Exercise 5

1. Consider the following linear regression model.

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

where the pairs x_{i2}, x_{i3} lie on a $\{1, \ldots, 10\} \times \{1, \ldots, 10\}$-grid, i.e.,

```
x2 <- rep(1:10,10)
x3 <- as.vector(sapply(1:10,rep,10))
```

a) Simulate 100 datasets[1] from model (1) and compute each time classical "normal theory" 0.95-confidence intervals and bootstrap 0.95-confidence intervals for the three regression parameters. How often do the confidence intervals include the true values under the following i.i.d. distributions of the $\epsilon_i,\ i = 1, \ldots, n$:

- $\mathcal{N}(0, 1)$.
- t_3 (`rt`).
- $\epsilon_i = e_i - 1,\ e_i$ exponential(1)-distributed (`rexp`).

R-hints: To make your results reproducible, use `set.seed(11)` at the beginning of your simulation experiment.
classical confidence intervals for output objects of `lm` must be computed manually:

```
lmc <- lm(y~x2+x3)
pars <- coef(lmc)                          # parameter estimators
se <- coef(summary(lmc))[,2]               # their standard errors
cubdy[i,] <- pars + se * qt(0.975,97)
clbdy[i,] <- pars - se * qt(0.975,97)
```

Remark: There exist R-functions `confint` and `vcov` which could be used to accomplish these tasks. Check their help-files!

The function `boot` from package `boot` allows automatic bootstrapping of statistics on given data. To apply this function, you have to write an own R-function to return the regression coefficients such that input and output are compatible with the demands of `boot` (see help page, parameter `statistic`). Such a function may look like this:

```
lmcoefs <- function(dat, ind)       # dat is a data frame containing variables
coef(lm(y~x2+x3,data=dat[ind,]))    # y, x2, x3.
                                    # ind is a vector of indices
                                    # generated by boot.
```

Bootstrap confidence intervals are computed by `boot.ci` which may look as follows

```
bstci <- boot.ci(bst,type="basic",index=k)
```

`bst` is the output of `boot`, `index` should be 1 for the intercept parameter, 2 and 3 for the regression parameters (if computed as in `lmcoefs` above). The interval bounds come as values `bstci$basic[4]` and `bstci$basic[5]`.

[1]It depends on the computer time you can spend, if you try 50, 100, 200 or 1000 simulations. It may need lots of time, because each time a complete bootstrap simulation has to be carried out. You can always downsize your simulations.

b) Now write your own bootstrap-routine and do the 100 simulations again. Compare all the three confidence-interval types (normal, bootstrap, own-bootstrap) and estimate the actual coverage for each of them for all three error distributions.

R-Hints: To sample the bootstrap-indices for your own bootstrap-routine, use the functions `sample` and/or `replicate` (Look at the help-files!).

c) In this part of the exercise we want to compare the usual L_1-loss $\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{m}(x_i)|$ with the L_1-generalization error $\mathbf{E}\left[|Y_{\text{new}} - \hat{m}(X_{\text{new}})|\right]$. This time the L_1-generalization-error is estimated by bootstrapping instead of cross-validation as described in the manuscript. Do 100 simulations for each of the given error distributions. In each simulation calculate the two quantities of interest and compare their averages over the whole range of simulations. A histogram of the two quantities may be informative too. You might want to recycle the bootstrap-samples you generated above.

Deadline: Friday, Mai 20, 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.math.ethz.ch`, or Lukas Meier, `meier@stat.math.ethz.ch`.