Exercise 6

1. Consider once again the linear regression model from exercise 5:

$$Y_i = 1 - 2 \cdot x_{i2} + 3 \cdot x_{i3} + \epsilon_i, \quad i \in 1, \ldots, 100,$$

where the pairs x_{i2}, x_{i3} lie on a $\{1, \ldots, 10\} \times \{1, \ldots, 10\}$ grid. We assume the following error distribution:

$$\epsilon_i \sim 1 - \text{Exp}\,(1)\,.$$

The single-bootstrap confidence intervals from exercise 5 yield only approximately the correct coverage level of 95%. Using a double-bootstrap technique described in the manuscript this coverage level can be made much more precise. In this exercise you are going to write your own double-bootstrap routine and for each of the three regression-parameters in the above model we want compute a refined nominal coverage level $1 - \alpha'$ to get confidence-intervals with an approximate actual coverage level $1 - \alpha$ of about 0.95. To do this, complete the following steps:

a) Generate data from this model and store it in a data-frame.
R-Hint: For reproducibility use `set.seed(11)`.

b) Following the algorithm described in the manuscript, write your own double-bootstrap routine which estimates for every parameter and for a given *nominal* coverage level $1 - \alpha$, the corresponding *actual* coverage level $1 - \alpha'$ and evaluate your function on the following grid:

```
alpha = 1 - seq(0.999,0.8,length=20)
```

R-Hints: You may want to extend your single-bootstrap source-code from exercise 5. The evaluation of your double bootstrap-routine on the whole grid might take some time. If your computer-power allows, use `M=500` first-level and `B=999` second-level bootstraps.

c) Plot $1 - \alpha'$ against $1 - \alpha$ for every parameter and for $1 - \alpha' = 0.95$ deduce the corresponding $1 - \alpha$- value for every parameter by "graphical inversion" from the plots.

R-Hints: Graphical inversion can be done using the R-function `locator` which helps to find coordinates of arbitrary positions in a plot which are defined by "mouseclicking".

d) A more rigorous procedure to find the corresponding nominal levels $1 - \alpha$ for the three regression-parameters consists of smoothing the curve from **b)** and doing numerical root-finding.

R-Hints: The R-function `splinefun` performs spline-interpolation through given data points. Its output is the interpolating spline as a functional object. To find the desired coverage levels you could therefore search for the roots of the following function:

```
flev <- function(x)
            splinefun(1-alpha,cge[j,])(x) - 0.95
```

where `cge` is a 3×20-matrix containing your estimated actual coverage levels from the double-bootstrap-routine in **b)** and j defines which parameter is considered at the moment. Rootfinding can be done using the R-function `uniroot`.

2. The data-frame `parboot.dat` contains simulated data from the following model:

$$y = 8 \cdot x + 4 \cdot \cos(14 \cdot x) + \epsilon_i, \quad i \in 1, \ldots, 70,$$

where $x \in \{\frac{j}{70}, j = 1, \ldots, 70\}$ and $\epsilon_i \sim P$ iid. for an unknown distribution P.

In this exercise we want to compare confidence-intervals for nonparametric-regression which are generated by 3 different techniques, that are:

- hat-matrix approach (as in exercise 3)
- parametric bootstrap with assumption $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$
- model-based bootstrap with no assumptions about the errors.

To do this, fit a smoothing-spline (automatic choice of degrees of freedom) to the `parboot`-data and compute confidence-intervals at selected locations. Those locations are:

```
x.pre <- seq(5,62,by=3)/70
```

Plot the data, the spline-fit, the original curve and and all confidence intervals at the selected locations into the same plot and comment on the results.

R-Hints: The data is located at `http://stat.ethz.ch/Teaching/Datasets/parboot.dat`. Use $R = 2000$ bootstrap-samples in each case. For the hat-matrix approach you need to compute the hat-matrix for smooth.spline for the given data. This can again be done by smoothing unit vectors as in exercise 3. Use the same degrees of freedom for fit and hat-matrix-generation. `smooth.spline` automatically calculates the degrees of freedom. For the parametric bootstrap approach you need an estimate for the error variance σ^2. You can use the same estimate as in hat-matrix-theory, that is

$$\hat{\sigma}^2 = \sum_{i=1}^{n} \frac{(Y_i - \hat{m}(x_i))^2}{n - df}.$$

As a hint for the interpretation you could check the Gaussian assumption that the parametrical bootstrap-technique makes by looking at the normal-plot (`qqnorm`) for the residuals.

Deadline: Friday, May 27, 2005, at the beginning of the lecture.

Advice: Thursdays from 12.00-13.00, LEO C12.1, Leonhardstr. 27. Or contact either Maik Berchtold, `berchtld@stat.m ath.ethz.ch`, or Lukas Meier, `meier@stat.math.ethz.ch`.