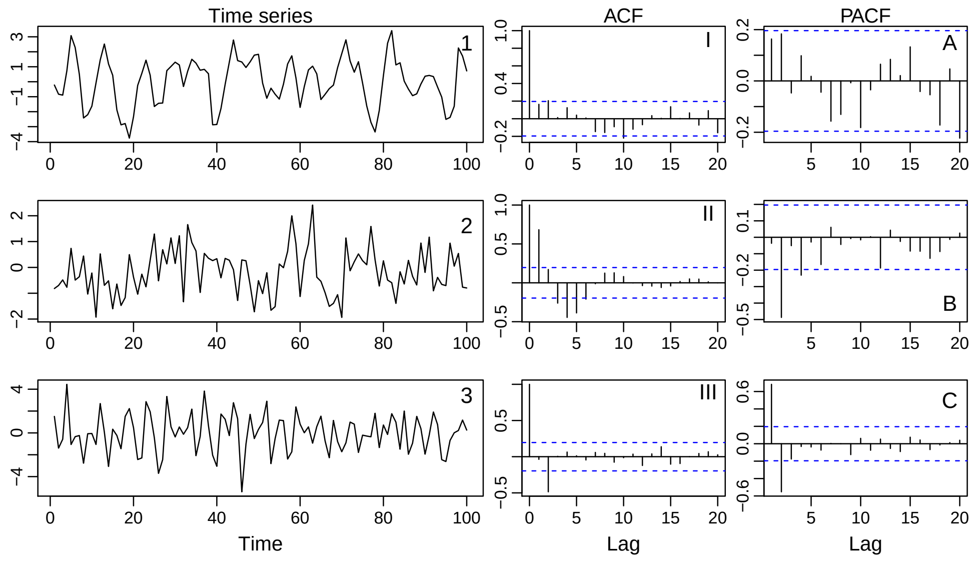
Series 3

1. In the figure below, three time series and their correlograms for usual and partial autocorrelations can be seen. Unfortunately, we have forgotten which correlogram belongs to which time series. Can you help?

Hint: First match up the usual autocorrelations with the time series they come from.



2. 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. Load the 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 = ...)
```

For `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(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`.

3. In this exercise we look at the yield of a chemical process. The relevant data from 70 successive experiments can be found in the dataset `yields.dat`. The aim of this exercise is to estimate the mean yield and construct a 95% confidence interval.

R hint: Load the dataset and create a time series as follows:

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

a) Make a time series plot, estimate the mean yield and mark this in the plot.
R hint: Use `mean()` to estimate the mean yield. You can then draw a horizontal line with intercept a using the command `abline(h = `a`)`.

b) Investigate the dependence structure of this time series. Look at its autocorrelations. Compare with lagged scatterplots, and characterise the dependence structure.
R hints:

```
> acf(...)
> lag.plot(t.yields, lag = ..., layout = c(..., ...), do.lines = FALSE)
```

c) Construct a 95% confidence interval for μ by estimating each of the autocorrelations that differ from 0.
How large would this confidence interval be if independence were falsely assumed?
R hint: You can compute $\widehat{\gamma}(0)$ with either of the following commands:

```
> var(t.yields) * (length(t.yields) - 1) / length(t.yields)
> acf(t.yields, type = "covariance", plot = F)$acf[1]
```

4. We revisit the analysis of the yield of a chemical process and will have another look at this time series and its autocorrelations.

R hint: 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 from an AR-process? If yes, what is the order p?
R-hint: look at the `acf` and `pacf`

b) Using the autocorrelations, compute the Yule-Walker estimate of α by hand. Furthermore, find the estimated mean $\widehat{\mu}_X$ as well as the innovation variance $\widehat{\sigma}^2$. Check your results using R.
R hint:

```
> r.yw <-  ar(t.yields, method="yw", order.max=...)
> 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 hint:

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

or

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

The procedure `arima()` does have some advantages, including the following: if `include.mean=T` 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.

e) Look at the formula for $\mathrm{Var}(\widehat{\mu})$ we have derived in the lecture and compute how this formula looks for an AR(1) process. Then compute an estimate of this variance. Use the coefficients estimated with the Burg method to do this (i.e. find out how you can estimate $\gamma(0)$ and all other coefficients of $\mathrm{Var}(\widehat{\mu})$ with the coefficients given by the Burg method). With your estimate of the variance calculate a 95% confidence interval for μ. Compare this to the other confidence intervals you computed in part d) of this exercise.

Preliminary discussion: Monday, March 20.