RTP Exercise Sheet

Series 3

Exercise 3.1

Simulations are key to validate models. Thus, we would like to use this exercise to simulate several time series by means of an ARMA model.

a) AR(2) model with coefficients $\alpha_1 = 0.9$ and $\alpha_2 = -0.5$.

b) MA(3) model with coefficients $\beta_1 = 0.8$, $\beta_2 = -0.5$ and $\beta_3 = -0.4$.

The innovation E_t shall follow a standard normal distribution $\mathcal{N}(0;1)$ in every model. For each of the three models, do the following:

i) First think of how the autocorrelations should behave theoretically.

ii) Use the procedure ARMAacf() to compute the theoretical autocorrelations of the models and plot them.

```
## Theoretical autocorrelations for X_t = 0.9 X_t-1 - 0.5
## X_t-2 + e_t
plot(0:30, ARMAacf(ar = c(0.9, -0.5), lag.max = 30), type = "h",
    ylab = "ACF")
## Theoretical partial autocorrelations
plot(1:30, ARMAacf(ar = c(0.9, -0.5), lag.max = 30, pacf = TRUE),
    type = "h", ylab = "PACF")
plot(0:30, ARMAacf(ma = c(..., ..., ...), ...), ...)
```

iii) Now simulate a realisation of length $n = 200$ for the models a) - c). Repeat each simulation several times to develop some intuition on what occurs by chance and what is structure.

Hint: You may use the procedure arima.sim() to simulate the time series. The length of the simulated series is choosen by setting the argument **n**. The model is set by the parameter **model** (to a list!).

```
r.sim1 <- arima.sim(n = ..., model = list(ar = c(0.9, -0.5)))
```

iv) Inspect the time series plot and the correlograms with the ordinary and partial autocorrelations.

Exercise 3.2

In this exercise we consider some examples of AR(p) models and check their stationarity.

a) Test the models

   i) $X_t = 0.5X_{t-1} + 2X_{t-2} + E_t$

   ii) $Y_t = Y_{t-1} + E_t$

   with the innovation E_t on stationarity with the help of the R function **polyroot**.

   Hint: During the lectures you saw that only roots with absolute value greater than 1 lead to stationary models. Confer the lecture notes at page 21.

b) For which value of the coefficient α_2 of X_{t-2} is the model $X_t = 0.5X_{t-1} + \alpha_2 X_{t-2} + E_t$ stationary?

c) Why is the model $Y_t = \alpha Y_{t-1} + E_t$ not stationary for $|\alpha| \geq 1$? Calculate the characteristic function and determine its roots to confirm this observation.

   Hint: Confer the hint at part a).

Exercise 3.3

We visit the analysis of the yield of a chemical process and will have a look at the *http://stat.ethz.ch/Teaching/Datasets/WBL/yields.dat* time series and its autocorrelations.

Read in the data with:

```
yields <- read.table("http://stat.ethz.ch/Teaching/Datasets/WBL/yields.dat",
    header = FALSE)
t.yields <- ts(yields[, 1])
```

a) Could these data be generated by an AR-process? If yes, what is the order p? R-hint: look at the `acf()` and `pacf()`

   Hint: How does a shift effect the expectation value of an AR(1) process?

b) Using the autocorrelations, compute the Yule-Walker estimate of α by hand. Recall the Yule-Walker equation for the estimated autocorrelation function at lag 1 reads:

$$\hat{\rho}(1) = \alpha \cdot \hat{\rho}(0)$$

   Furthermore, find the estimated mean $\hat{\mu}_X$ as well as the innovation variance $\hat{\sigma}^2$. Check your results using **R**.

R-hints:

```
r.yw <- ar(t.yields, method = "yw", order.max = ...)
r.yw$resid
str(r.yw)
```

For **order.max** use the order **p** you have detected in a).

c) Use the Burg method to compute the parameters of the AR model. Check its residuals.

```
r.burg <- ar(t.yields, method = "burg", order.max = ...)
r.burg$resid
str(...)
```

d) Use Maximum Likelihood to estimate these parameters.

R hint: There are two ways to achieve this:

```
r.mle <- ar(t.yields, method = "mle", order.max = ...)
r.mle$resid
str(...)
```

or

```
arima(t.yields, order = c(..., 0, 0), include.mean = TRUE)
```

The procedure arima() does have some advantages, including the following: if include.mean = TRUE is called (this is the default setting), a confidence interval for μ can be computed, since standard errors are in the output as well. Compute this confidence interval, with the given standard error or by looking at the component var.coef of the object constructed using arima(). Consult the R help for arima() if necessary.

Exercise 3.4

In this exercise we examine measurements of the vertical force acting on a cylinder in a water tank. A total of 320 measurements were taken at intervals of 0.15 seconds. Load the data from *http://stat.ethz.ch/Teaching/Datasets/WBL/kraft.dat* and convert them to a time series using

```
d.force <- read.table("http://stat.ethz.ch/Teaching/Datasets/WBL/kraft.dat",
    header = FALSE)
ts.force <- ts(d.force[, 1])
```

It is already known that at the time of the experiment, the water in the tank formed waves with (randomly changing) periods of around 2 seconds.

a) Create a subset of the data containing only the first 280 observations:

```
ts.forceA <- window(ts.force, end = 280)
```

Is a periodic behaviour to be expected in these data? If so, what should the period be? Does the plot of the times series agree with your expectations?

b) Suppose you would like to fit the time series `ts.forceA` by an AR(p) model. Which order p should this model have?

Choose a suitable order once by looking at the partial autocorrelations, and once by using the Akaike information criterion (AIC).

R hints: To calculate the AIC, fit an AR model with the R function ar():

```
ar.force <- ar(ts.forceA, method = ...)
```

For **method** use a method of your choice (`mle`, `burg` or `yw` are suitable options). AIC values for different orders p can then be found in `ar.force$aic`. (For this purpose, you don't need to specify the argument order.max in the ar()-function)

c) Fit an AR(p) model using maximum likelihood for the time series ts.forceA, where p is the order specified in Part b). Analyze the residuals. Is the model appropriate for this time series?

R hint: To fit an AR model using Maximum Likelihood with order **p**, you can use the R function arima():

```
ar.force <- arima(ts.forceA, order = ..., method = "ML")
```

d) Optional: Use the model fitted in Part c) to compute point predictions and prediction intervals for the next 40 measurements. Compare these graphically to the actual measurements.

R hints:

```
force.pred <- predict(ar.force, n.ahead = 40)
plot(window(ts.force, start = 250))
```

Then, plot the point predictions and the confidence intervals into the plot using lines(); consult the R help to find out how to get these estimates out of the object force.pred.

Disclaimer: Parts of the exercises are adopted from 'Applied Time Series Analysis' course at ETHZ by Marcel Dettling.