Computer Exercise 7 (R)

1. The purpose of this exercise is to familiarize you with bootstrapping (pp. 349, 385-387 in the R Book). You will use the data in **WaterChem.txt**. These are the data you analyzed in the first computer exercise (in problem 4 in that exercise). Your job now is to compute 95% confidence intervals for the mean, standard deviation and skewness for the variables SO4 and CL in the data file.

To do bootstrapping you need to load the package `boot` (used in an examples on pp. 385 – 387, 478 – 481 in the R Book). I have also provided some functions to help you perform the analysis. These are in the file `BootStat.R` (load the file with the command `source("BootStat.R")`). The functions in the file are called `boot.mean`, `boot.median`, `boot.var`, `boot.sd`, `boot.skew` and `boot.kurtosis` and are slight modifications of the mean, median, etc. functions that are suitable to use in bootstrapping. To do the bootstrapping of the mean of SO4, for instance, you give the command

```
bt1 <- boot(dat$SO4, boot.mean, R=10000)
```

R=10000 is the number of computer-generated replicate samples (you can use a different number if you want. To get some basic information from the bootstrapping, just write

```
bt1
```

which will tell you the observed mean, the estimated bias in this mean, and the bootstrap estimated standard error of the mean (compare this with the observed standard error). You can take a look at the distribution of the bootstrapped means with the command

```
hist(bt1$t) or plot(bt1)
```

To get a bootstrap 95% confidence interval, give the command

```
boot.ci(bt1, conf=0.95, type="bca")
```

Go on to compute the confidence intervals for the other statistics (standard deviation and skewness) and repeat the procedure for the CL variable. What do you conclude about the skewness of the distributions of SO4 and CL?

2. Now you will use Bayesian statistics, which can be particularly suited for situations requiring mixed effect models. The approach allows you to estimate parameters and variance components, together with confidence intervals (sometimes called “credible intervals” in Bayesian language). The R Book has a whole chapter on Bayesian statistics (Ch. 22, pp. 752 – 767). The MCMCglmm package calculates Bayesian statistics by **M**arkov **C**hain **M**onte **C**arlo simulations. Load this package (you may first need to install it).

```
library(MCMCglmm)
```

You will use this package more in the next computer exercise to analyze data in mixed models, but for now we will start by analyzing the data in **Timber.txt**, containing the diameter (in inches), height (in feet), and merchantable volume of wood, in cubic feet, of 31 trees (this is the data set you analyzed as problem 2 in Exercise 3). Before you used simple and multiple regression to find a good formula to predict the volume of wood. Your job now is to repeat some of that model fitting using Bayesian statistics. The purpose of the exercise is to show you how Bayesian statistics works in "standard" situations that you are already familiar with.

Begin by log-transforming the measurements (`dat$logVolume <- log(dat$Volume)`, and so on), since we found before that log transformation gave slightly better model fits.

You can visualize this data, for example using the ggplot2 package:

```
library(ggplot2)

ggplot(aes(x=logHeight, y=logVolume), data = dat) + geom_point() +
theme_classic()

ggplot(aes(x=logDiam, y=logVolume), data = dat) + geom_point() +
theme_classic()
```

Now you perform the Bayesian MCMC analysis by giving a command like

```
    fm1 <- MCMCglmm(logVolume ~ logDiam, data=dat, verbose=FALSE)
```

For this first model, you are doing a simple regression of log volume on log diameter. There are two fixed-effect parameters, the intercept and the slope of the regression line. The sample from the posterior distribution of these parameters is found in `fm1$Sol` (the posterior distribution of the variance components, in this case just the residual variance, is stored in `fm1$VCV`). You can have a look at the posterior distribution of the fixed-effect parameters by giving the command `plot(fm1$Sol)`. You get more information, including Bayesian estimates and confidence intervals of the parameters by giving the command `summary(fm1)`. How do these results compare with what you obtained before from the "standard" simple regression? Next, make the same comparison for the bivariate regression

```
    fm2 <- MCMCglmm(logVolume ~ logDiam + logHeight,

           data=dat, verbose=FALSE)
```

Finally, from a Bayesian perspective, is there a significant effect of log height in the regression? One way of approaching this question is to look at a confidence interval for the slope parameter corresponding to log height. You can use the command `HPDinterval(fm2$Sol, 0.99)` to get 99% confidence intervals for the fixed-effect parameters (or some other confidence level). Do you find that the estimated coefficient for log height is significantly different from zero?

3. Read the data in **Mice.txt**. These are the data from the experiment on survival of mice after treatment: Out of 111 mice, 57 were injected with bacteria plus an antiserum and 54 were injected with only bacteria, and for each mouse it was noted whether it survived the infection. The data file contains the variables Treatm and Surv, giving the treatment (AntiBact or BactOnly) and the survival (D or S, representing Died and Survived) for each mouse. Your job is to fit a general linear model to the data and to test the null hypothesis that the treatment did not influence survival. You do this with commands like

```
fm <- glm(Surv ~ Treatm, data=dat, family=binomial)

Anova(fm)
```

The family=binomial argument informs the glm function that Surv is assumed to follow a binomial distribution (the default is family = gaussian, which corresponds to the ordinary linear model). The family argument can also specify the link function; for the binomial the default is a logit link function. You could, just as well, have given the command

```
fm <- glm(Surv~Treatm, data=dat, family=binomial(link="logit"))
```

which gives the same result as the one you got. Now, try to assess model fit using the DHARMa package (don't forget to install it first using install.packages("DHARMa")). Try this line of code:

```
simulationOutput <- simulateResiduals(fittedModel = fm)

plot(simulationOutput)
```

The function simulates data based on the assumed distribution of your model, and compares the distribution of residuals in the simulation to the actual observed ones. For example, the black line in the boxplot is the median of the residuals in your data, and should fall around the 0.5 mark, and the lower and upper part of the "box" is the first and third quartile (and should fall roughly on the 0.25 and 0.75 mark, respectively. Do you think the model fits?

What do you conclude about the effect of the treatments on mouse survival? To read more about glm and binomial functions see pp. 557 – 564 in the R Book.

Read the data in **Mice2.txt** and name it *dat2* for instance. These are the same as in Mice.txt but arranged differently.

```
fm2 <- glm(cbind(Nsurv, Ndead) ~ Treatm, data=dat2,

family=binomial)

Anova(fm2)
```

Also compare

```
summary(fm)
summary(fm2)
```

Have a look at the residual deviance and degrees of freedom in the summary table(s). What does this mean? Look at pp. 631-2 in the R Book and make appropriate changes to your model.

4. Read the data in **Wildebeest2.txt**. These data represent a three-way classification of wildebeest carcasses. The qualitative variables used for classification were SEX (either FEMALE or MALE), MARROW (state of bone marrow: SWF, OG, or TG), and cause of death: either PRED or NPRED. The bone marrow variable gives an estimate of the nutritional status of the animal (SWF is well fed, OF is intermediate, and TG is in bad condition). Your job is to analyze how these variables are related to each other. Fit the generalized linear model as follows

```
fm <- glm(cbind(PRED,NPRED) ~ SEX + MARROW + SEX:MARROW,
data=dat, family=binomial)
```

The response is a two column matrix, with the first column giving the number of deaths by predation and the second column the number of other deaths, for the particular combination of SEX and MARROW. For statistical testing, it may be convenient to load the `car` package and give the command

```
Anova(fm)
```

How do you interpret the output from this test and what do you conclude? It might be helpful to look at a plot. Try loading the `effects` package (command `library(effects)`) and giving the command

```
plot(effect("SEX:MARROW",fm))
```

5. Read the data in **Lizards.txt**. These data give the presence or absence of lizards of the genus *Uta* on a number of islands in the Gulf of California. Your job is to investigate if the presence of lizards is related to the length of the island perimeter divided by the island area (which could be taken as a measure of the input of detritus to the island). The variables are: ISLAND: island name, PARATIO: Perimeter:area ratio, UTA: lizards absent/present, PA: present/absent in numerical 0/1. Logistic regression might be a good idea (see pp. 628 – 635, 650 - 655). Try either

R console:

```
fm <- glm(PA ~ PARATIO, data=dat, family=binomial)
```

or

```
fm <- glm(UTA ~ PARATIO, data=dat, family=binomial)
```

test the regression with

```
Anova(fm)
```

And visualize, for example with ggplot2:

```
ggplot(aes(x=PARATIO, y=PA), data = dat) +

  geom_smooth(method="glm",col='black', se=T, method.args =
  list(family="binomial")) +

  geom_jitter(width=0.00,height=0.03,shape=1) +

  scale_y_continuous("Lizard presence (0/1)") +

  scale_x_continuous("PARATIO") +

  theme_classic()
```

a) What do you conclude?

b) You have gone on a holiday trip to this island group. Your particular island has a parameter/area-ratio of 6.4. As you like lizards you are thinking about going out to look for them, so you would like to know the chance of having them on your island. Use the parameter estimates from your model to calculate the probability. Hint: use `summary(fm)` to get the estimates. Look at the notes from Lecture 13 if you need help.

You can try graphically checking your prediction by plotting a fitted logistic curve using your model estimates by giving the following commands:

```
plot(PA ~ PARATIO, data=liz)

x <- seq(0, 63, 0.01)

lines(x, predict(fm, list(PARATIO=x), type="response"), lwd=2)
```

Write `?seq` to find out exactly what the sequence function does (here we create a vector from 0 to 63 with 0.01 steps).