Series 4

1. In this exercise we shall 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 (dataset `kraft.dat`). Load these data 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 contained waves with (randomly changing) periods around 2 seconds.

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

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

Is 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 want to fit the time series `ts.forceA` by an AR model. Which order 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 = ...)
```

Use a method of your choice (`mle`, `burg` or `yw` are suitable options). AIC values for different orders p can now be found in `ar.force$aic`.

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 with *fixed* order p, you can use the R function `arima()`:

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

d) 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(d.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`.

2. Since simulations can be of use in model validation, we would like to use this exercise to simulate several time series by means of an ARMA model:

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

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

(iii) ARMA$(1, 2)$ model with coefficients $\alpha_1 = -0.75$, $\beta_1 = -1$ and $\beta_2 = 0.25$.

For all models, the error E_t follows the standard normal distribution $N(0, 1)$.

a) How should the autocorrelations behave based on the theory?

b) Use the function `ARMAacf()` to compute the *theoretical* autocorrelations and partial autocorrelations, and plot them up to lag 30.

R hints: Use the arguments `ar` and `ma` of `ARMAacf()` to specify the parameters of the models. With the argument `pacf`, you can specify whether partial or "normal" autocorrelations should be calculated.

c) Simulate all three models (i) to (iii). Take several different lengths for the time series: $n = 200$, $n = 500$ and $n = 1000$. Repeat these simulations several times to develop some intuition on what is "chance" and what is "structure". You don't have to print out all these plots, just have a look at them.

For each model, make a plot of one simulation for $n = 200$ and the corresponding correlograms. Compare the empirical autocorrelations to the theoretical ones from Part b).

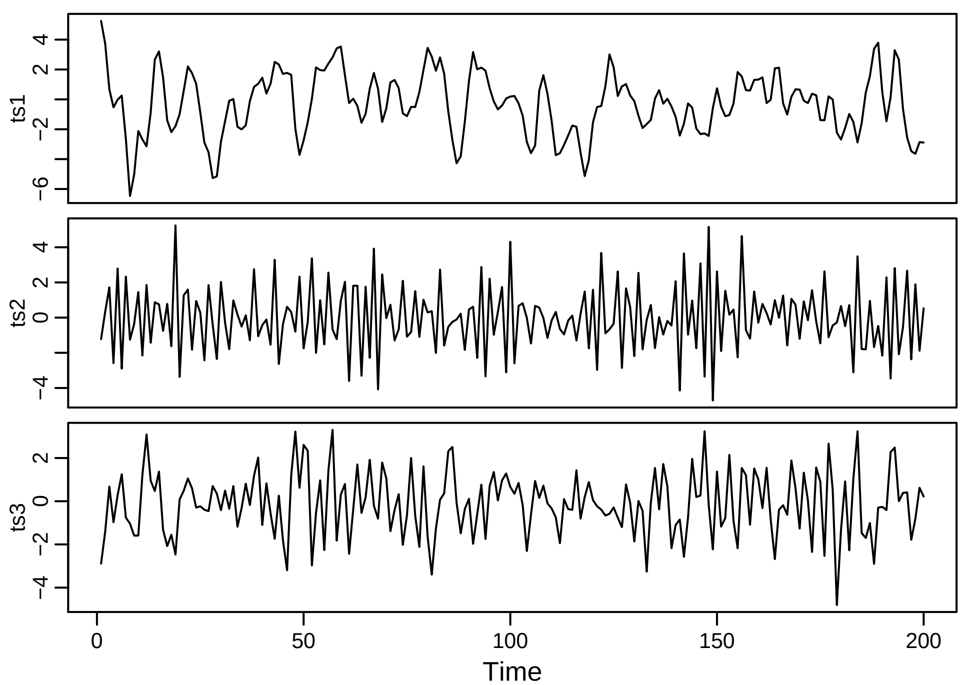
R hints:

You can use the procedure `arima.sim()` to simulate the time series. The length of the simulated series you can choose by setting the argument `n`, and the model by setting the parameter `model` (to a list!).

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

3. For each of the following three time series, find a suitable ARMA(p, q) model and estimate its parameters. To read the data, call:

```
> data <- read.table("http://stat.ethz.ch/Teaching/Datasets/ARMAsim.dat",
+   header = TRUE)
> ts1 <- ts(data[, "ts1"])
> ts2 <- ts(data[, "ts2"])
> ts3 <- ts(data[, "ts3"])
```



Preliminary discussion: Monday, April 11.

Deadline: Monday, April 18.