

# Exercise 7

Philipp Drebes

13.04.2023

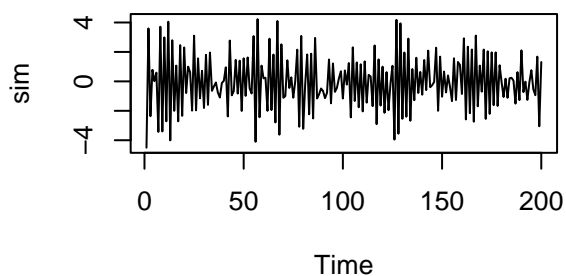
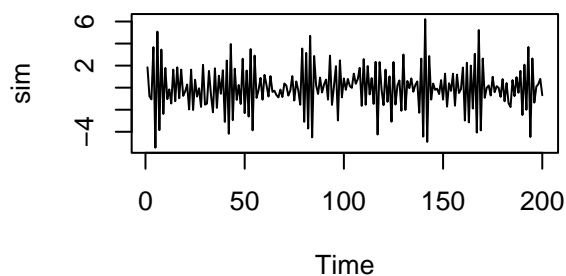
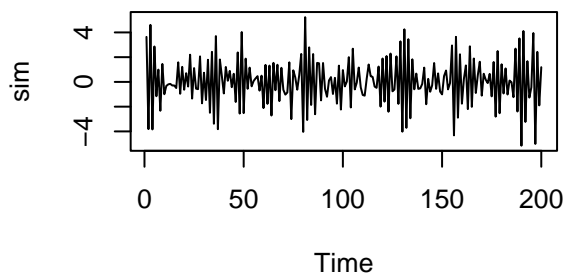
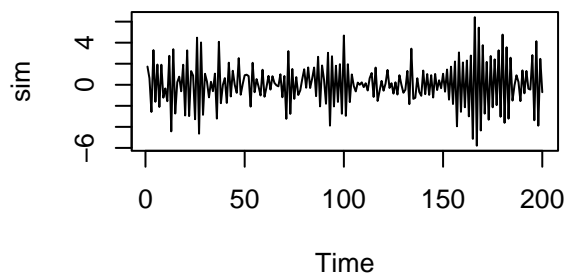
## Exercise 7.1

Similar to exercise 5.1, we start with some simulations. Thus, 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) ARMA(1,2) model with coefficients  $\alpha_1 = -0.75$ ,  $\beta_1 = -0.3$  and  $\beta_2 = 0.25$ .

```
set.seed(112233)
par(mfrow=c(2,2))

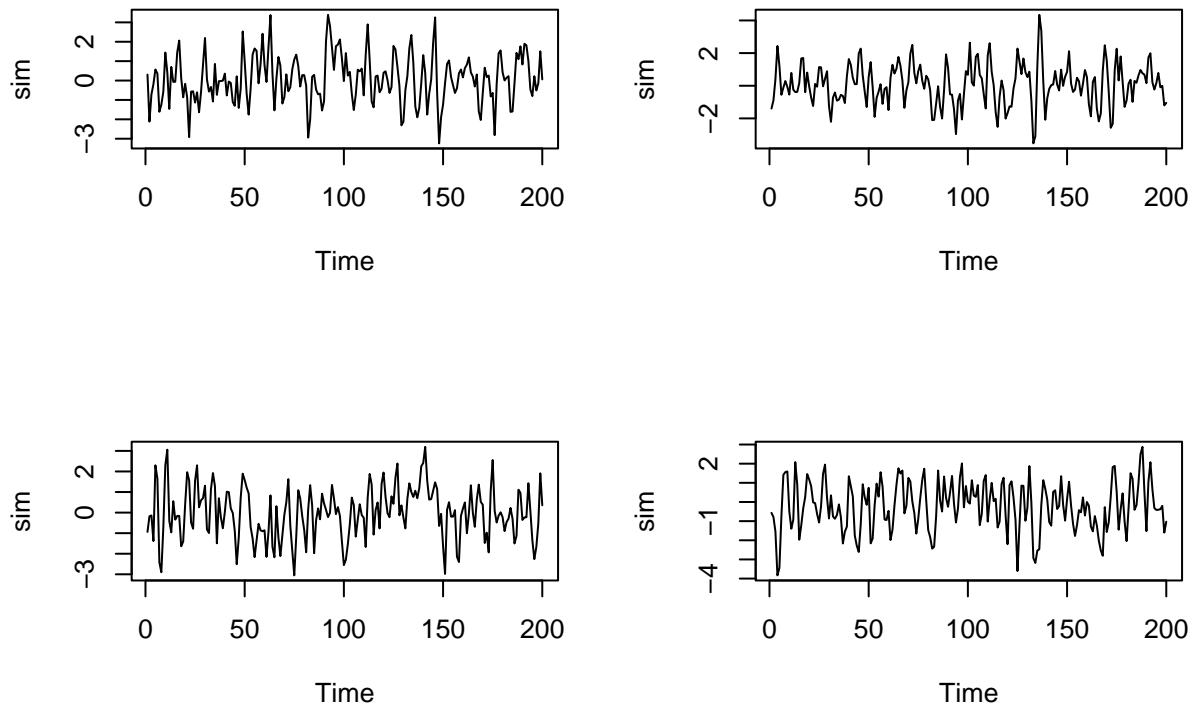
replicate(4, plot(arima.sim(n = 200, model = list(ar = c(-0.75), ma = c(-0.3, 0.25))),
                  ylab = 'sim'))
```



- b) ARMA(2,1) model with coefficients  $\alpha_1 = 0.5$ ,  $\alpha_2 = -0.3$  and  $\beta_1 = 0.25$ .

```
set.seed(112233)
par(mfrow=c(2,2))

replicate(4, plot(arima.sim(n = 200, model = list(ar = c(0.5, -0.3), ma = c(0.25))),
                  ylab = 'sim'))
```



## Exercise 7.2

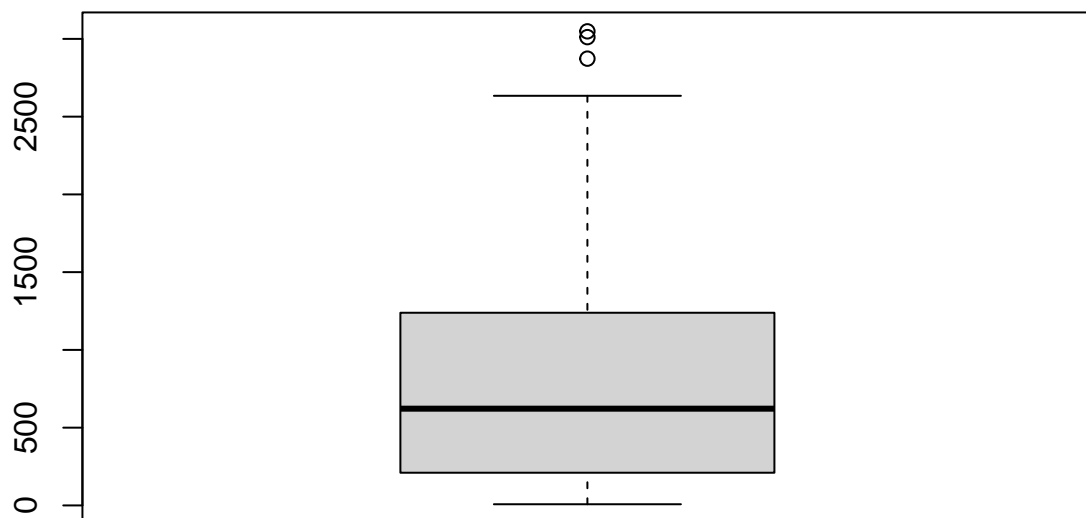
In this exercise, we look at the time series *sunspotarea*, which is available in the package *fpp*. It contains yearly data about the area of sunspots averaged over all days of the year (in units of millionths of a hemisphere). Sunspots are magnetic regions that appear as dark spots on the surface of the sun.

- a) Plot the time series. Why does it make sense to log-transform the time series?

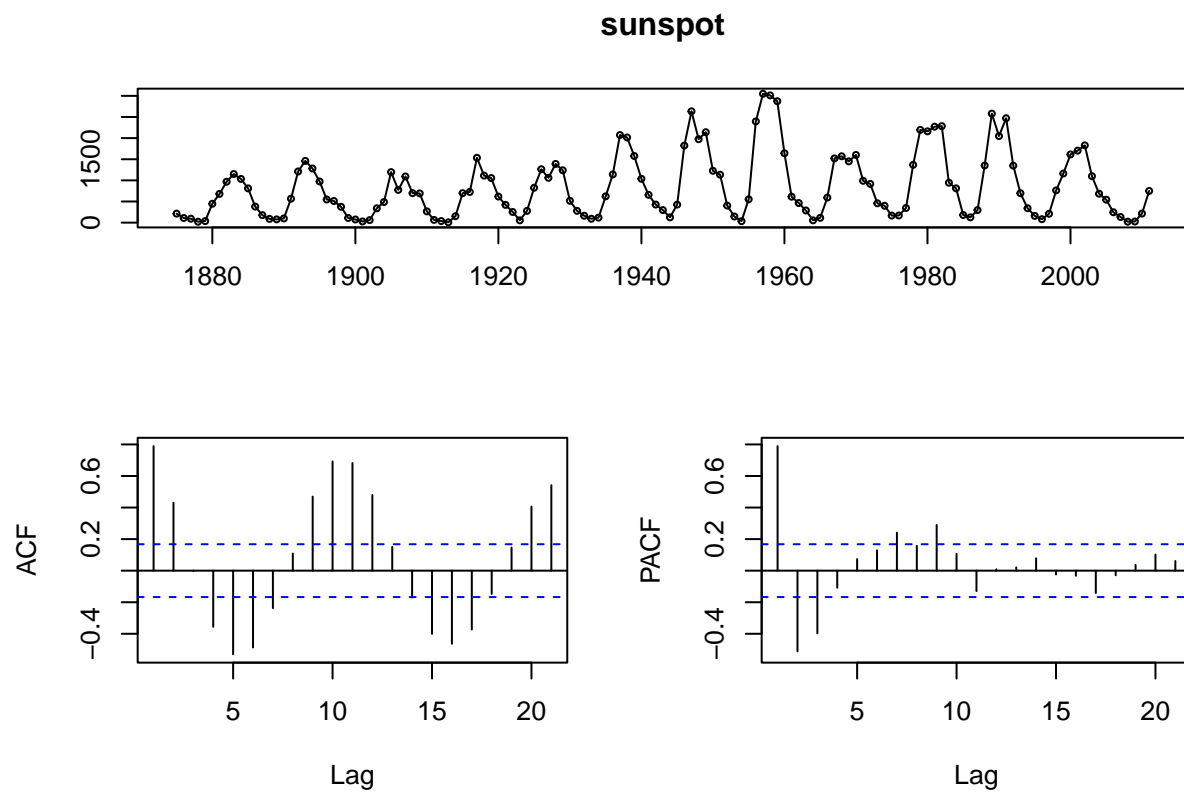
```
library(fpp)

sunspot <- sunspotarea

boxplot(sunspot)
```

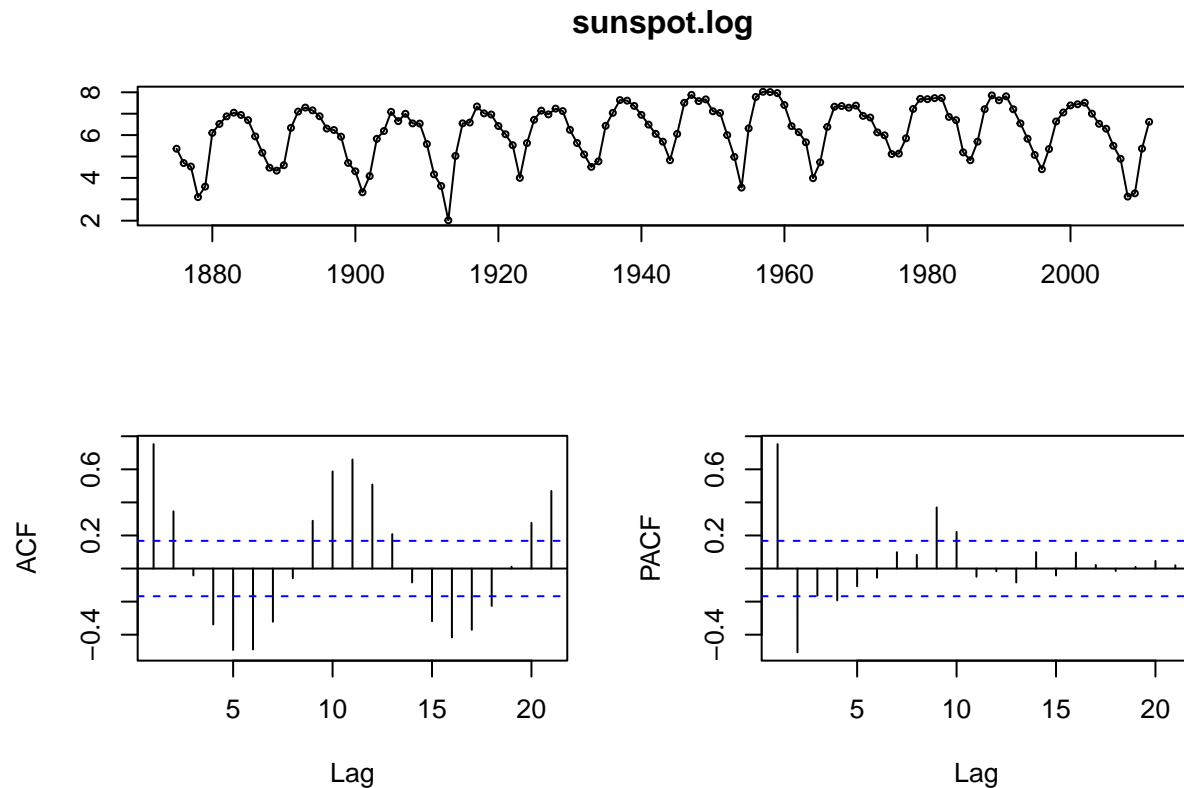


```
tsdisplay(sunspot)
```



Because it is right skewed, it makes sense to log-transform the data.

```
sunspot.log <- log(sunspot)
tsdisplay(sunspot.log)
```



b) Choose a suitable AR-model only based on the first 100 observations (1875 - 1974) of the log-transformed series.

First we fit an AR model with max order of 10, as this is the lag with the last significant value in the PACF.

```
sunspot.log.100 <- window(sunspot.log, start = 1875, end = 1974)
```

```
fit.ar <- ar(sunspot.log.100, method = "mle", order.max = 10)
```

```
fit.ar
```

```
##
```

```
## Call:
```

```
## ar(x = sunspot.log.100, order.max = 10, method = "mle")
```

```
##
```

```
## Coefficients:
```

```
##      1      2      3      4      5      6      7      8
```

```
## 0.7172 -0.1888 0.0046 -0.0621 0.0219 -0.1139 0.1695 -0.2285
```

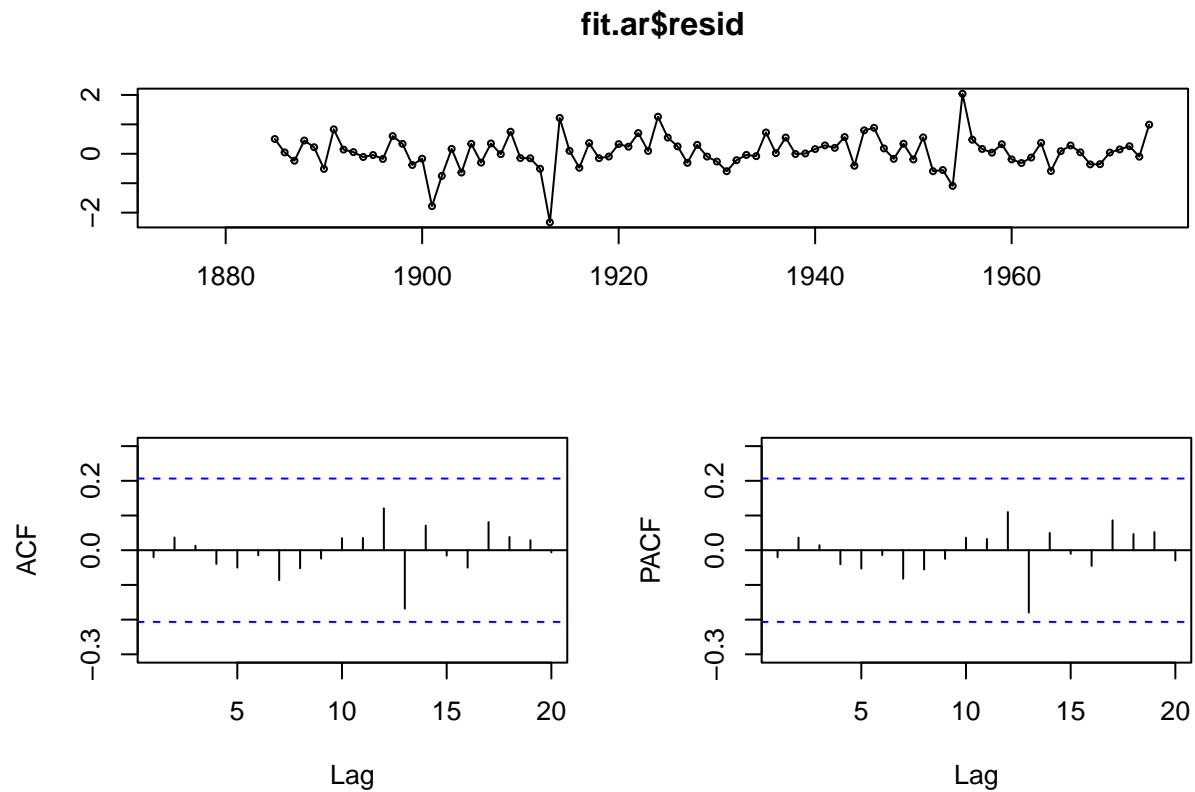
```
##      9      10
```

```
## 0.1272 0.3600
```

```
##
```

```
## Order selected 10  sigma^2 estimated as  0.3439
```

```
tsdisplay(fit.ar$resid)
```



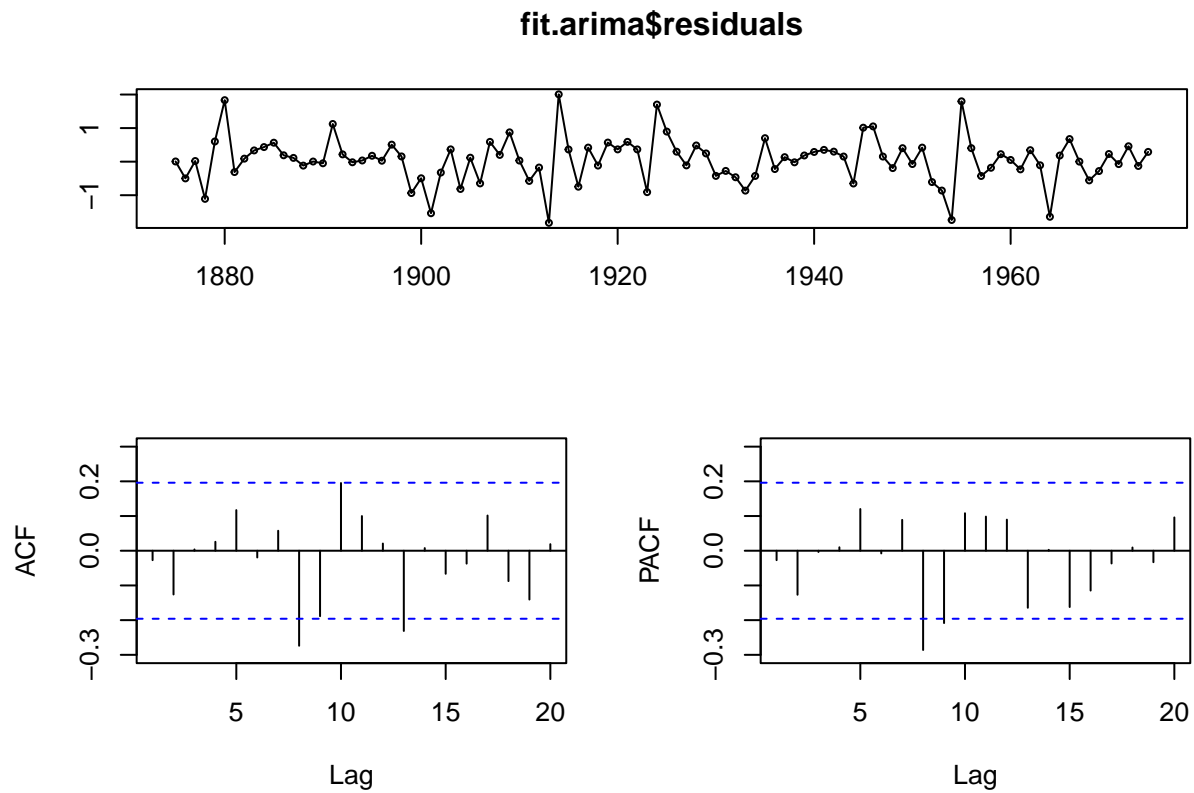
The residuals are stationary.

Let's also create an ARIMA model and compare the two.

```
fit.arima <- auto.arima(sunspot.log.100)
summary(fit.arima)
```

```
## Series: sunspot.log.100
## ARIMA(2,1,2)
##
## Coefficients:
##      ar1      ar2      ma1      ma2
##      1.5952 -0.8903 -1.7653  0.8370
## s.e.  0.0521  0.0470  0.0752  0.0858
##
## sigma^2 = 0.4758:  log likelihood = -103.45
## AIC=216.91  AICc=217.56  BIC=229.89
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 0.04943892 0.6723469 0.4846166 -0.9537964 9.53447 0.7137111
##              ACF1
## Training set -0.02719702
```

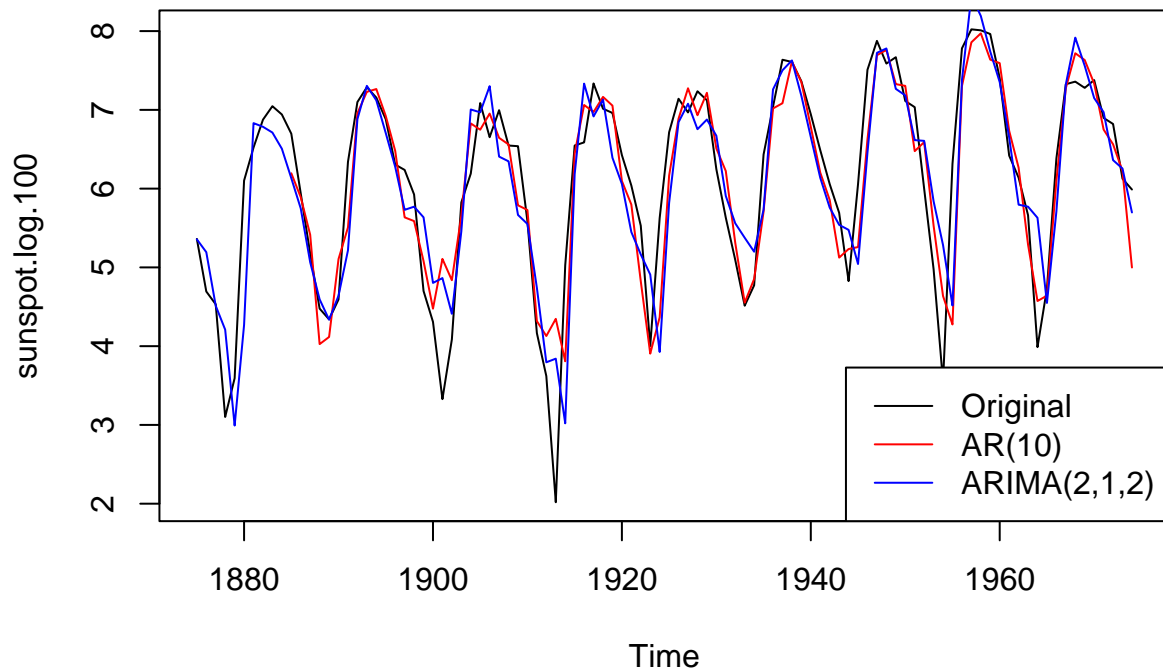
```
tsdisplay(fit.arma$residuals)
```



The autocorrelations of the residuals of the ARIMA model show a few significant values. The model might not be as good a fit as the previously created AR model.

```
plot(sunspot.log.100, main="Annual average sunspot area")
lines(sunspot.log.100-fit.ar$resid, col="red")
lines(sunspot.log.100-fit.arma$residuals, col="blue")
legend(x = 'bottomright', legend = c('Original', 'AR(10)', 'ARIMA(2,1,2)'),
      col = c('black', 'red', 'blue'), lwd = 1, bg = 'white')
```

## Annual average sunspot area



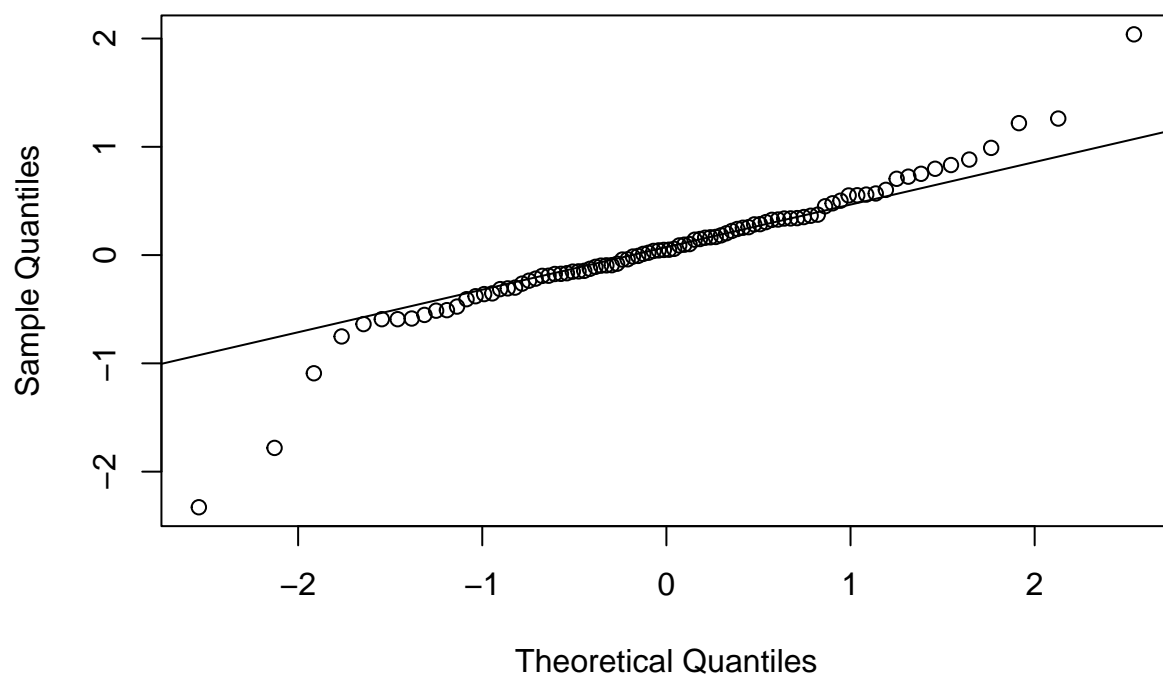
Compare the Q-Q plots:

```
par(mfrow=c(2,1))
qqnorm(fit.ar$resid, main='Q-Q Plot: AR(10)')
qqline(fit.ar$resid)

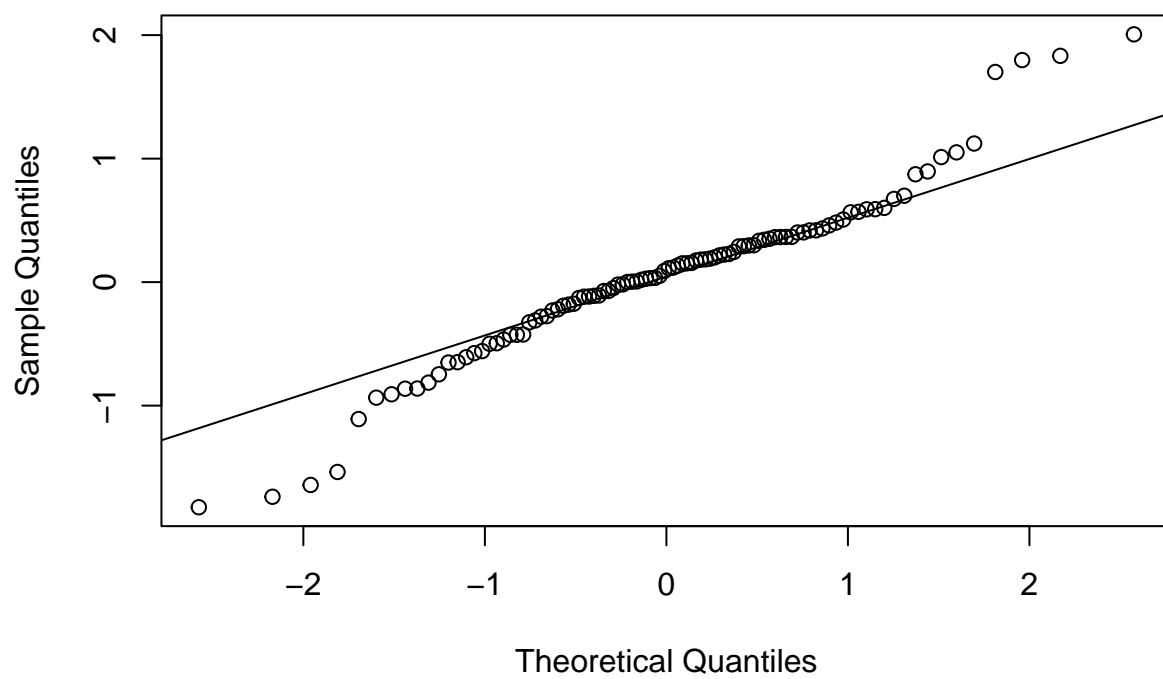
qqnorm(fit.arma$residuals, main='Q-Q Plot: ARIMA(2,1,2)')
qqline(fit.arma$residuals)
```



**Q-Q Plot: AR(10)**



**Q-Q Plot: ARIMA(2,1,2)**



The AR(10) model aligns better in the Q-Q plot and might therefore be the preferred model.

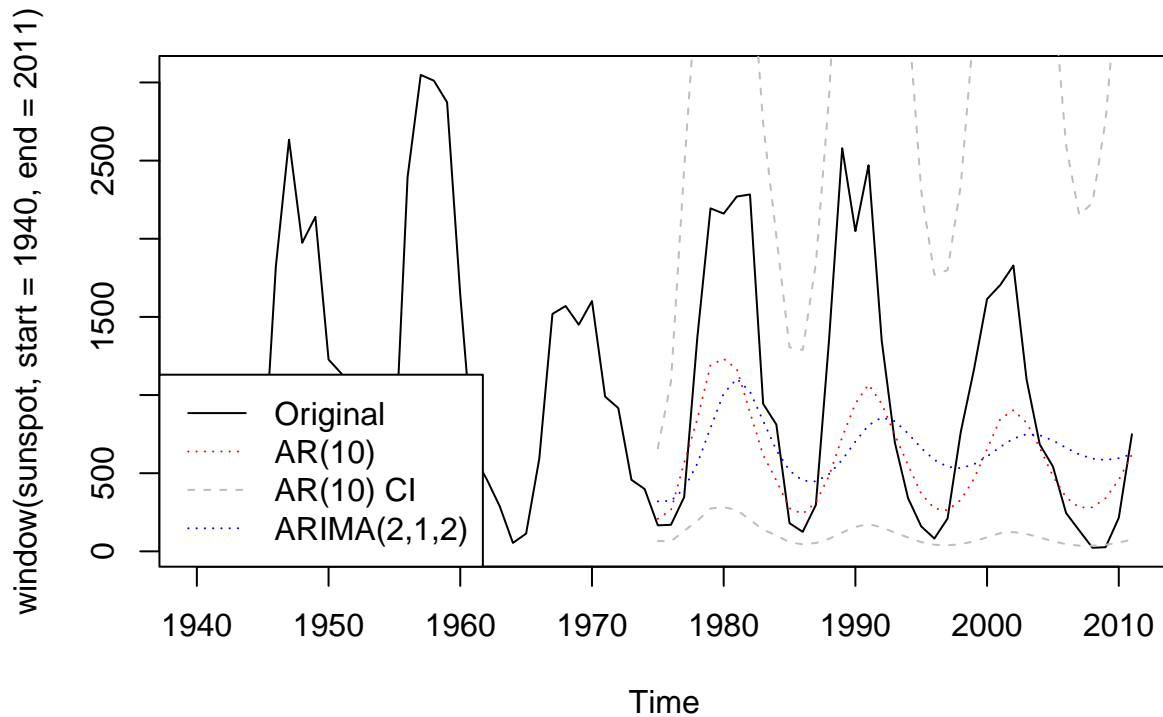
Let's predict the values for the years 1975 - 2011 by using both models and compare them to the actual recorded data.

```
fit.ar.pred <- predict(fit.ar, n.ahead = 37)
fit.arima.pred <- predict(fit.arima, n.ahead = 37)

plot(window(sunspot, start = 1940, end = 2011))
lines(exp(fit.ar.pred$pred), col = 'red', lty = 3)
lines(exp(fit.arima.pred$pred), col = 'blue', lty = 3)

lines(exp(fit.ar.pred$pred + 1.96 * fit.ar.pred$se), col = 'gray', lty = 2)
lines(exp(fit.ar.pred$pred - 1.96 * fit.ar.pred$se), col = 'gray', lty = 2)

legend(x = 'bottomleft',
      legend = c('Original', 'AR(10)', 'AR(10) CI', 'ARIMA(2,1,2)'),
      col = c('black', 'red', 'gray', 'blue'),
      lty = c(1, 3, 2, 3), lwd = 1, bg = 'white')
```



The AR(10) model has a superior fit compared to the ARIMA(2,1,2) model. Although, both models are not a very reliable fit when compared to the original data.

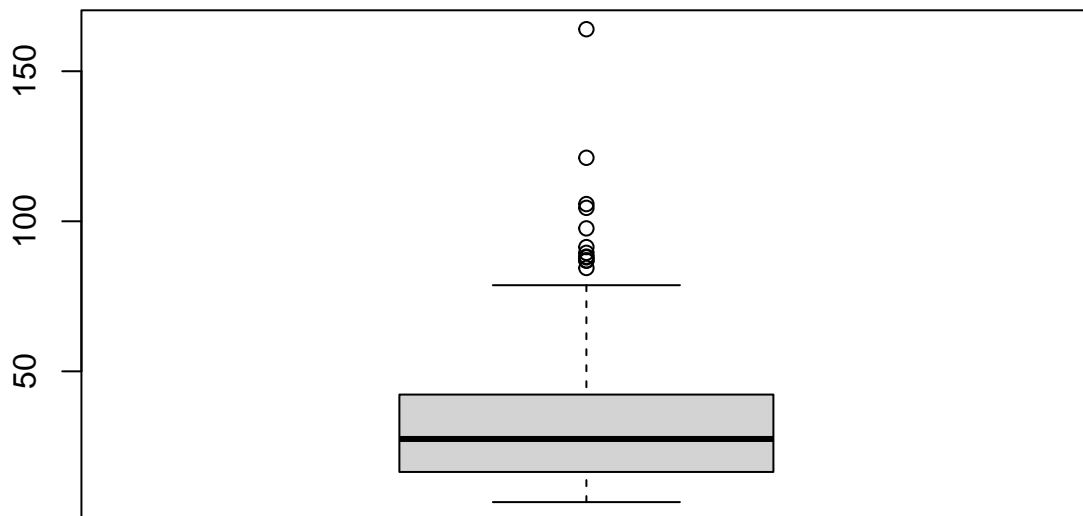
## Exercise 7.3

During their yearly spring melt, glaciers deposit layers of sand and mud. These annual sediments, known as varves, can be reconstructed in New England for the whole time between the beginning (about 12'600 years ago) till the end (6'000 years ago) of glacial retreat. From these varves, approximations of paleoclimatic parameters can be computed, such as temperature (a warmer year yields more sediment). In the dataset *varve.dat*, you will find 350 annual sediment diameters (contained in lines 201 through 550) starting at 11'660 years ago. After loading these data, first construct a time series object from them:

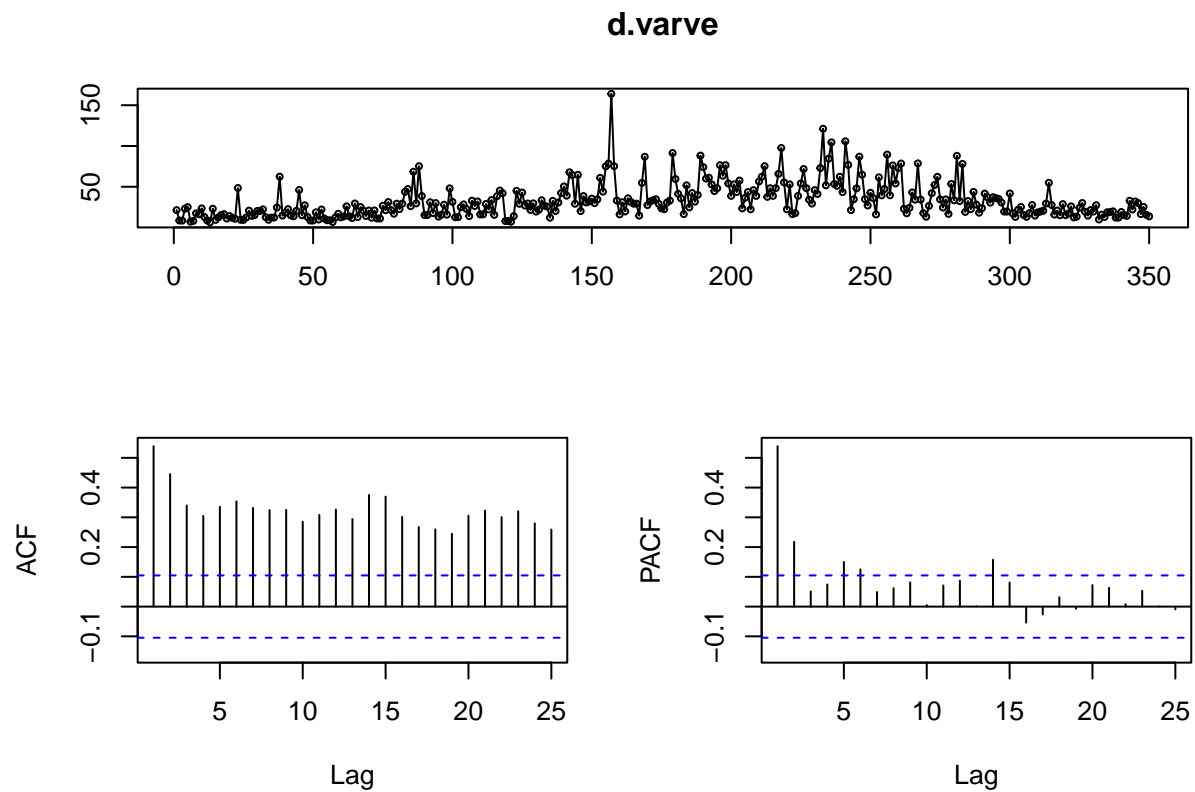
```
t.url <- "http://stat.ethz.ch/Teaching/Datasets/WBL/varve.dat"
d.varve <- ts(scan(t.url)[201:550], frequency=1)
```

a) It is advisable to log-transform the time series. Why?

```
boxplot(d.varve)
```



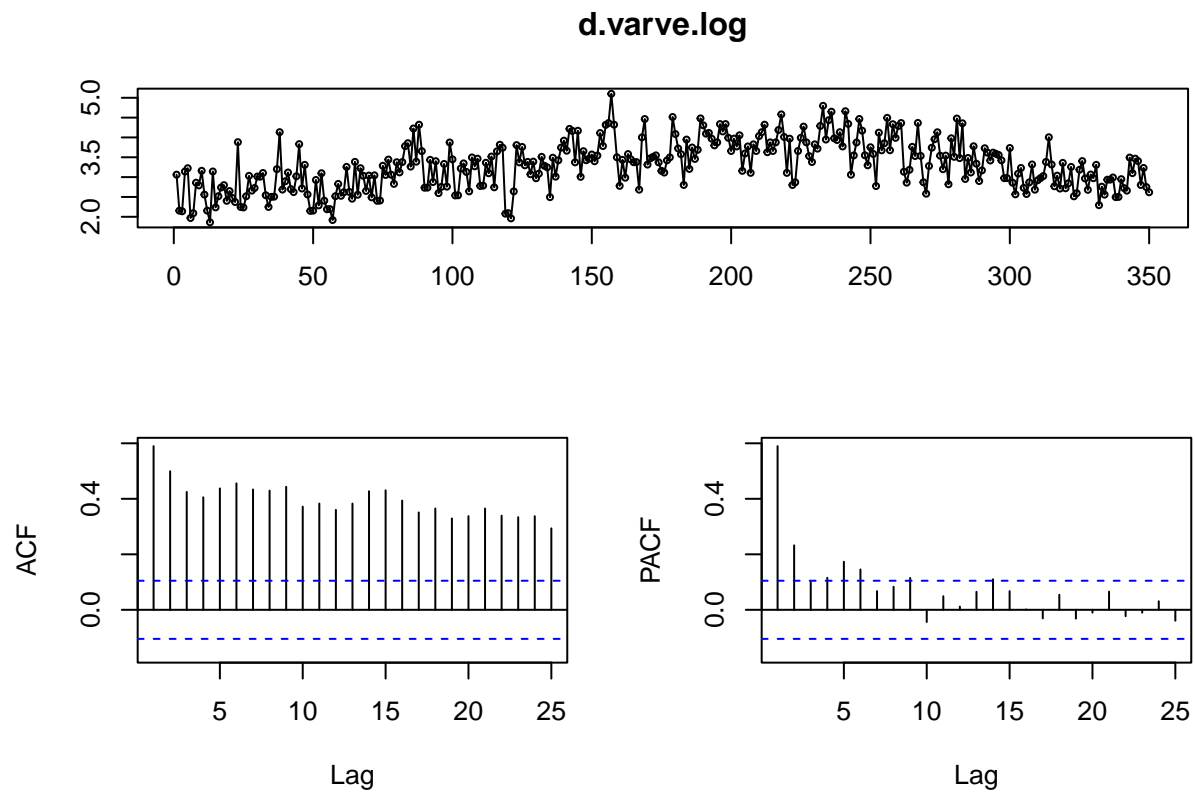
```
tsdisplay(d.varve)
```



The data appears to be right skewed which can be a reason to log-transform the data.

b) Is the log-transformed time series stationary? If not, how can you make this time series stationary?

```
d.varve.log <- log(d.varve)
tsdisplay(d.varve.log)
```



No, the log-transformed data is not stationary. We can clearly see an oscillatory behavior in the autocorrelation function, as well as a slow decay. These indicate that there is still seasonality and trend in the time series.

If the time series should be reduced to a stationary process, depending on the use case, we can apply differencing, filtration or decomposition.

Example for the STL decomposition:

```
plot(stl(ts(d.varve.log, frequency = 10), s.window = "periodic"))
```

