

## Series 3

1. In Exercise 1 of the last Series we have seen several methods for descriptive decomposition and elimination of trends. In this exercise we will use the approach of parametric modeling for the Mauna Loa atmospheric  $CO_2$  concentration data. This is a time series with monthly records from January 1959 to December 1997. The data is included in the default packages in R and you can load it with the command `data(co2)`.

a) Make a time series plot. Is this a stationary time series? If not, what kind of non-stationarity is evident? Into which components might this time series sensibly be decomposed?

b) Decompose the time series into the components specified in a) using a parametric model. Try out polynomials of different degrees and plot the time series, including fitted values

**R-Hint:** For fitting the parametric model we follow more or less the R-Code on page 43 in the script:

```
> tr <- as.numeric(...)
> tc <- tr-mean(tr) # centering to mitigate potential collinearity
> mm <- factor(rep(month.name, length(co2)/12), levels=month.name)
> fit <- lm(...)
> plot(co2)
> lines(tr, fitted(fit), col="red") # Plotting the fitted values
```

c) Try to compare the different polynomials by plotting them and comparing how well they fit the real data.

**R-Hint:** For plotting the different polynomials into the time series plot, follow again the R-code on page 43-44 in the script.

d) Compare the different fittings by plotting the residuals versus time (`tr`) and adding a LOESS smoother. Check if there is still a structure.

**R-Hint:** Follow again the R-Code on page 44 in the script.

2. The performance of a machine that analyses the creatine content of human muscular tissue is to be investigated using 157 known samples. These samples are given to the machine one at a time for it to determine their creatin content.

The data are from an investigation into the correct functioning of automated analysis machines. You can find them in the data

<http://stat.ethz.ch/Teaching/Datasets/WBL/kreatin.dat> .

In this exercise, we will focus on one of the variables in this data, namely `gehalt` (content).

a) Which stochastic model should this series of data follow if the machine is working correctly?

b) Use the time series plot, the autocorrelations and the partial autocorrelations to determine whether or not these data fit the ideal model found in Part a).

**R-Hints:**

Converting the data frame (`d.creatine`) to a time series:

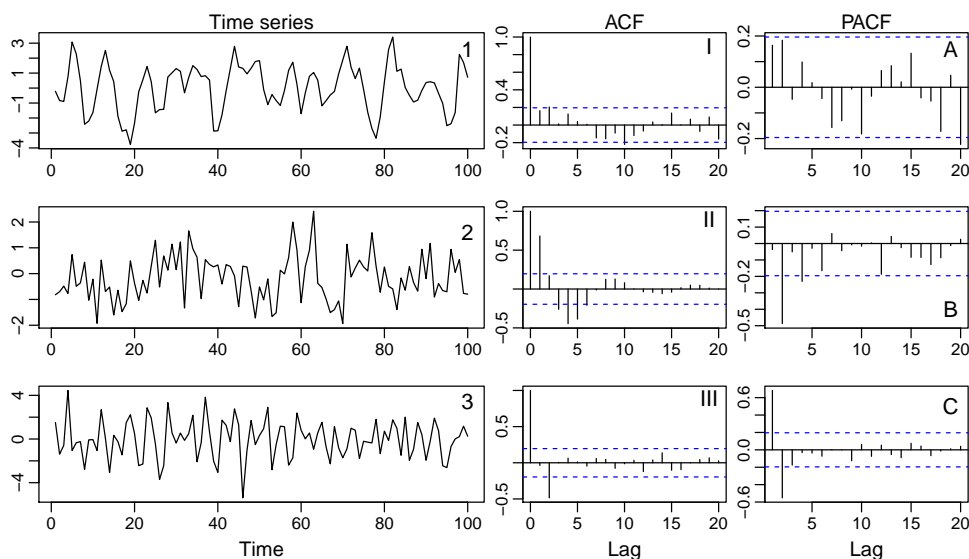
```
> t.creatine <- ts(d.creatine[, 2], start = 1, frequency = 1)
```

Plotting ACF and PACF:

```
> acf(t.creatine, plot = TRUE)
> acf(t.creatine, type = "partial", plot = TRUE)
> ## or
> pacf(t.creatine)
```

3. 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.



4. In this exercise, we consider the Australian electricity production, a time series indicating the monthly supply of electricity (in millions of kWh) in Australia over the period from January 1958 to December 1990. We use different methods to decompose the time series into trend, seasonal effect and remainder (cf. Series 2, Exercise 1) and compare the remainders by looking at their correlogram.

The data can be read in and stored as a time series with:

```
> dat <- read.table("http://staff.elena.aut.ac.nz/Paul-Cowpertwait/ts/cbe.dat",
                    sep=" ", header=T)
> tsd <- ts(dat, start=1958, freq=12)
> tselec <- tsd[, "elec"]
```

- Look at the plot of the time series. Why is the correlogram of this time series not meaningful? Explain in a few sentences.
- Decompose the time series into trend, seasonal component and remainder using the R function `decompose()`; plot the remainder and its correlogram. Interpret the plots in few sentences.

**R-Hints:**

```
> decomp <- decompose(tselec, type = "multiplicative")
> plot(...)
> acf(..., na.action = na.pass, plot = TRUE)
```

See Series 2, Exercise 1 for hints on extracting the remainder from the object `decomp`, or use the R-help: `?decompose`. The function uses a filter to estimate the trend; therefore, the first and the last few entries of the decomposition are not defined (value NA in R). For this reason we have to use the parameter `na.action = na.pass`, otherwise R complains about missing values.

- Decompose the log-transformed time series using the R function `stl()`. Estimate the seasonal effect once by averaging (parameter `s.window = "periodic"`) and once by choosing an appropriate smoothing window (parameter `s.window = ...`, where you have to choose an odd integer; cf. Series 2, Exercise 1c). To determine an appropriate smoothing window, you can look at the `monthplot`, cf. Series 2, Exercise 1d.

For both estimation approaches (averaging and smoothing window), plot the remainder and its correlogram, and comment on the plots.

**R-Hints:**

```
> elec.stl <- stl(log(tselec), s.window = ...)
> plot(...)
> acf(..., plot = TRUE)
```

- d) Explain why you used the parameter `type = "multiplicative"` in Task b), and why you log-transformed the time series before performing an `stl()` decomposition in Task c).
- e) Use the differencing approach. Choose `lag = 12` in order to get rid of the trend and period. Plot the new time series and acf. Compare to the previous methods.

**Exercise hour:** Monday, March 03.