

# 732A62 Lab 1

*Emil K Svensson & Rasmus Holm*

2017-09-12

## Assignment 1

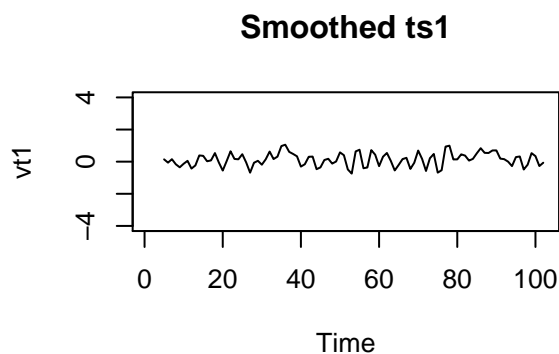
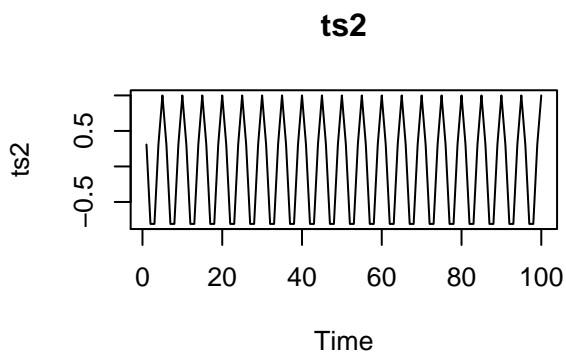
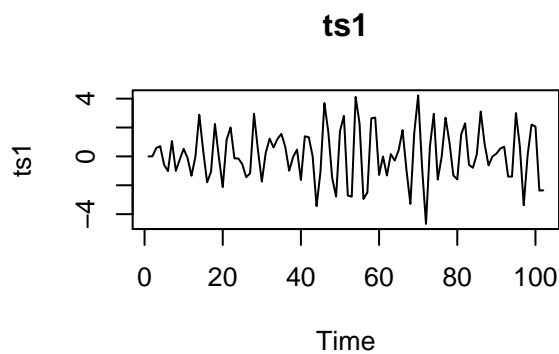
a)

```
set.seed(12345)
t1 <- c(0, 0, rnorm(100, 0, 1))
ts1 <- filter(t1, filter = c(0, -0.8),
              method = "recursive", sides = 1)

t2 <- 1:100
ts2 <- cos(2 * pi * t2 / 5)

vt1 <- filter(ts1, filter = c(rep(0.2, times = 5)),
              method = "convolution", sides = 1)

vt2 <- filter(ts2, filter = c(rep(0.2, times = 5)),
              method = "convolution", sides = 1)
```



Time series 1 (ts1) show no noticable change in its random pattern except the scale which is transformed in to a smaller scale. Time series 2 (ts2) is flattened by the soothing filter and all values are now basicly 0. This is because the average of ts2 lies around zero it is also resonable to expect that a moving average smoother would generate the same (or similar) result.

b)

```

leftside <- c(1, -4, 2, 0, 0, 1) # the x's
rightside <- c(1, 0, 3, 0, 1, 0, -4) # The w's

causal <- polyroot(leftside) #Not causal
invertible <- polyroot(rightside) #Non invertible

complex_dist <- function(x) {
  sqrt(Re(x)^2 + Im(x)^2)
}

print("The causal")

## [1] "The causal"
sapply(causal, complex_dist)

## [1] 0.2936658 1.6793817 1.0000000 1.4239626 1.4239626
print("The invertible")

## [1] "The invertible"
sapply(invertible, complex_dist)

## [1] 0.6874372 0.6874372 0.6874372 0.6874372 1.0580446 1.0580446

```

Since both parts contains values below 1 they are inside the unit circle and therefor are not causal nor invertible.

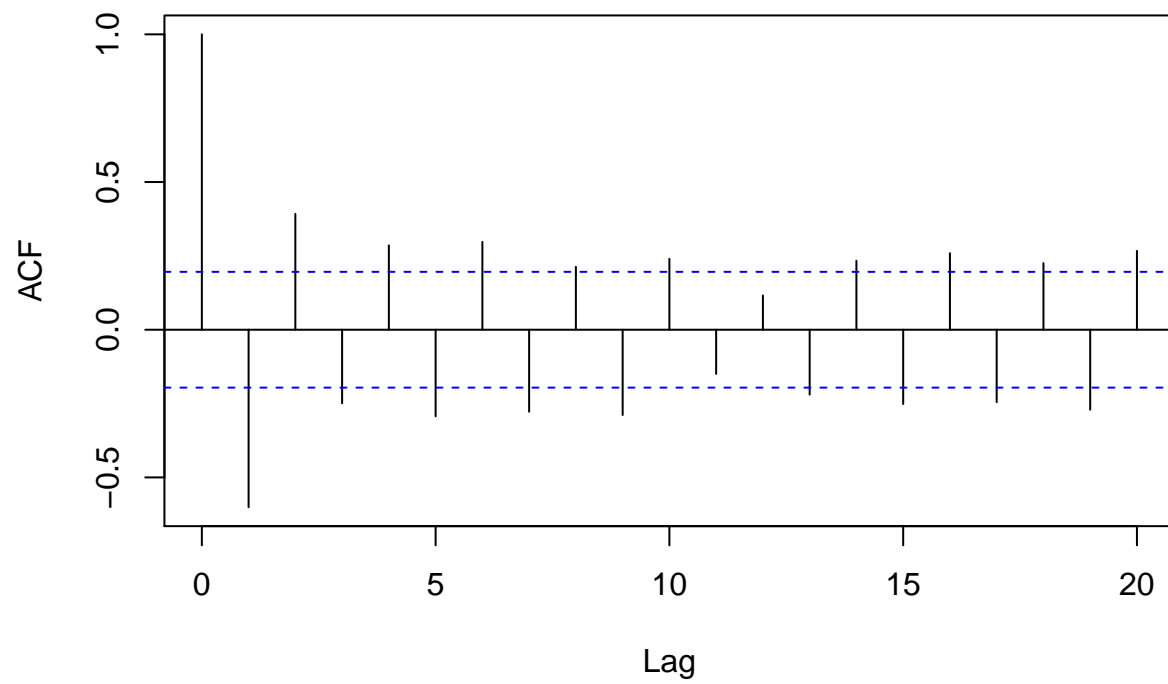
c)

```

set.seed(54321)
model1c <- arima.sim(n = 100, list(ar = c(-3 / 4), ma = c(0, -1 / 9) ))
acf(model1c)

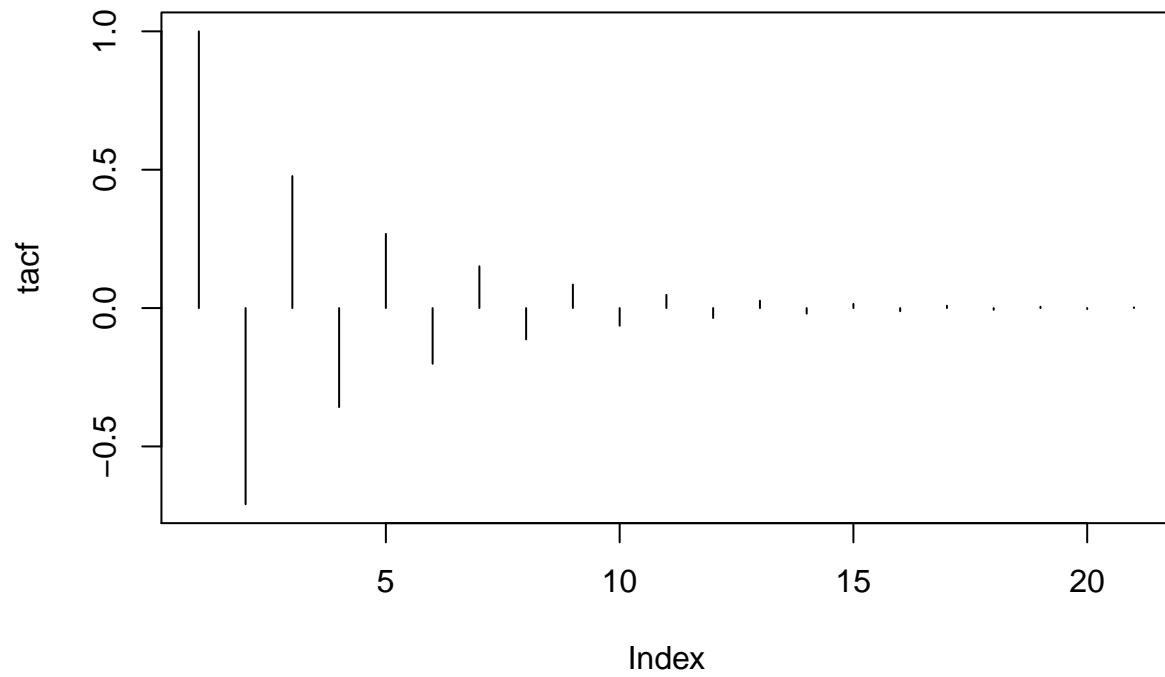
```

### Series model1c



```
tacf <- ARMAacf(ar=c(-3 / 4), ma=c(0, -1/9), lag.max=20)
plot(tacf, type="n", main="Theoretical")
segments(1:length(tacf), rep(0, length(tacf)), 1:length(tacf), tacf)
```

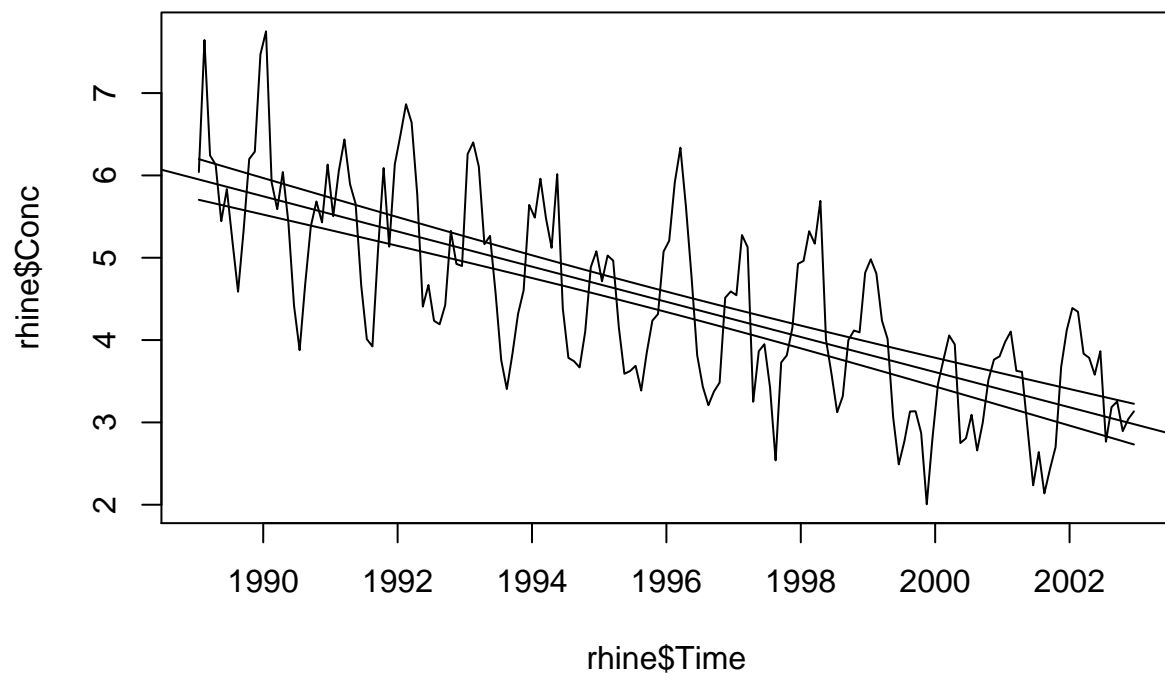
## Theoretical

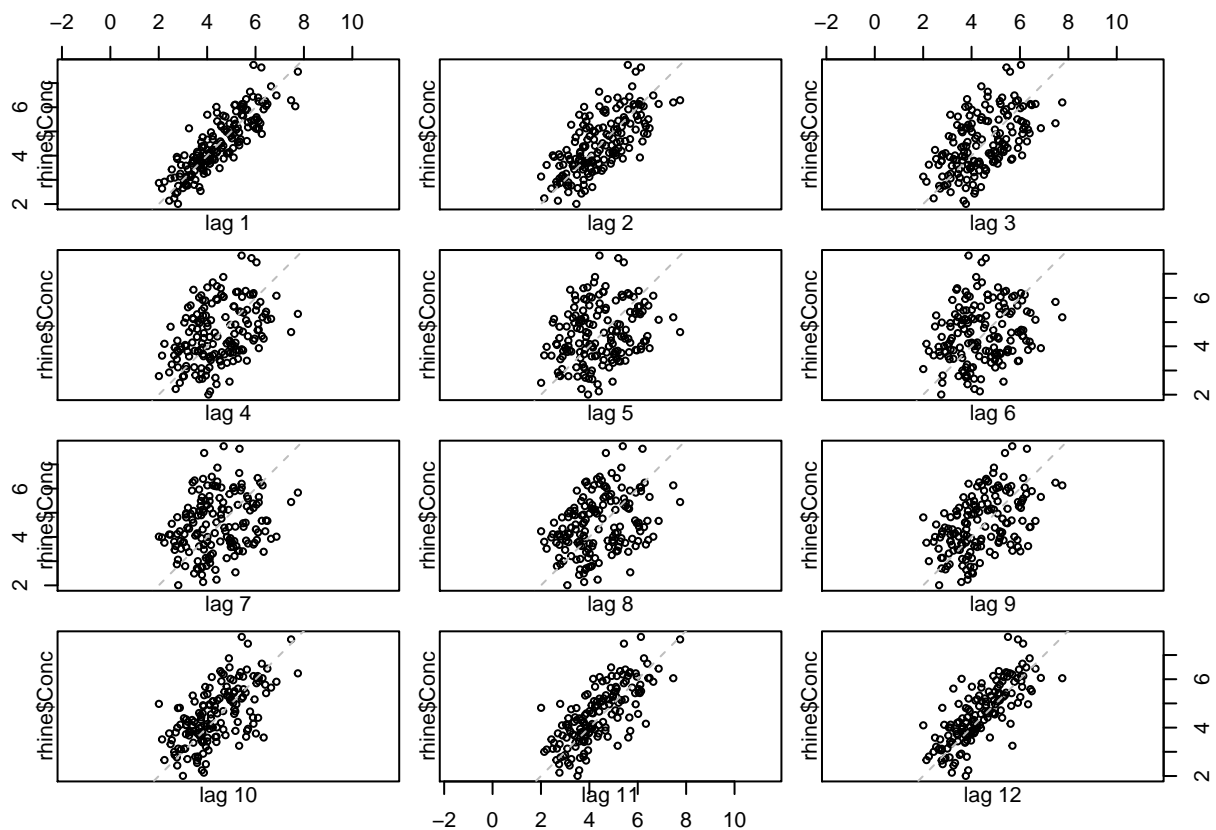


## Assignment 2

a)

```
rhine <- read.csv2("../data/Rhine.csv")  
colnames(rhine)[4] <- "Conc"
```





b)

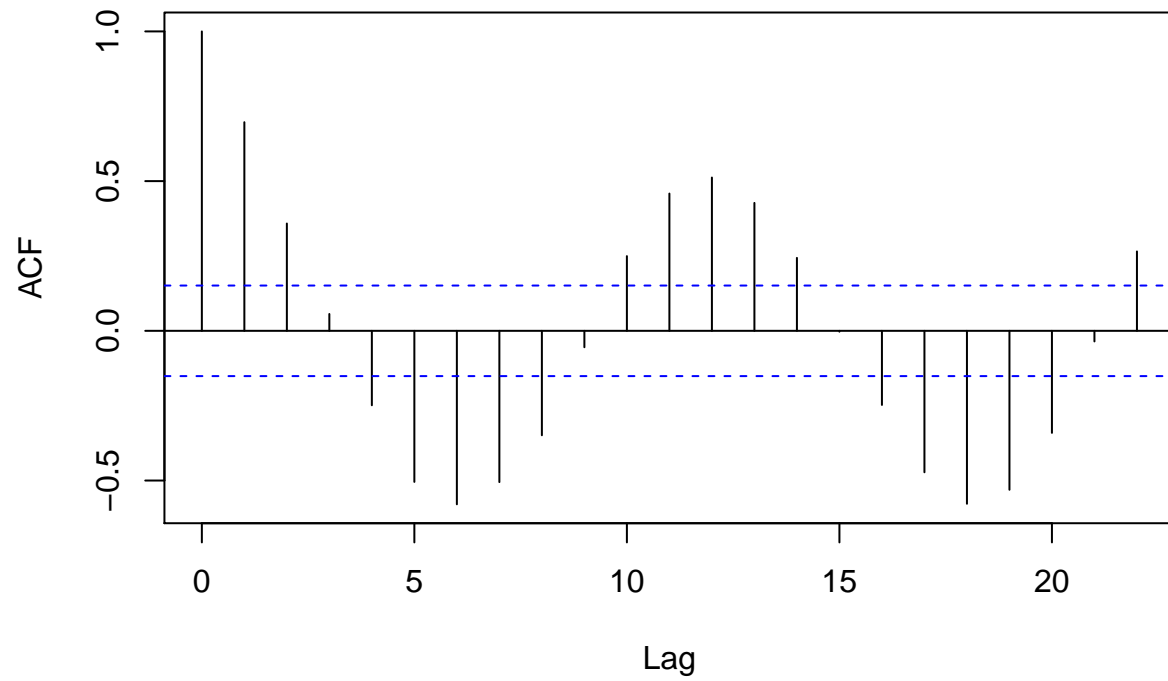
```
lmobj <- lm(Conc ~ Time, data = rhine)
predobj <- predict(lmobj, se.fit = TRUE)
```

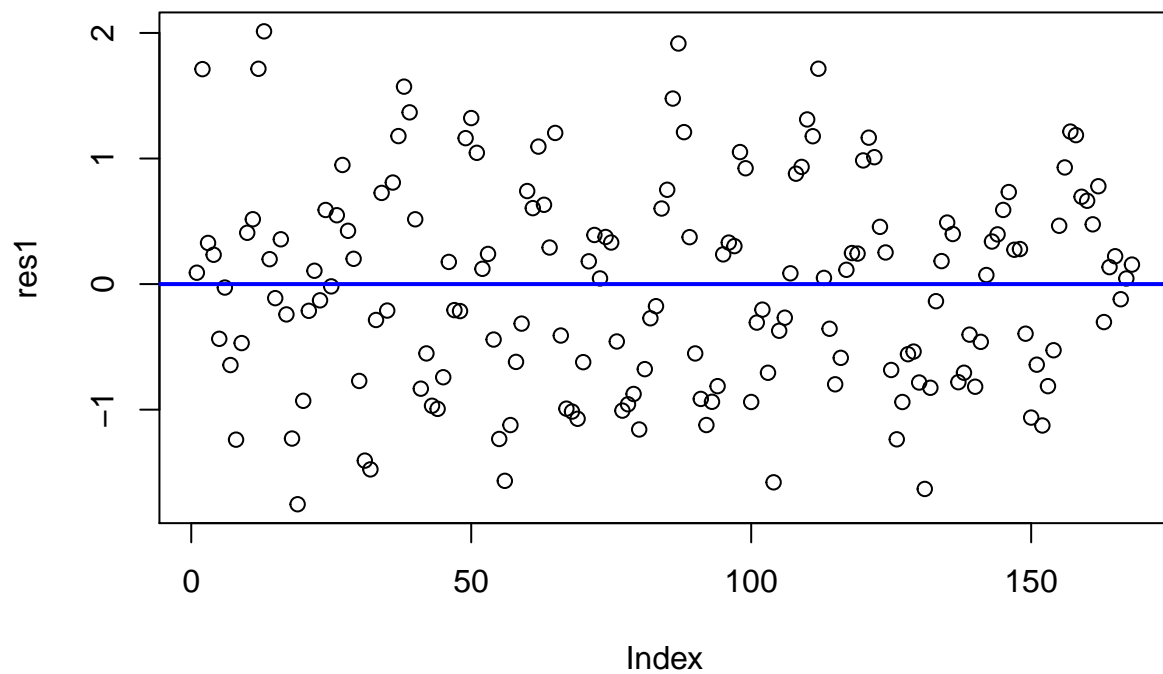
```
res1 <- residuals(lmobj)
summary(lmobj)
```

```
##
## Call:
## lm(formula = Conc ~ Time, data = rhine)
##
## 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)  430.70725   31.26570   13.78  <2e-16 ***
## Time         -0.21355    0.01566  -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

### Series res1

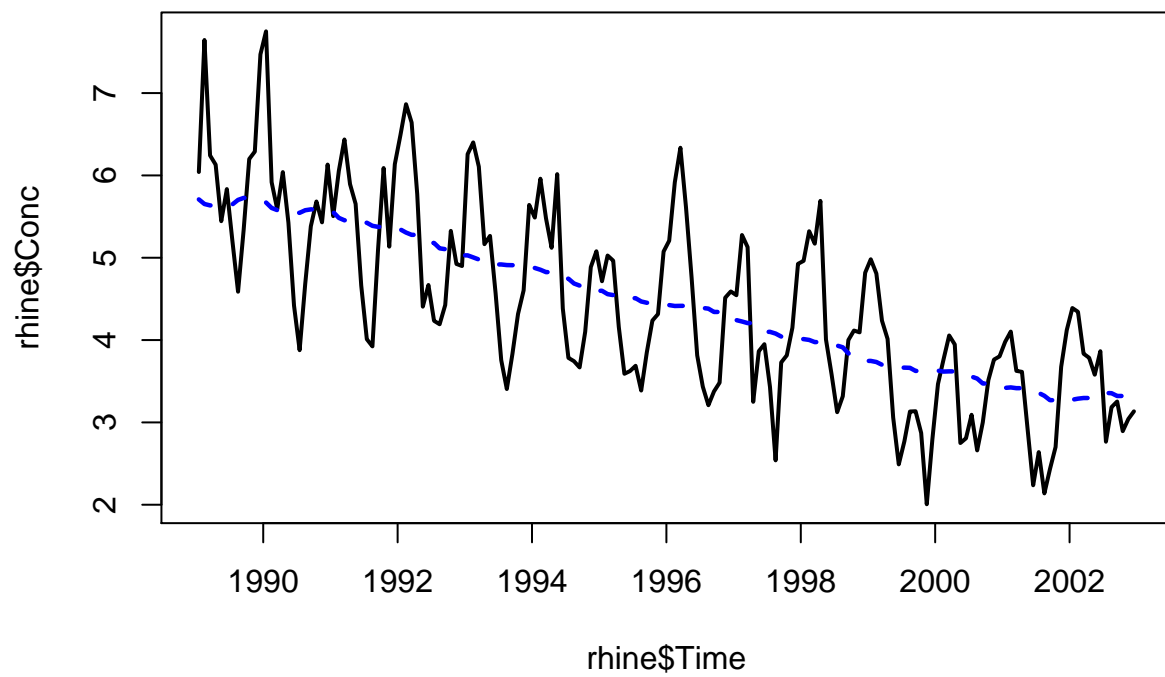


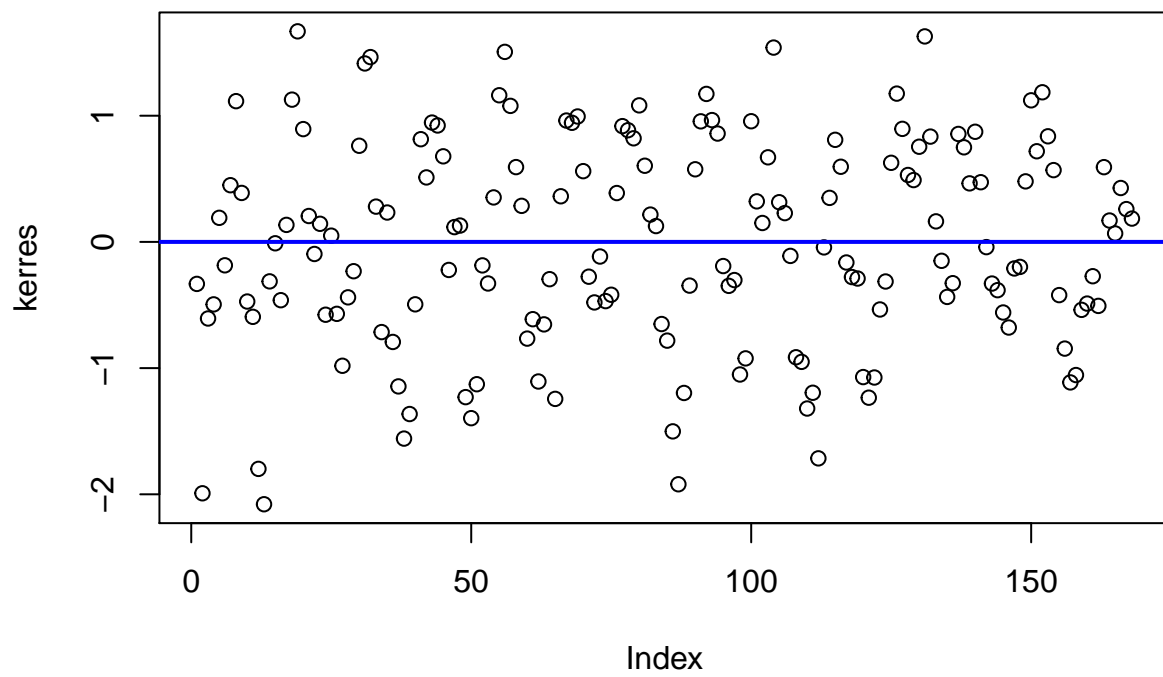


c)

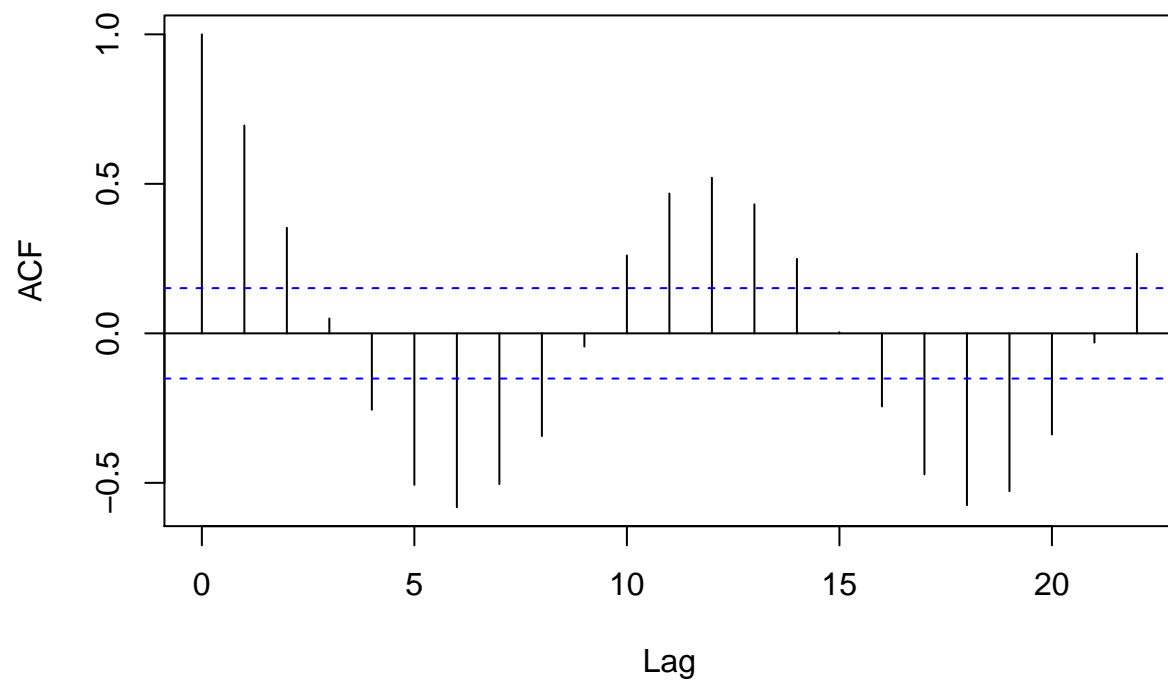
```
kersmo <- ksmooth(y = rhine$Conc, x = rhine$Time, bandwidth = 5)
kerres <- kersmo$y - rhine$Conc
```





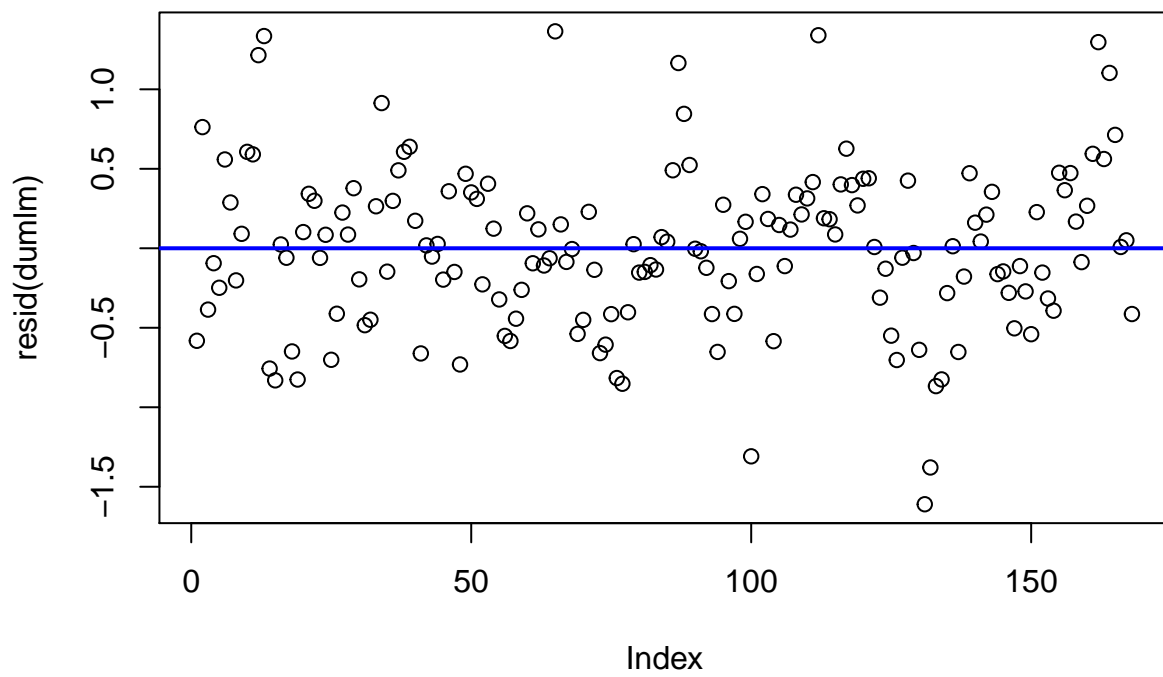


### Series kerres

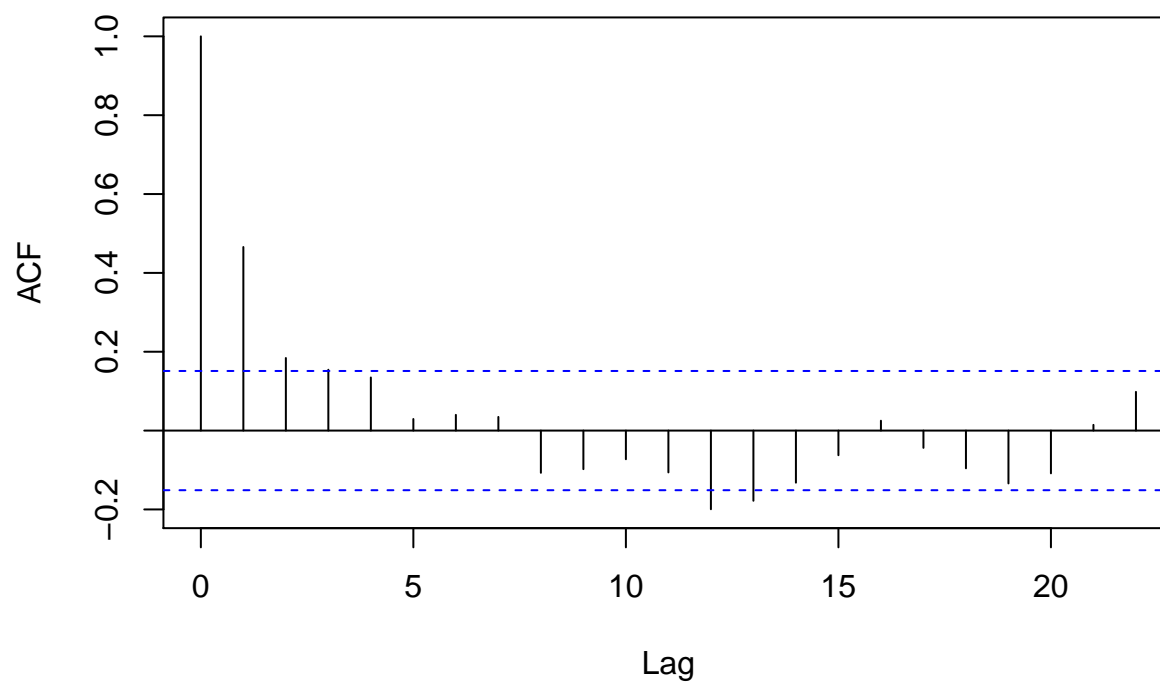


d)

```
rhine$Month.f <- as.factor(rhine$Month)
dumlm <- lm(Conc ~ Time + Month.f, rhine)
```



### Series resid(dumlm)



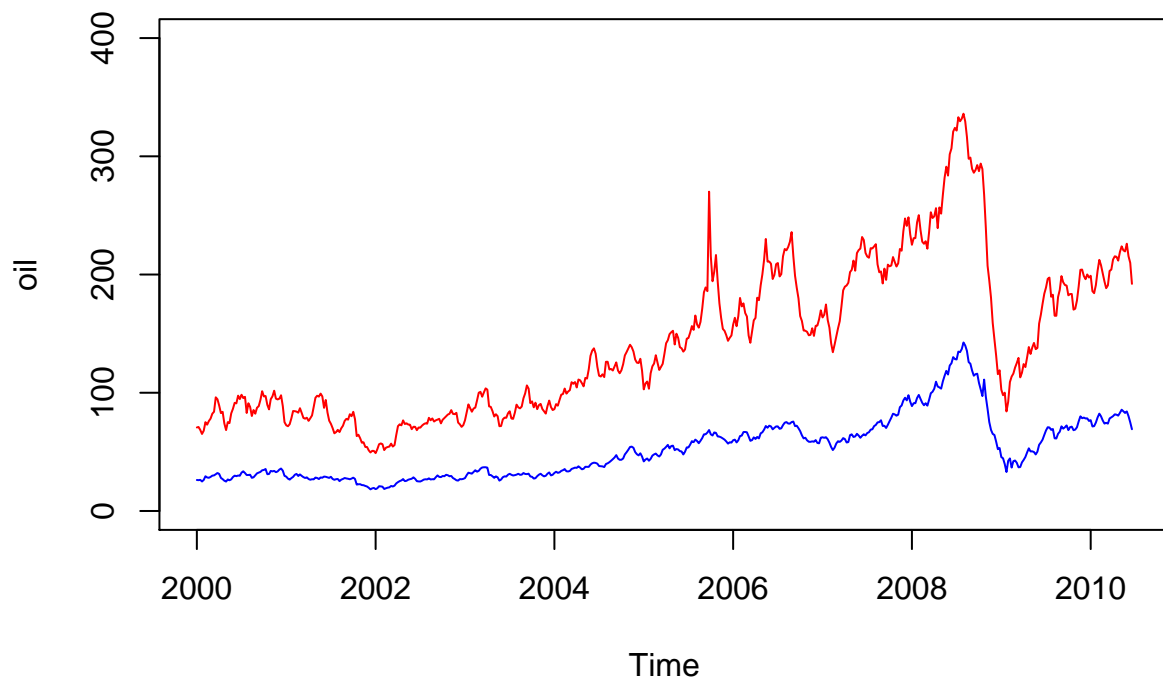
e)

```
library(MASS)
stepAIC(dumlm, direction = "backward", steps = 1000)
```

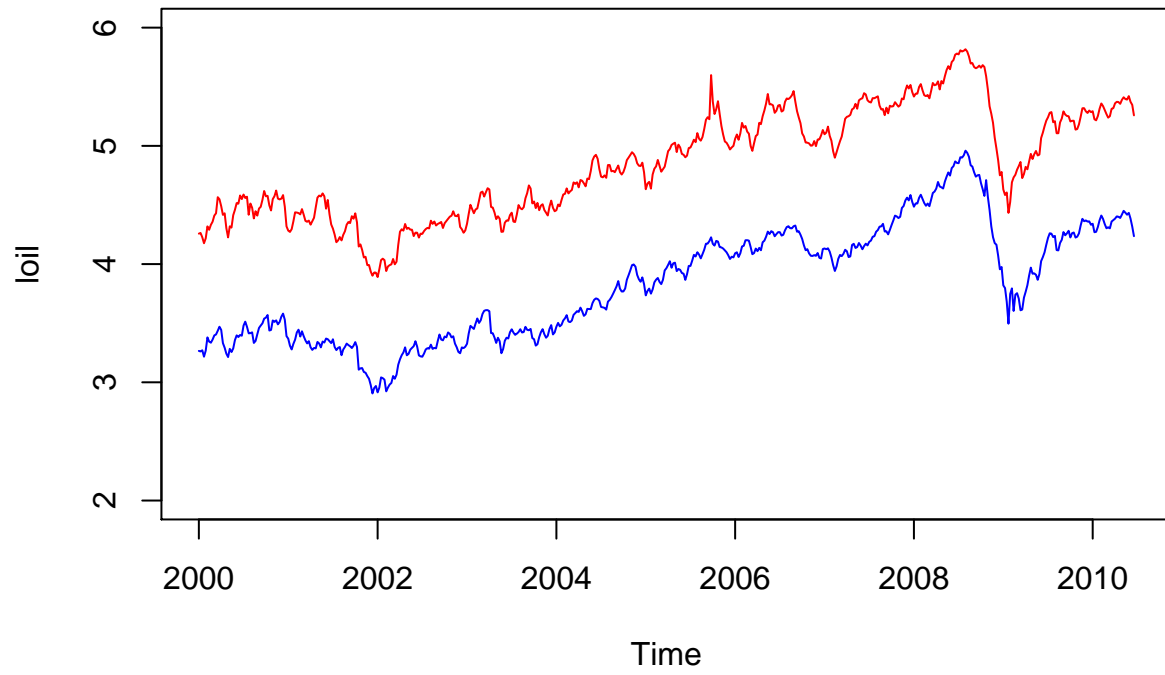
```
## Start:  AIC=-202.02
## Conc ~ Time + Month.f
##
##           Df Sum of Sq    RSS      AIC
## <none>                43.237 -202.023
## - Month.f 11      68.524 111.761  -64.477
## - Time     1     118.387 161.624   17.499
##
## Call:
## lm(formula = Conc ~ Time + Month.f, data = rhine)
##
## Coefficients:
## (Intercept)      Time    Month.f2    Month.f3    Month.f4
##  420.82746   -0.20824    0.27659    0.04006   -0.34643
##   Month.f5    Month.f6    Month.f7    Month.f8    Month.f9
##  -0.86165   -1.26114   -1.60808   -1.71242   -1.23669
##   Month.f10   Month.f11   Month.f12
##  -0.87446   -0.75127   -0.17745
```

## Assignment 3

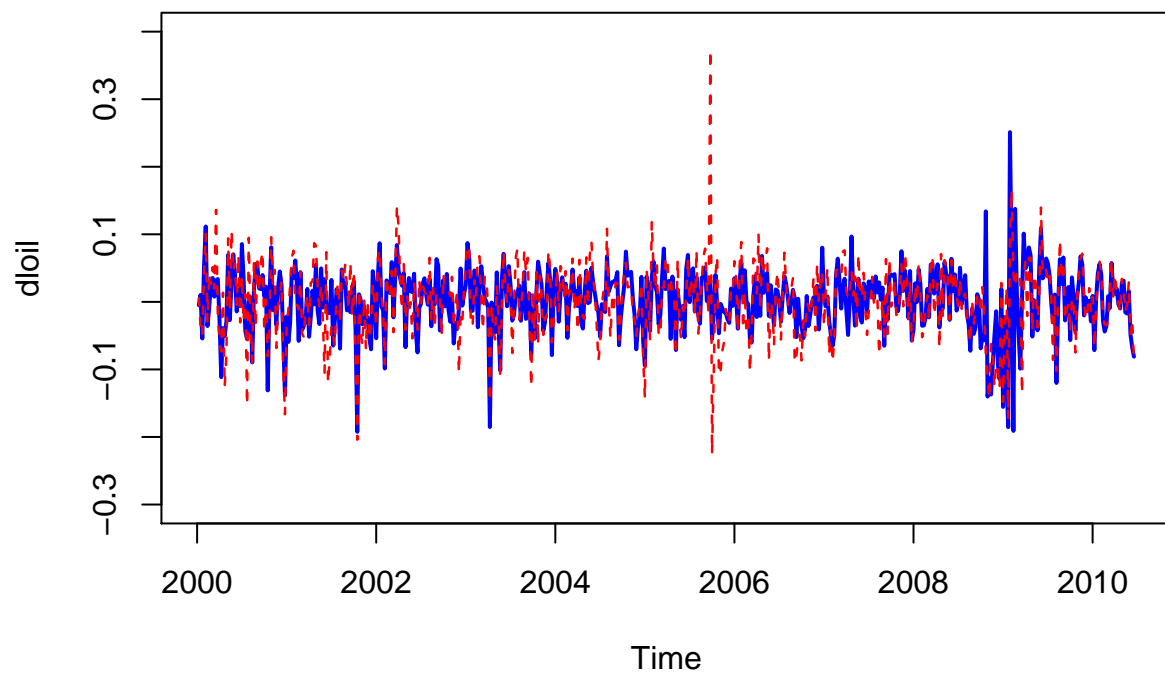
a)



b)

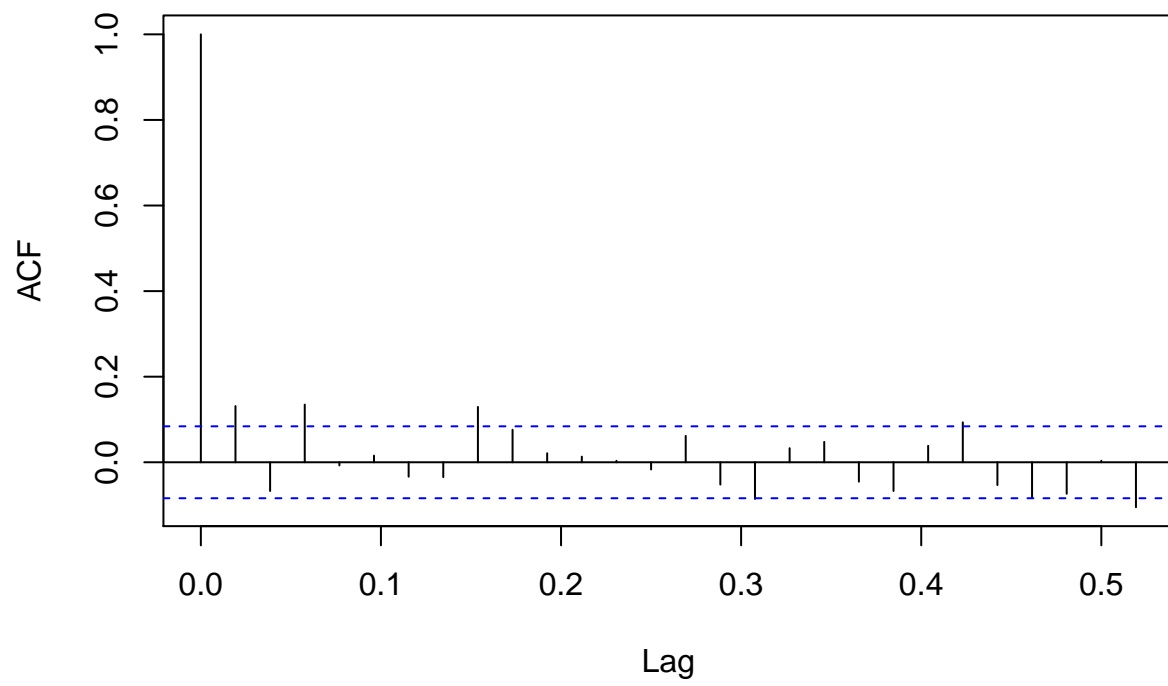


c)

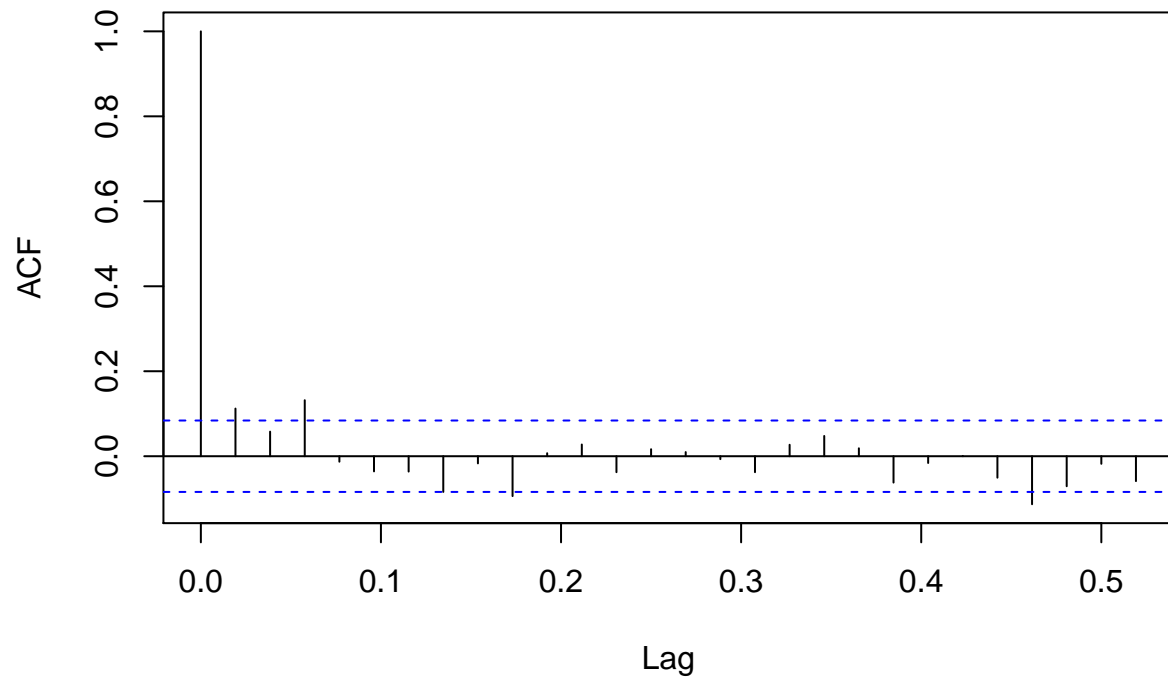




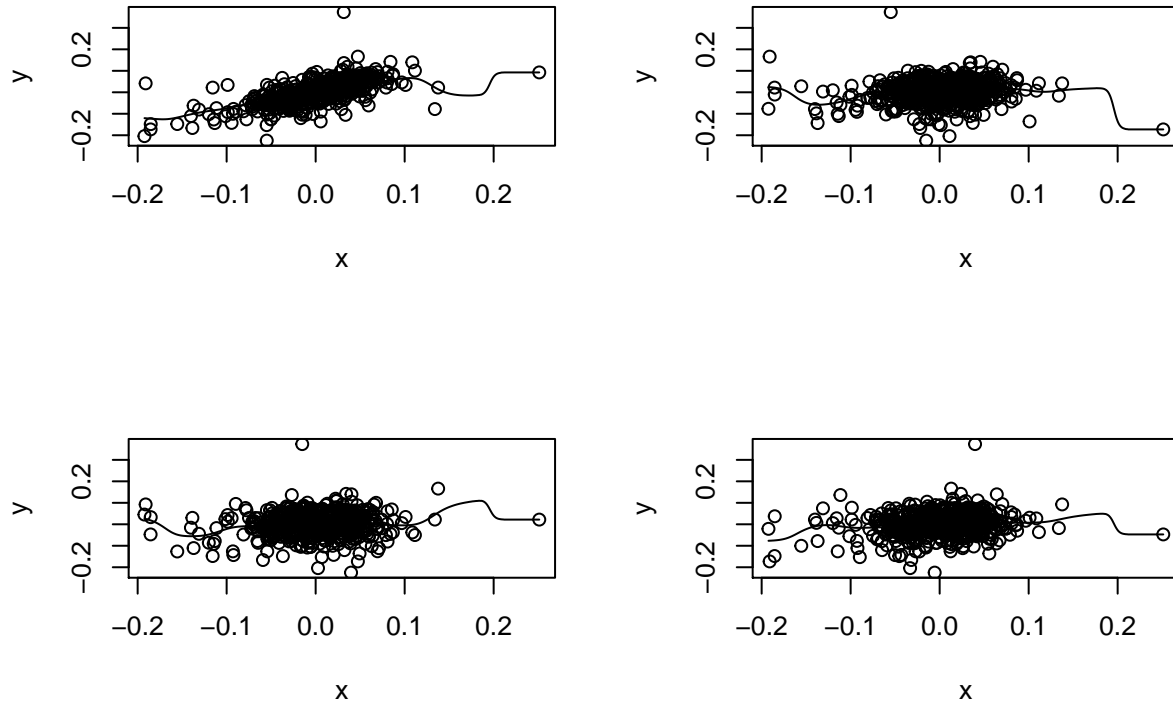
### Series dloil



**Series dlgas**



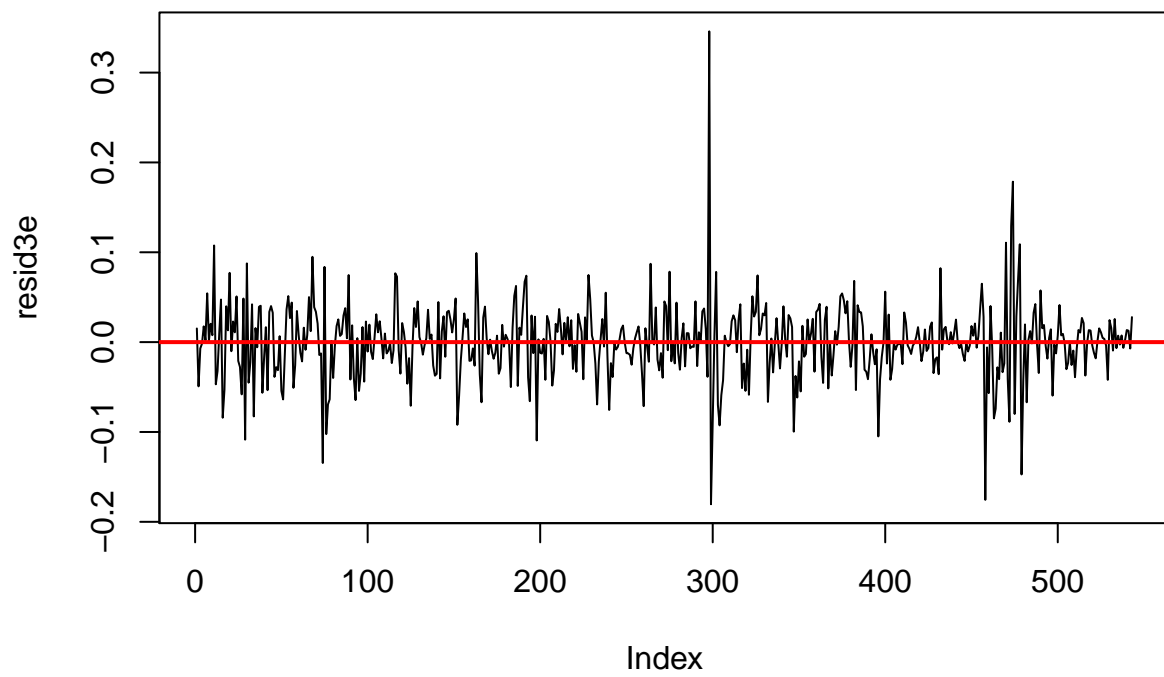
d)



e)

```
xt <- dloil
yt <- dlgas

tss <- ts.intersect(yt=yt, xt=xt, lag1xt=lag(xt, 1), dummy=xt > 0)
model3e <- lm(yt ~ xt + lag1xt + dummy, data = tss)
resid3e <- resid(model3e)
```



**Series resid3e**

