‘                 0.015964   0.038796   0.412   0.6809
16 x_tpos                    0.012691   0.005544   2.289   0.0225 *
17 rnorm(length(x_t), 0, 1) -0.002285   0.001810  -1.262   0.2074
18 ---
19
20 Residual standard error: 0.042 on 538 degrees of freedom
21 Multiple R-squared:  0.4466,  Adjusted R-squared:  0.4425
22 F-statistic: 108.6 on 4 and 538 DF,  p-value: < 2.2e-16
```

It looks like x_t is the most significant coefficient since it also affects more the value of our regression. Moreover, the variable for positive values of x_{tpos} seems to indicate also a great influence according to pvalues. It means that information that affects oil directly affects gas, making the same investors in the market affect prices at the same time for both variables. This is logical since both resources are related to energy and investments on portfolios for that probably have some distribution of money allocated to each one.

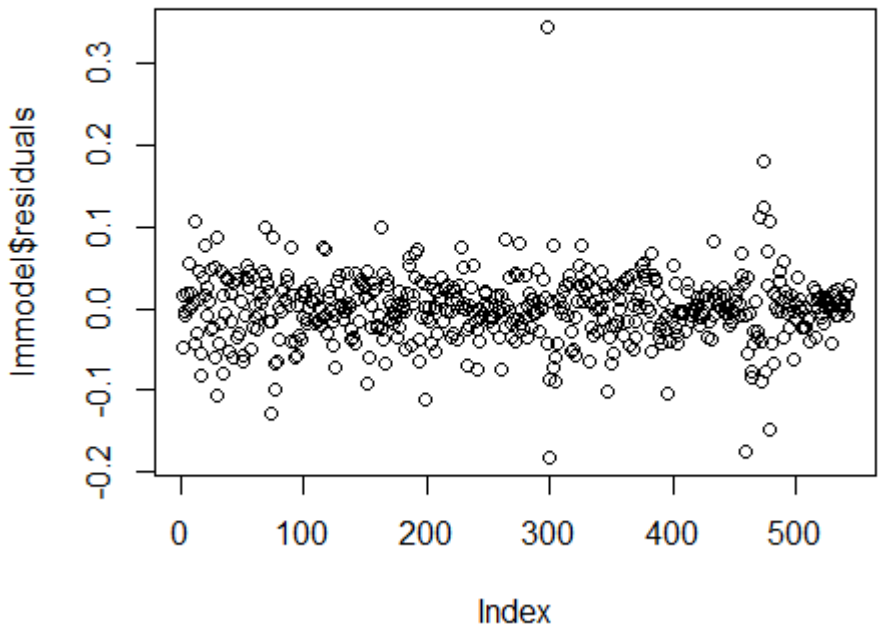

Figure 22: Plot of residuals for the model

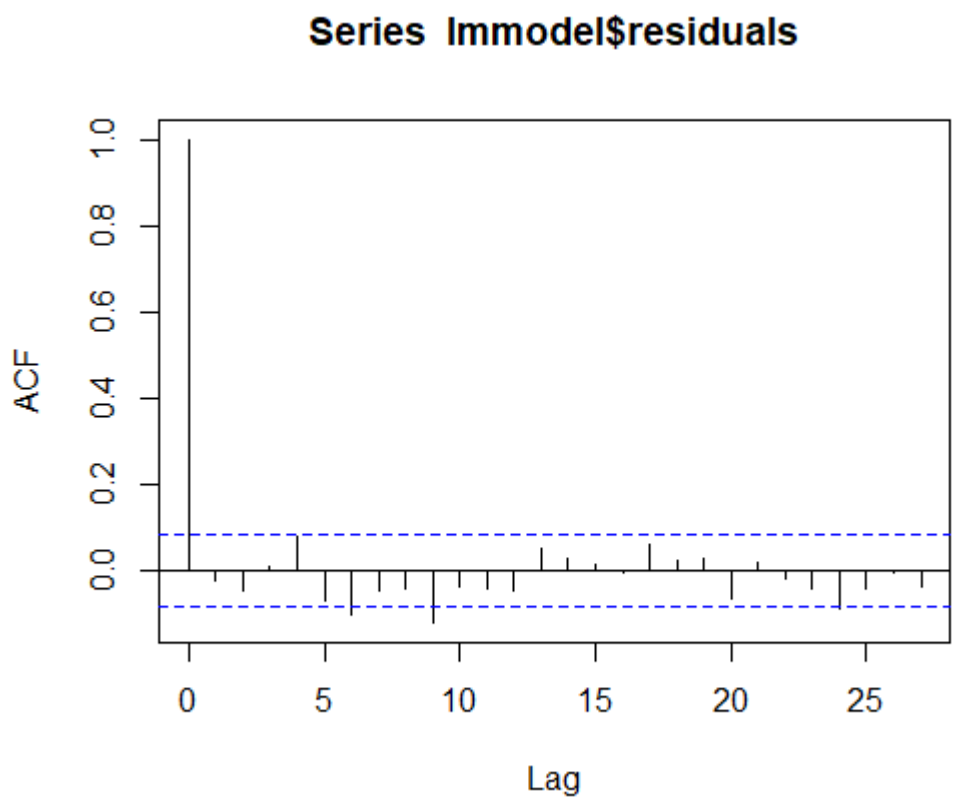

Figure 23: ACF of the residuals of the model

The residuals seems to be stationary around 0, without correlation for any more time lags than itself. Still, the relation between lags seems to still exist in a really small level, which is not significant. All in all, we could say that our residuals are stationary.

Contributions

All results and comments presented have been developed and discussed together by the members of the group.

Appendix

Poisson regression-the MCMC way

```
1
2  ###Time Series Lab 1
3  library(stats)
4  set.seed(12345)
5  ##1a
6  w_t<-rnorm(100)
7  x_t1<- integer(100)
8  x_t2<-integer(100)
9  for(t in 3:100){
10   x_t2[t] <- cos(2*pi*t/5)
11   x_t1[t] <- -0.8*x_t1[t-2]+w_t[t]
12 }
13 myfilter<- 0.2*(x_t1)
14 fx_t1<-stats::filter(x_t1, filter = rep(0.2,5), method = "convolution")
15 fx_t2<-stats::filter(x_t2, filter = rep(0.2,5), method = "convolution")
16
17
18
19 par(mfrow=c(1,1))
20 plot(x_t1, type = "l", col= "red", ylim=c(-2,4), lwd= 2)
21 lines(fx_t1, type = "l", col= "black")
22 legend("topright", # places a legend at the appropriate place
23        c("without filter", "with filter"), # puts text in the legend
24        lty=c(1,1), # gives the legend appropriate symbols (lines)
25        lwd=c(2.5,2.5),col=c("red","black")) # gives the legend lines the correct color and
              width
26
27
28 plot(x_t2, type = "l", col= "red", ylim=c(-2,4))
29 lines(fx_t2, type = "l", col= "black")
30 legend("topleft", # places a legend at the appropriate place
31        c("without filter", "with filter"), # puts text in the legend
32        lty=c(1,1), # gives the legend appropriate symbols (lines)
33        lwd=c(2.5,2.5),col=c("red","black")) # gives the legend lines the correct color and
              width
34
35
36 par(mfrow=c(1,1))
37
38 ##b
39
40 #HINT1: x_t is related to AR
41 #HINT2: w_t is related to MA
42 z1<- c(1,-4,2,0,0,1)
43 ##causal = the absolute value of the roots is >1
44
45 causal1 <- abs(polyroot(z1)) #polyroot solves the polyomial
46 causal1>1 #if(causal>1){print("It is causal")}else{print("it is NOT causal")}
47 #There are 2 falses.
48
49 z2<- c(1,0,3,0,1,0,-4)
50 causal2 <- abs(polyroot(z2))
51 causal2>1
52
53
54 #c
55 set.seed(54321)
56 ## in order to write it as an arima model, we need to leave the data in the following form:
57 ## x_t = phi*x_{t-1}...+pi_p*x{t-p}+w_t+ theta_1*w_{t-1}+...+theta_q*w{t-q}, recall that w_t
       does not have coefficient, so in the arima model will be 0
58
59 model<-arima.sim(n = 100, model = list(ar = c(-3/4), ma=c(0,-1/9)))
60
61 #theoretical
62 theoreticARMA<-ARMAacf(ar = c(-3/4), ma=c(0,-1/9), lag.max = 20)
63 plot(theoreticARMA, type = "h")
64 acf(model , lag.max = 100)
65 #if the result is inside the acf plot lines, it means that there might be white noise
66
67
68 #2
69 #a
70 library(astsa)
71 Data<-read.csv("C:/Users/Carles/Desktop/MasterStatistics-MachineLearning/Master_subjects/Time_
       Series_Analysis/Rhine.csv", sep =";", dec = ",")
72 tsdata<-ts(data = Data$TotN_conc)
73 plot(tsdata)
74 lag1.plot(tsdata, max.lag = 12)
75 ?lag1.plot
76 rho <- acf(tsdata, type="correlation", plot=T)
77 par(mfrow=c(3,4))
```

```
for(i in 1:12){
  plot(tsdata,lag(x = tsdata, k = i))
}
par(mfrow=c(1,1))


##b
mydata<- cbind(X= 1:168, y=tsdata)
fit = lm(y~X,mydata) # generate linear model
summary(fit)
par(mfrow=c(1, 1))
plot(resid(fit), type='o', main="Detrended")
plot(diff(tsdata), type='o', main='First Difference')

par(mfrow=c(3, 1))
acf(tsdata, 48, main='gtemp')
acf(resid(fit), 48, main='detrended')
acf(diff(tsdata), 48, main='first differences')

tsdata
par(mfrow=c(1,1))

#c
ks2<-ksmooth(time(tsdata), tsdata, 'normal', bandwidth=2)
ks10<-ksmooth(time(tsdata), tsdata, 'normal', bandwidth=10)
plot(tsdata, type='p')
lines(ks2, col = "green")
lines(ks10, col = "red")
legend("topright", # places a legend at the appropriate place
       c("Kernel = 2", "Kernel = 10"), # puts text in the legend
       lty=c(1,1), # gives the legend appropriate symbols (lines)
       lwd=c(2.5,2.5),col=c("green","red")) # gives the legend lines the correct color and
            width

residual2<- ks2$y-tsdata
residual10<- ks10$y-tsdata
par(mfrow= c(1,1))
plot(residual2)
plot(residual10)
acf(residual2)
acf(residual10)

##d


mynewdata1<-cbind(mydata, rep(1:12,14))
colnames(mynewdata1)<- c("x", "y", "month")
res<-lm(y~., data=mynewdata1)
acf(res$residuals)
plot(res$residuals, type = "l")



##e

step<-step(res,direction="both")
step$coefficients
AIC(step)
plot(step$coefficients)
acf(step$residuals)
summary(step)
summary(res)



AIC(step)
AIC(res)

##3


library(astsa)
#a
plot(oil, type = "l", col= "red", ylim=c(min(oil), max(gas)), lwd= 3, ylab = "prices")
lines(gas, type = "l", col= "black")
legend("topleft", # places a legend at the appropriate place
       c("oil", "gas"), # puts text in the legend
       lty=c(1,1), # gives the legend appropriate symbols (lines)
       lwd=c(2.5,2.5),col=c("red","black")) # gives the legend lines the correct color and
            width


#b

logoil<-log(oil)
loggas<-log(gas)
```

```
plot(logoil, type = "l", col= "red", ylim=c(min(logoil), max(loggas)), lwd= 3, ylab = "
    logprices")
lines(loggas, type = "l", col= "black")
legend("topleft", # places a legend at the appropriate place
       c("oil", "gas"), # puts text in the legend
       lty=c(1,1), # gives the legend appropriate symbols (lines)
       lwd=c(2.5,2.5),col=c("red","black")) # gives the legend lines the correct color and
           width

##c
par(mfrow= c(1,1))
difflogoil<-diff(logoil)
diffloggas<-diff(loggas)
plot(difflogoil, type='o', main='First Difference oil')
plot(diffloggas, type='o', main='First Difference gas')
#par(mfrow=c(3, 1))
acf(diffloggas, 48, main="first differences gas")
acf(difflogoil, 48, main="first differences oil")


x_t<-difflogoil
y_t<-diffloggas
##d
par(mfrow=c(2, 2))

length(x_t)
length(y_t)
lags<-c(0,1,2,3)

for(lag in lags){
  xlag <- lag(x_t, -lag, na.pad = TRUE)

  length(y_t)
  smoother<-ksmooth(xlag,y_t, bandwidth = 0.1)
  plot(x= xlag, y = y_t, main = paste0("lag = ", lag),
       xlab = paste0("log oil lag = " ,lag), ylab = "log gas")
  lines(smoother, col = "red")

}

##e
set.seed(12345)
par(mfrow= c(1,1))
myframe<- ts.intersect(y_t = y_t, x_t = x_t,lag =stats::lag(x_t, k =1), I= as.ts(x_t>0))

colnames(myframe)<- c("y", "x_t", "x_{t-1}","x_tpos")
lmmodel<-lm(y~.+rnorm(length(x_t),0,1), data = myframe)
summary(lmmodel)

lmmodel$coefficients
acf(lmmodel$residuals)
plot(lmmodel$residuals)
```