Example 15

MULTIPLE LINEAR REGRESSION

The question addressed here is about mothers and their babies and if there is an influence of different parameters (like gestation time, family income, mother's age) (=explanatory variables) on birth weight (=outcome) based on a multiple linear regression model. The data can be found as standard data set (data frame) *babies* in the library *UsingR (*missing values are indicated for different variables as 98, 99, 999 …).

Define a multiple regression model. Show pair-wise scatter plots of the variables, show if there is an influence of several explanatory variables to the outcome (birth weight) by a global F-test, give the goodness-of-fit measures (R^2 and adjusted R^2). Interpret regression coefficients. How can you decide if a variable should be included in the model by a partial F-test or the Akaike information criteria? Are the model assumptions are complied? Calculate regression coefficients also with the hat matrix.

```
# Download and install the library "UsingR"

> library()        # shows which libraries are installed

# If "UsingR" is not installed download http://genome.tugraz.at/biostatistics/UsingR_0.1-1.zip to your
# local directory and install (>Pakete>Installiere Paket(e) aus lokalen Zip-Dateien...)

> library(UsingR)  # include library/packages
> attach(babies)   # include in path so you can use "gestation" instead of "babies$gestation"
> babies           # there are some missing values indicated for different variables as 99,98,999
> names (babies)   # show names of the variables from the data.frame babies

> not.these = (gestation == 999) | (age == 99) | (inc == 98) | (dwt == 999) | (ht == 99)
> tmp = babies[!not.these, c("gestation", "age", "wt", "inc", "ht", "dwt")]

> pairs(tmp)       # shows multiple scatter plot for each pair of the variables

> res.lm=lm(wt ~ gestation + age + ht + wt1 + dage + dht + dwt, data=babies, subset=
   gestation<350 & age<99 & ht<99 & wt1 <999 & dage<99 & dht<99 & dwt<999)
> res.lm
> plot(fitted(res.lm), resid(res.lm))  #see also plot(res.lm)
> summary(res.lm)

> library(MASS)    # include library MASS (>Pakete>Installiere Paket(e))
> stepAIC(res.lm)  # choose a model by Akaike-information-criteria in a stepwise algorithm

#  Select a model based on a partial F-test. If a new parameters are not really important than there should
#  be little difference in the sum of squares.

> res.lm1=lm(wt ~ gestation + age + ht, data=babies, subset=
   gestation<350 & age<99 & ht<99 & inc <98)
> res.lm2=update(res.lm1, .~. + inc)
> anova(res.lm1, res.lm2)
```

Example 16

LOGISTIC REGRESSION

The dataset *birthwt* within the library *MASS* contains data on risk factors associated with low infant birth weight. The variable *low* is coded as 0 or 1 to indicate whether the birth weight is low (less than 2500 grams). Perform a logistic regression modeling on *low* by the variables *age*, *lwt* (mothers weight), *smoke* (smoking status), *ht* (hypertension), and *ui* (uterine irritability).

Which variables are flagged as significant? Which model is selected? What is the odds ratio? Calculate 95% confidence interval of the regression coefficients.

```
> library(MASS)    # include library MASS (>Pakete>Installiere Paket(e))
> attach(birthwt)  # include in path so you can use “smoke” instead of “birthwt$smoke”
> birthwt          # show dataset birthwt
> names (birthwt)  # show names of the variables from the data.frame birthwt

> res.glm=glm(low~age+lwt+smoke+ht+ui, family=binomial)  # generalized linear model
> res.glm
> or=prod(exp(res.glm$coeff)) # odds ratio including all parameters
> summary(res.glm)
> stepAIC(res.glm)
```

Example to show logistic regression

```
> n <- 100
> x <- c(rnorm(n), 1+rnorm(n))
> y <- c(rep(0,n), rep(1,n))
> plot(y~x)
> abline(lm(y~x), col='red')
> xp <- seq(min(x),max(x),length=200)
> r <- glm(y~x, family=binomial)
> yp <- predict(r, data.frame(x=xp), type='response')
> lines(xp,yp, col='blue')
```

Example 17

CORRESPONDENCE ANALYIS & PRINCIPAL COMPONENT ANALYSIS

Load gene expression data from http://genome.tugraz.at/biostatistics/microarray.txt and perform a correspondence analysis between genes and samples (since starting point is usually a frequency table take ratios (not log2ratios) and transform to whole numbers) and a principal component analysis.

CA

```
> library(MASS)
> ma<-read.table("http://genome.tugraz.at/biostatistics/microarray.txt", header=TRUE, sep="\t")
> m<-ma[1:nrow(ma),-2]                       # remove gene names
> n<-m[which(duplicated(m$UNIQID)==FALSE),]  # remove duplicated genes
> M<-n[-1]                                   # remove column with UNIQID
> row.names(M)<-n$UNIQID                     # add UNIQID as row names to the data frame
> A<-floor(100*2^M)                          # transformation
> B<-corresp(floor(100*2^M), nf=2)           # correspondence analysis
> v<-rep("+",nrow(M))                        # replace names by symbols
> row.names(B$rscore)<-v                     # replace names by symbols
> biplot(B,cex=c(0.7,1))                     # draw biplot (or directly biplot(B) with names instead of symbols)

# or with library ca

> library(ca)
> B<-ca(A)                                   # correspondence analysis
> plot(B)                                    # draw biplot
```

PCA

```
ma<-read.table("http://genome.tugraz.at/biostatistics/microarray.txt", header=TRUE, sep="\t")
m<-ma[1:nrow(ma),-2]
n<-m[which(duplicated(m$UNIQID)==FALSE),]
M<-n[-1]
row.names(M)<-n$UNIQID
P<-prcomp(M)                                # Principal component analysis
plot(P)                                     # variances of PCs (show importance of PCs)
biplot(P,var.axes=F)                        # scatter plot in 2 dimensional space spanned by the first 2 PCs
PC1<-P$rot[,1]                              # Get values for first PC
plot(PC1, type="o", col="blue")             # Plot profile of first PC
```

Example 18

SURVIVAL ANALYSIS

Perform a survival analysis on the data *lung* form the library *survival*. Construct a Kaplan-Meier survival curve for the censored data (status) and Kaplan-Meier survival curves separately for women (sex=2) and men (sex=1). What are the median survival times? Perform a log-rank test to find out if there is a significant difference between women’s and men’s survival curves and plot the hazard functions. Fit an Exponential and a Weibull distribution to the survival function. Build a Cox regression model with sex and age as explanatory variables and determine regression coefficients. Should both parameters (age and sex) kept in the regression model? Show “expected” survival curves for 2 cases with different defined parameters (eg. age=40, sex=1 and age=90, sex=2).

```
> library(survival)                                                        # include library survival
> attach(lung)
> lung                                                                     # show dataset lung
> lung.surv<-survfit(Surv(time,status),data=lung)                          # estimate survival function
> lung.surv
> summary(lung.surv)
> plot(lung.surv)                                                          # plot Kaplan-Meier survival curve
> lung.surv1<-survfit(Surv(time,status)~sex,data=lung)                     # estimate survival function
> plot(lung.surv1)
> survdiff(Surv(time,status)~sex,data=lung,rho=0)                          # log-rank test
> summary(survreg(Surv(time,status)~1,dist="exponential",data=lung)) )     # fit Exponential function
> summary(survreg(Surv(time,status)~1,dist="weibull",data=lung))           # fit Weibull function


# Cox regression (proportional hazard model)

> fit<-coxph(Surv(time,status)~sex+age,data=lung)
> summary(fit)
> plot(survfit(fit), conf.int=FALSE, lty=2, xlim=c(0,1100),xlab=”survival time (days)”)
> lines(survfit(fit, newdata=data.frame(age=40, sex=1)), col=”blue”, lwd=3)
> lines(survfit(fit, newdata=data.frame(age=90, sex=2)), col="red", lwd=3)

# hazard functions

> sfit <- survfit(Surv(time, status) ~ sex, data=lung)
> temp1 <- smooth.spline(sfit[1]$time, 1-sfit[1]$surv, df=5)
> temp2 <- smooth.spline(sfit[2]$time, 1-sfit[2]$surv, df=5)
> plot( predict(temp1, deriv=1), type='l')
> lines(predict(temp2, deriv=1), col=2)
```