Exercise 8

Philipp Drebes

20.04.2023

Exercise 8.1

As in exercise 7.1, we would like to use this exercise to simulate several time series by means of an ARMA model. Please perform the same steps as in exercise 5.1 for the following models. The innovation E_t shall follow a standard normal distribution $\mathcal{N}(0;1)$ in every model.

a) Why is it not possible to simulate the ARIMA(2,1,2) model with coefficients $\alpha_1 = 0.5, \alpha_2 = 0.5, \beta_1 = -0.4$ and $\beta_2 = 0.3$ with $d = 1$ with arima.sim?

```
set.seed(989898)

# plot(arima.sim(n = 200, model = list(ar = c(0.5, 0.5), ma = c(-0.4, 0.3), d = 1)))

# Error in `arima.sim(n = 200, model = list(ar = c(0.5, 0.5), ma = c(-0.4, 0.3), d = 1))`:
# 'ar' part of model is not stationary
```

It's not possible to simulate the model with arima.sim, because the auto regressive part of the model is not stationary. X_1, X_0 and X_{-1} are not defined, so R assumes X_0 and X_{-1} to be 0. R uses a 'burn-in' period and discards this data before continuing the simulation[1]. If I understood this post on Stackexchange[2] correct, this will only work if the conditional distribution implies stationarity, because then the process will converge from the stationarity distribution in long run.

b) What is the equivalent ARMA model to the ARIMA(2,1,2) model in task c)?

It will be an ARMA(3,2) model.

$$(1 - 0.5B - 0.5B^2)(1 - B)X_t = (1 - 0.4B + 0.3B^2)E_t$$
$$(1 - 1.5B + 0.5B^3)X_t = E_t - 0.4E_{-1} + 0.3E_{-2}$$
$$X_t = 1.5X_{-1} + 0X_{-2} - 0.5X_{-3} + E - 0.4E_{-1} + 0.3E_{-2}$$

ARMA(3,2) with

$$\alpha_1 = 1.5, \quad \alpha_2 = 0, \quad \alpha_3 = -0.5, \quad \beta_1 = -0.4, \quad \beta_2 = 0.3$$

Polyroots of the ARIMA(2,1,2) model

```
abs(polyroot(c(1, -0.5, -0.5)))
```

```
## [1] 1 2
```

Polyroots of the ARMA(3,2) model

```
abs(polyroot(c(1, -1.5, 0, 0.5)))
```

```
## [1] 1 1 2
```

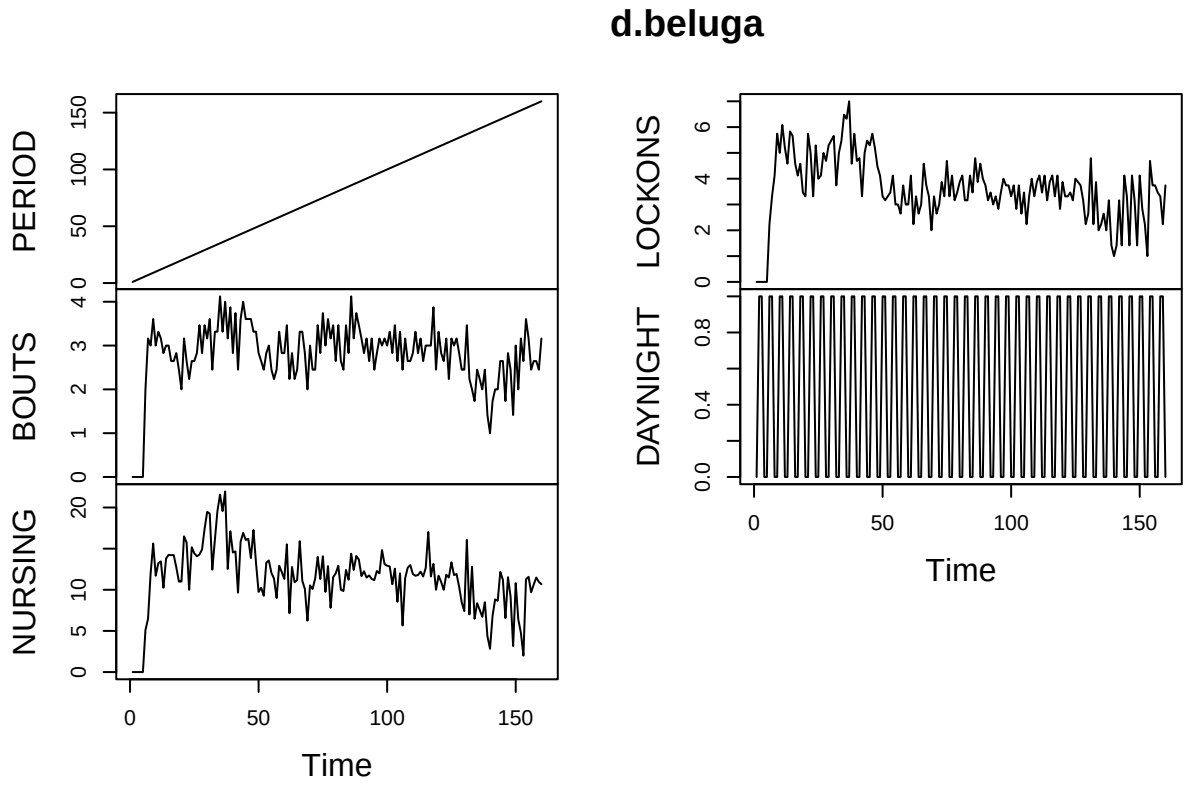
Exercise 8.2

There is a study on the development of beluga whales that focusses on the nursing behaviour of mother and calf. During a total of 160 time periods (each lasting 6 hours) subsequent to birth, the following variables were observed for Hudson, a beluga calf. Zoologists use this data to ascertain the health of this young whale. A short description of the data is given in the following table.

PERIOD	Index of time period
BOUTS	Square root of the number of nursing bouts
LOCKONS	Square root of the number of lock-ons (docking attempts)
DAYNIGHT	Day (1, 8am - 8pm) or night (0, 8pm - 8am) indicator
NURSING	Square root of the number of seconds spent successfully nursing during the period.

A nursing bout is defined as a successful nursing episode where milk was obtained. We would like to model the nursing time by means of the other variables. Count variables have already undergone a square root transformation to stabilize their variance (first-aid-transformation). You will find the data in the file *beluga.dat*. Load the data in the usual way and create a time series matrix:

```
d.beluga <- read.table("http://stat.ethz.ch/Teaching/Datasets/WBL/beluga.dat", header = TRUE)
d.beluga <- ts(d.beluga)
plot(d.beluga)
```



a) Fit the model

$$\text{NURSING} = \beta_0 + \beta_1 \text{PERIOD} + \beta_2 \text{BOUTS} + \beta_3 \text{LOCKONS} + \beta_4 \text{DAYNIGHT}$$

using ordinary linear regression. Check the independence of the residuals. What conclusions can zoologists draw from this analysis?

```
fit <- lm(NURSING ~ PERIOD + BOUTS + LOCKONS + DAYNIGHT, data = d.beluga)
summary(fit)
```

```
##
## Call:
## lm(formula = NURSING ~ PERIOD + BOUTS + LOCKONS + DAYNIGHT, data = d.beluga)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4457 -0.9018 -0.0850  1.0952  3.9548
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.5602842  0.5502170   1.018  0.31012
## PERIOD       0.0001998  0.0031937   0.063  0.95020
## BOUTS        0.8784967  0.3336237   2.633  0.00932 **
## LOCKONS      2.3903512  0.2035042  11.746  < 2e-16 ***
## DAYNIGHT    -0.3416237  0.2510156  -1.361  0.17550
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.582 on 155 degrees of freedom
## Multiple R-squared:  0.842,  Adjusted R-squared:  0.8379
## F-statistic: 206.5 on 4 and 155 DF,  p-value: < 2.2e-16
```

b) Due to the correlations involved, an AR(p) model should be assumed for the residuals. Determine the order p of this model, and estimate the parameters $\alpha_1, \ldots, \alpha_p$

```
r.burg <- ar(fit$residuals)
r.burg
```

```
##
## Call:
## ar(x = fit$residuals)
##
## Coefficients:
##      1       2
## 0.2837  0.3201
##
## Order selected 2  sigma^2 estimated as  1.831
```

The order p is 2 with $\alpha_1 = 0.2837$ and $\alpha_2 = 0.3201$

c) Estimate the regression coefficients and the AR parameters using Generalized Least Squares with Maximum Likelihood estimation.
To ensure convergence of the algorithm, known estimates of the AR parameters can be passed to corARMA() as starting values using the optional argument values. In this particular case, this does not change the outcome.(correlation = corARMA(..., value = r.burg$ar, ...))

```
library(nlme, quietly = T)
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
```

```
##   as.zoo.data.frame zoo
```

```
##
## Attaching package: 'forecast'
```

```
## The following object is masked from 'package:nlme':
##
##     getResponse
```

```
r.bel.gls <- gls(NURSING ~ BOUTS + LOCKONS + DAYNIGHT + PERIOD, data = d.beluga,
                 correlation =
                   corARMA(form = ~PERIOD, p = r.burg$order, value = r.burg$ar, q = 0, fixed = FALSE),
                 method = "ML")
summary(r.bel.gls)
```

```
## Generalized least squares fit by maximum likelihood
##   Model: NURSING ~ BOUTS + LOCKONS + DAYNIGHT + PERIOD
##   Data: d.beluga
##       AIC      BIC   logLik
##   560.396 584.9974 -272.198
##
## Correlation Structure: ARMA(2,0)
##  Formula: ~PERIOD
##  Parameter estimate(s):
##      Phi1      Phi2
## 0.2739973 0.3653664
##
## Coefficients:
##                   Value Std.Error   t-value p-value
## (Intercept)  1.3218876 0.7678369  1.721573  0.0871
## BOUTS        0.2961687 0.3370588  0.878686  0.3809
## LOCKONS      2.5681919 0.1964012 13.076254  0.0000
## DAYNIGHT    -0.3080292 0.1549160 -1.988363  0.0485
## PERIOD       0.0024982 0.0062754  0.398089  0.6911
##
##  Correlation:
##          (Intr) BOUTS  LOCKON DAYNIG
## BOUTS    -0.303
## LOCKONS  -0.101 -0.811
## DAYNIGHT -0.014 -0.135  0.067
## PERIOD   -0.607 -0.233  0.251  0.024
##
## Standardized residuals:
##        Min         Q1        Med         Q3        Max
## -2.8005548 -0.5876371  0.0173882  0.6560202  2.4985404
##
## Residual standard error: 1.577032
## Degrees of freedom: 160 total; 155 residual
```

```
d.resid <- ts(resid(r.bel.gls))
```

d) Optional: Simplify the model if possible.

e) Optional: What transformation should you apply to obtain a linear model with independent errors? State it as a formula.
Hint: Cochrane-Orcutt Method.

f) Optional: How would you perform this transformation (or these transformations) in R? Use the

transformed time series to carry out another regression, and look at the correlation structure for the errors!
R-Hint: lag().

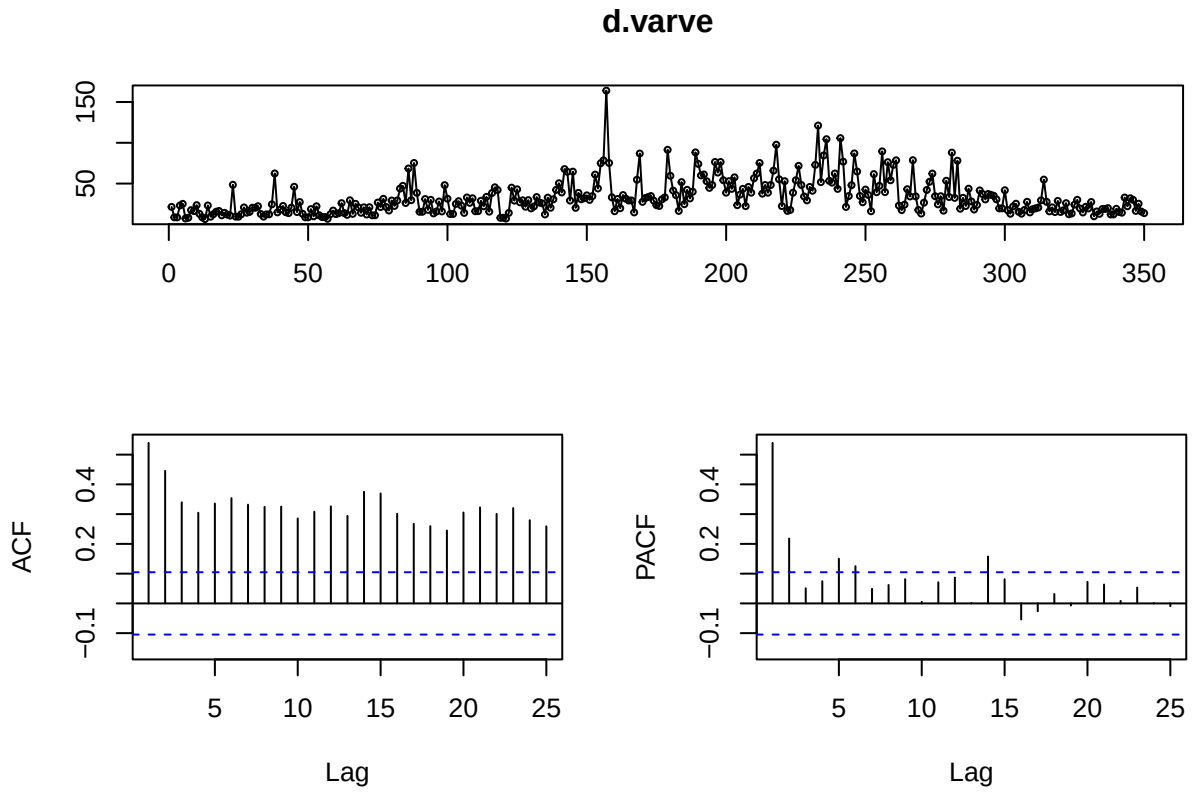
Exercise 8.3

With the new material in the course we would like to return to Exercise 7.3.

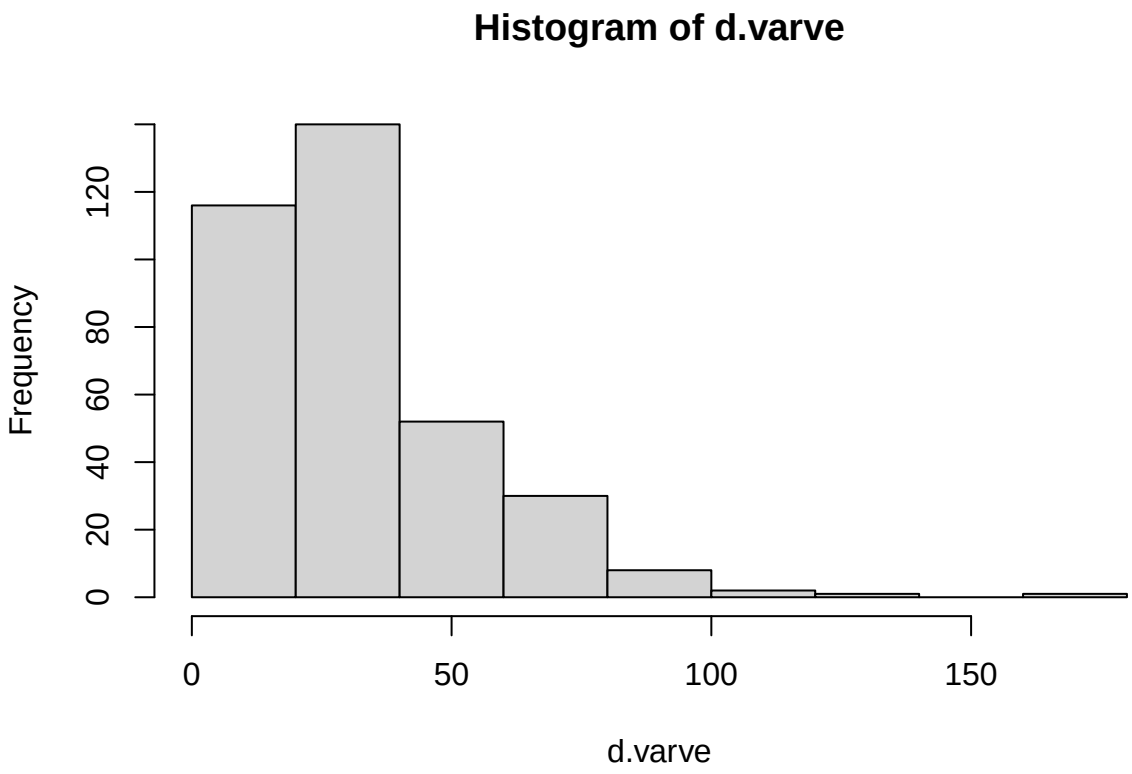
a) Choose a suitable model that fits the data. Does your model fit? Analyze the residuals and comment on your decision.

```
library(forecast)

t.url <- "http://stat.ethz.ch/Teaching/Datasets/WBL/varve.dat"
d.varve <- ts(scan(t.url)[201:550], frequency=1)
tsdisplay(d.varve)
```

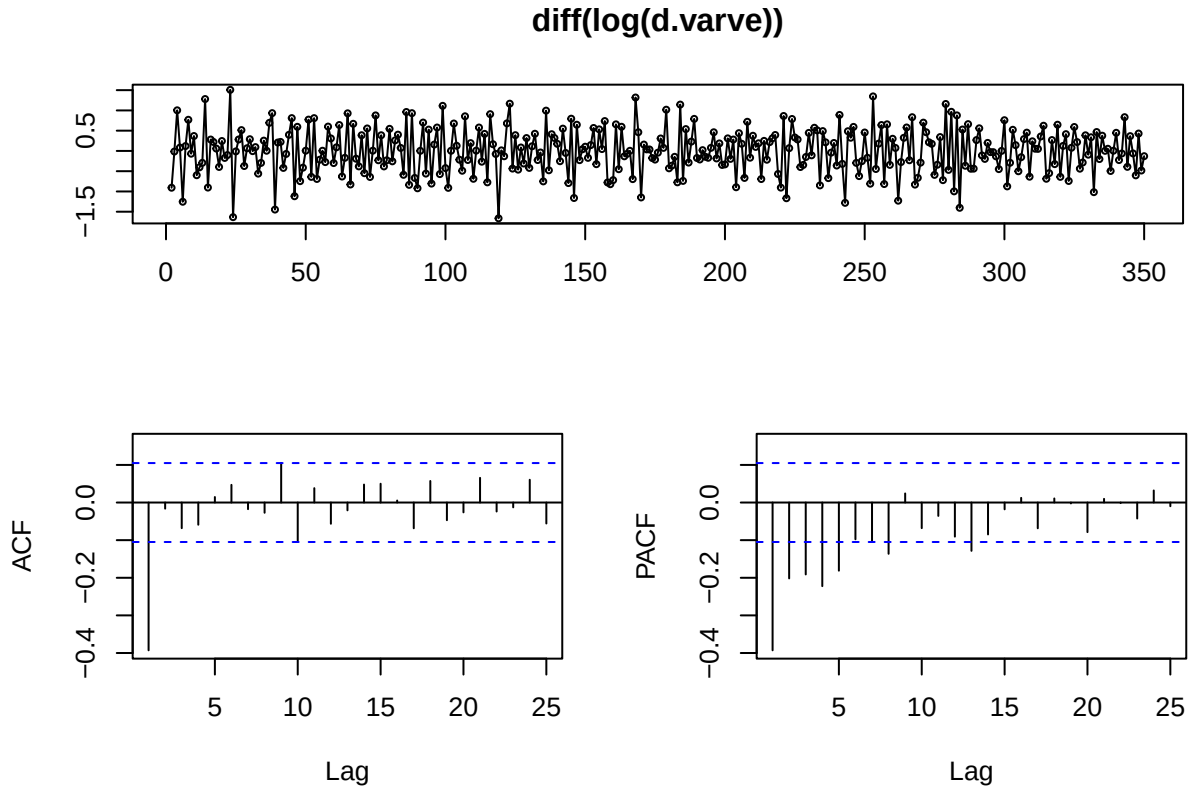



```
hist(d.varve)
```



Differencing

```
tsdisplay(diff(log(d.varve)))
```

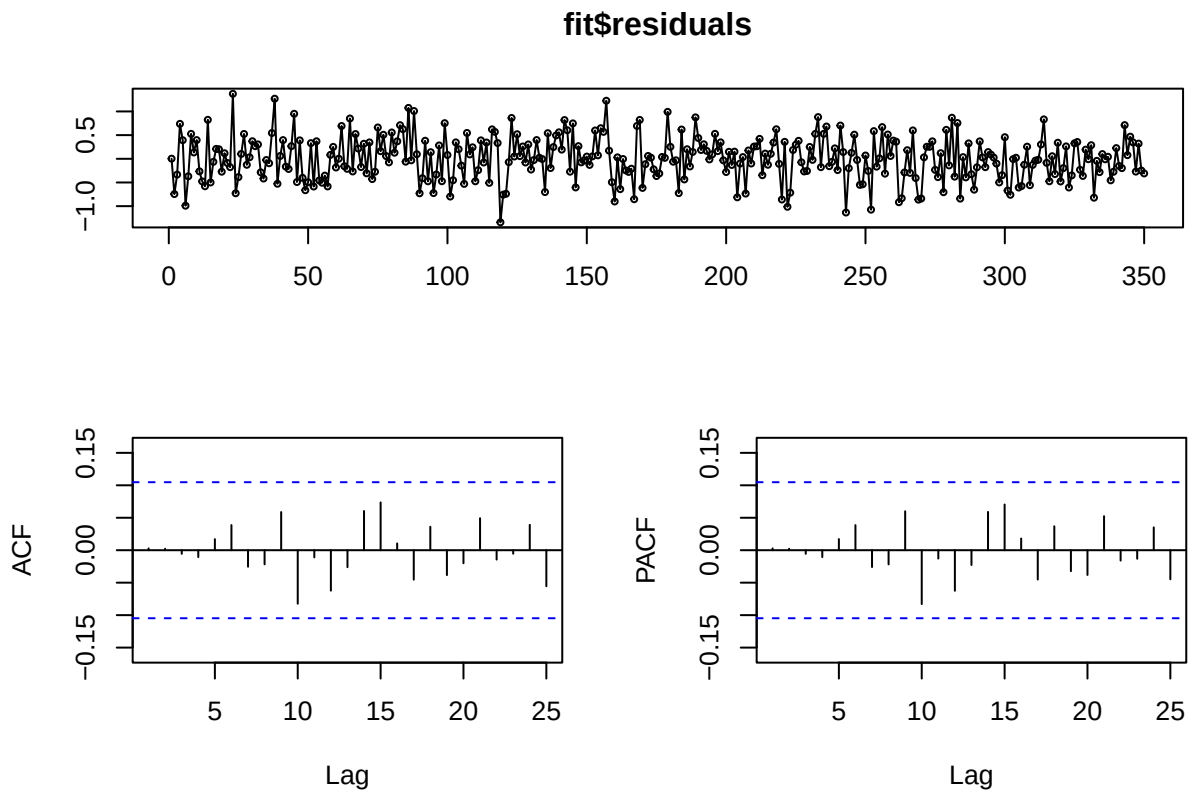


Observations: Cut-off of ACF at lag 1, Cut-off of PACF at lag 5 → Potentially suitable for ARMA(5,1) model → Suitable for fitting ARIMA(4,1,1) p=4, d=1, q=1

```
fit <- arima(log(d.varve), order = c(4, 1, 1))
fit
```

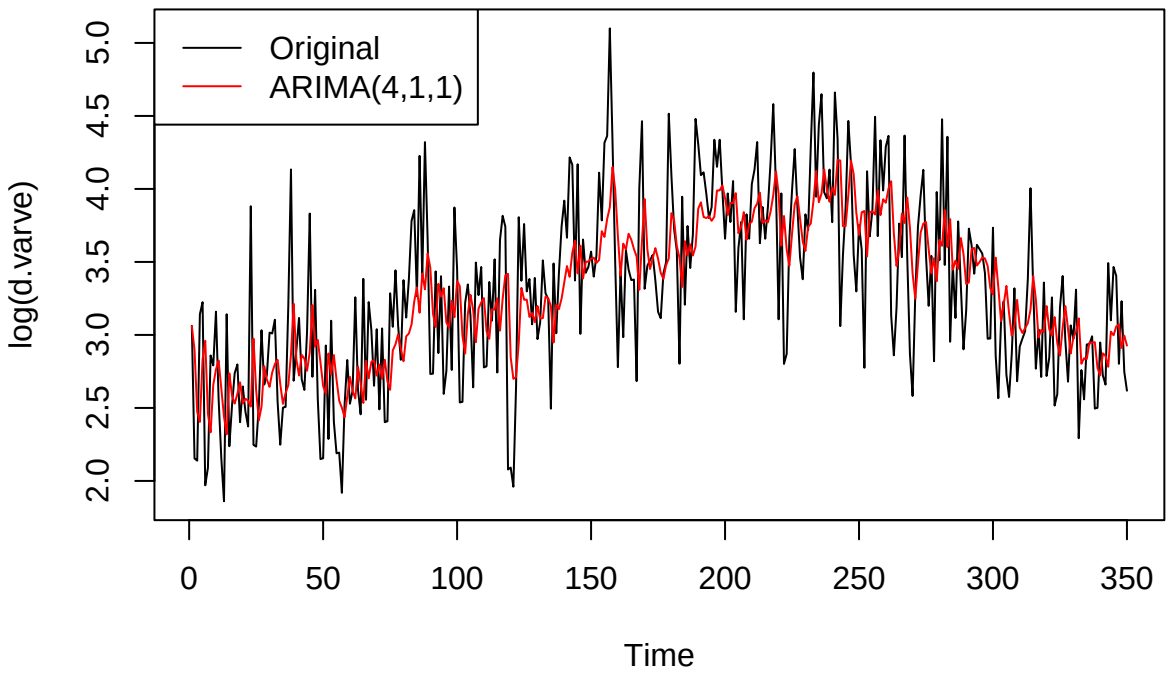
```
##
## Call:
## arima(x = log(d.varve), order = c(4, 1, 1))
##
## Coefficients:
##          ar1     ar2      ar3      ar4      ma1
##       0.2146  0.0393  -0.0691  -0.0608  -0.8922
## s.e.  0.0651  0.0592   0.0581   0.0597   0.0387
##
## sigma^2 estimated as 0.2118:  log likelihood = -225.1,  aic = 462.2
```

```
tsdisplay(fit$residuals)
```

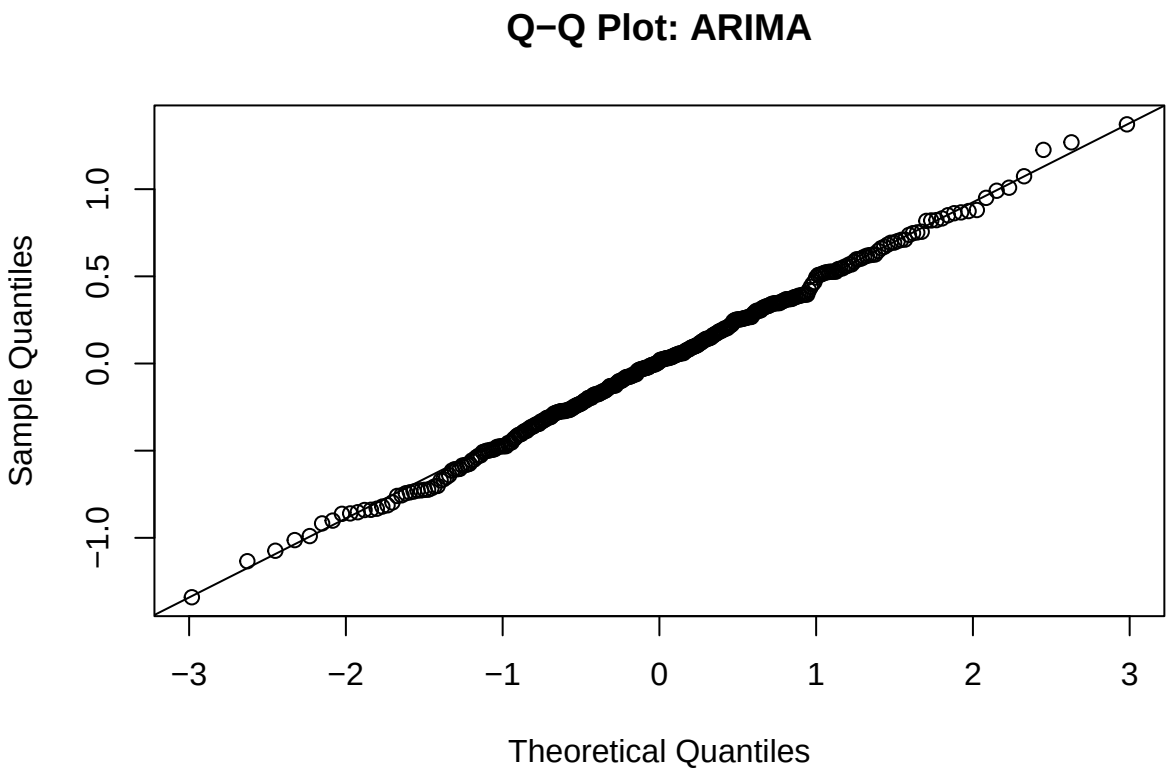



```
plot(log(d.varve))
lines(log(d.varve) - fit$resid, col="red")

legend(x = 'topleft', legend = c('Original', 'ARIMA(4,1,1)'),
       col = c('black', 'red'), lwd = 1, bg = 'white')
```




```
qqnorm(fit$resid, main='Q-Q Plot: ARIMA')
qqline(fit$resid)
```



b) Write down the model you chose in a) with its estimated coefficients.

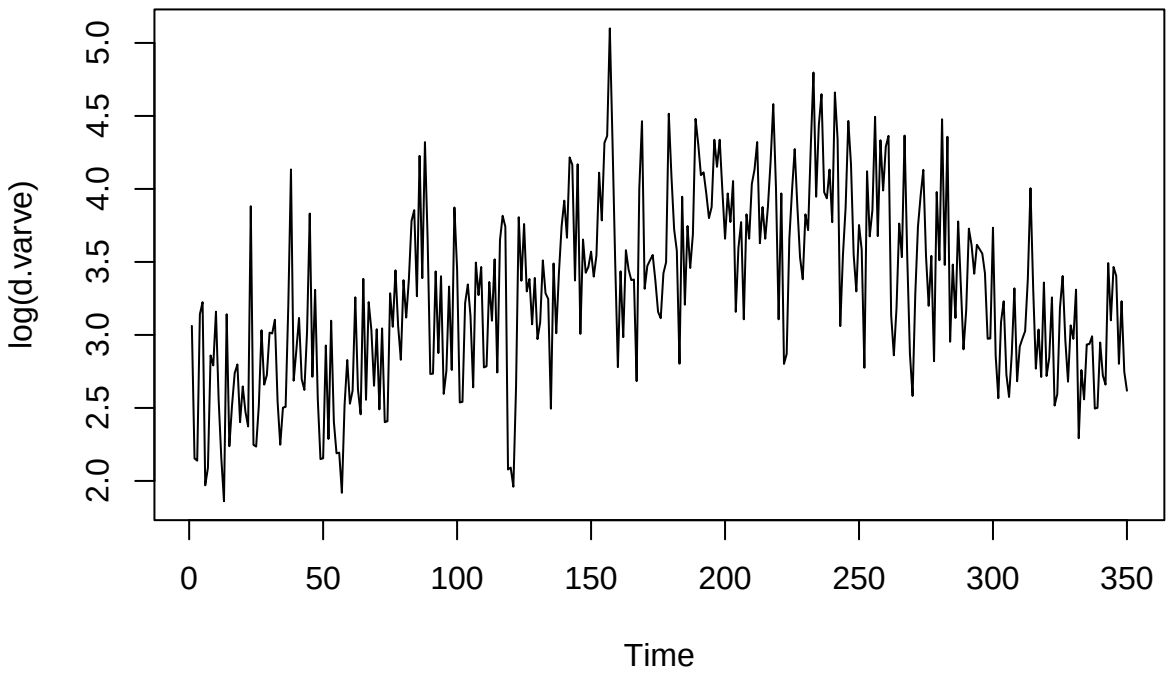
```
fit
```

```
##
## Call:
## arima(x = log(d.varve), order = c(4, 1, 1))
##
## Coefficients:
##          ar1     ar2      ar3      ar4      ma1
##       0.2146  0.0393  -0.0691  -0.0608  -0.8922
## s.e.  0.0651  0.0592   0.0581   0.0597   0.0387
##
## sigma^2 estimated as 0.2118:  log likelihood = -225.1,  aic = 462.2
```

$$\alpha_1 = 0.22, \quad \alpha_2 = 0.04, \quad \alpha_3 = -0.07, \quad \alpha_4 = -0.06, \beta_1 = -0.9$$

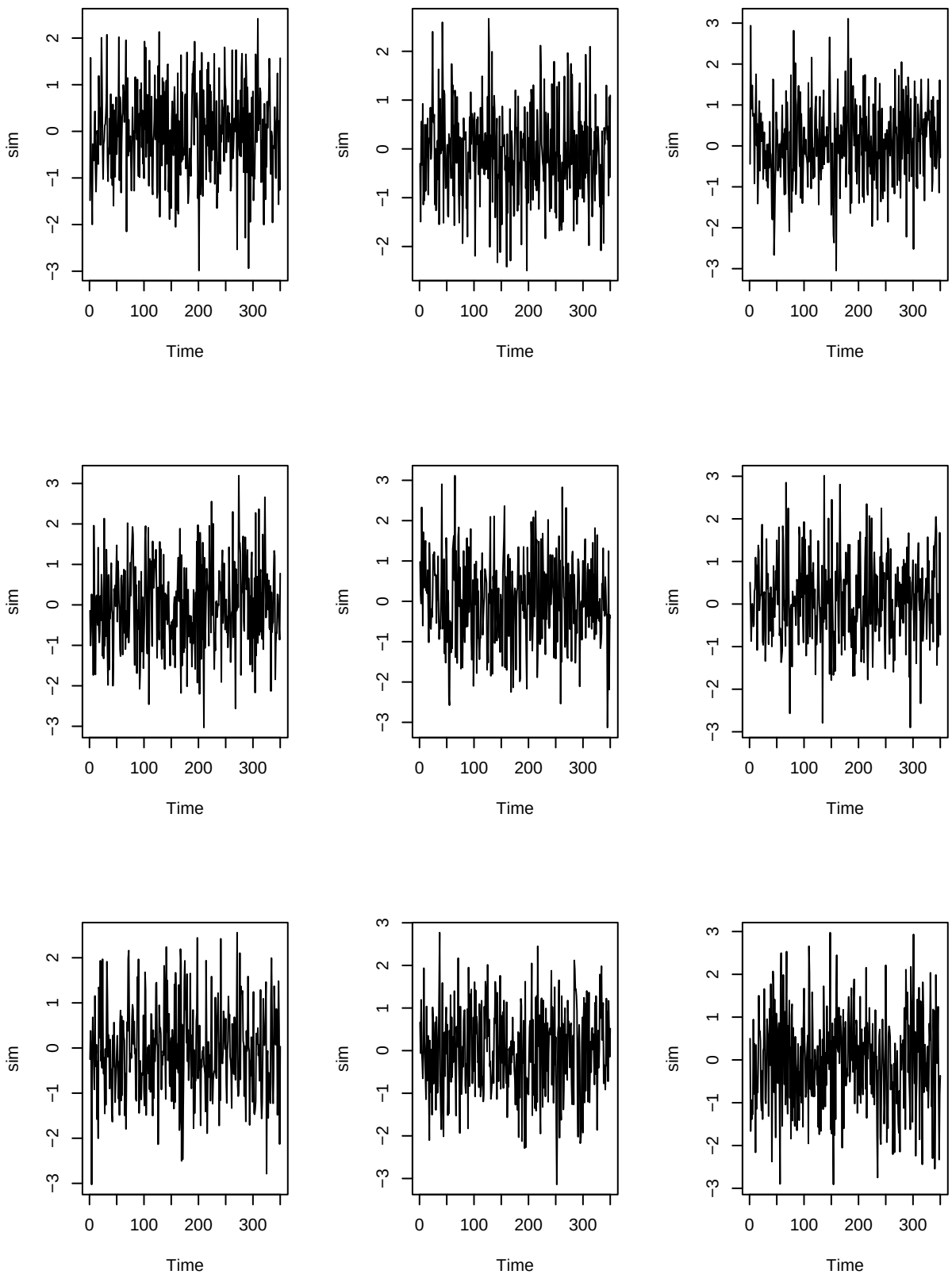
Generate time series with model and compare them to the original.

```
plot(log(d.varve))
```




```
set.seed(12)
par(mfrow=c(3,3))

replicate(9, plot(arima.sim(n = 350, model = fit$model), ylab = 'sim'))
```


sim
Time
0
100
200
300

The simulated time series look similar to the original data. However the range is not the same. The original lies between 2 and 5, whereas the simulated series have values between -3 and 3. Also, the slight curve in the original data is not represented in any of the simulations. I would assume this was due to an unusual effect, which is not happening in the simulations.

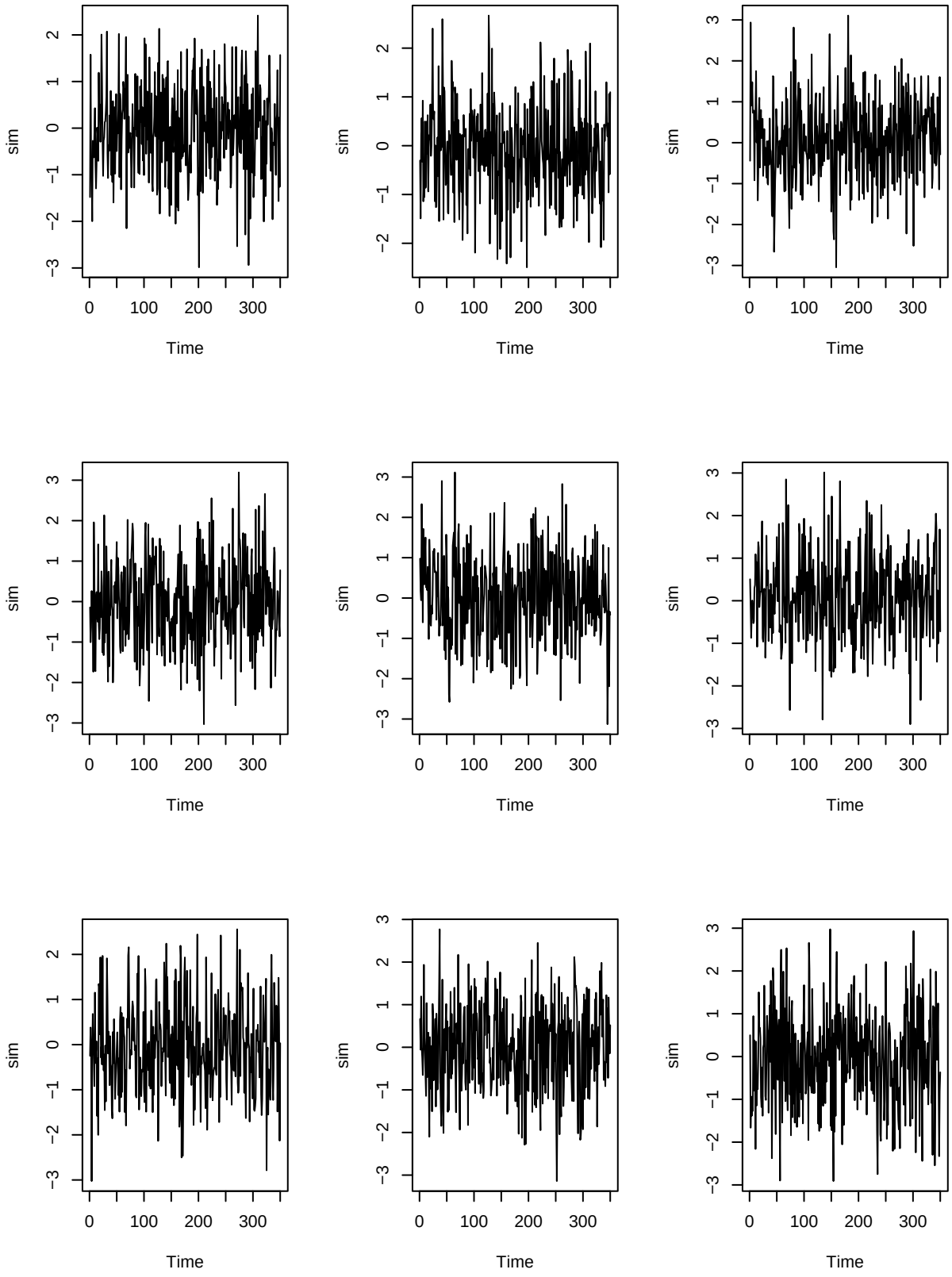
Just to make sure, let's use auto.arima (not implying that this gives us a perfect result).

```
set.seed(12)
par(mfrow=c(3,3))

fit.auto <- auto.arima(log(d.varve), max.p=10, max.q=10, seasonal=FALSE, ic="aic")
fit.auto
```

```
## Series: log(d.varve)
## ARIMA(4,1,3)
##
## Coefficients:
##           ar1      ar2     ar3      ar4      ma1     ma2      ma3
##       -0.5404  -0.7061  0.2386  -0.0205  -0.1251  0.2387  -0.8808
## s.e.   0.0648   0.0801  0.0744   0.0624   0.0363  0.0520   0.0429
##
## sigma^2 = 0.2123:  log likelihood = -222.26
## AIC=460.53   AICc=460.95   BIC=491.37
```

```
replicate(9, plot(arima.sim(n = 350, model = fit.auto$model) , ylab = 'sim'))
```


sim
Time
0
100
200
300
−3
−2
−1
0
1
2
3

Simulations are similar to our ARIMA(4,1,1) model. So we might be on the right track.

References

[1] *Arima.Sim Function - RDocumentation*. URL: https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/arima.sim (visited on 04/20/2023).

[2] Chris Haug. *Answer to "'ar' Part of Model in Not Stationary Error"*. Cross Validated. April 12, 2022. URL: https://stats.stackexchange.com/a/571299/385940 (visited on 04/17/2023).