Practical Exercises for **Day 5**

Sonja Hartnack, Terence Odoch & Muriel Buri

October 2017

Exercise 14

- Load the below data set and for further information check the command `?water`.

```
# install.packages("HSAUR3")
library("HSAUR3")
data("water")
str(water)
head(water)
summary(water)
```

- Try to plot the variables `mortality` against `hardness` from the `water` data set.

```
par(mfrow=c(1,1))
plot(x = water$hardness, y = water$mortality)
plot(mortality ~ hardness, data = water)
```

- Add a main title to the above plot (`mortality` against `hardness`).

```
plot(x = water$hardness, y = water$mortality,
     main = "Calcium concentration against mortality")
plot(mortality ~ hardness, data = water,
     main = "Calcium concentration against mortality")
```

- Change the ...

  1. font size of the axis annotation
  2. font size of the x- and y-axis labels
  3. the point sizes within the plot

... of the above plot (`mortality` against `hardness`).

```
# cex: number indicating the amount by which plotting text and symbols
# should be scaled
# cex.axis: magnification of axis annotation relative to cex
plot(x = water$hardness, y = water$mortality,
     cex.axis = 1.5, # (1) enlarge number of the axis
     cex.lab = 1.5, # (2) enlarge font size of axis labels
     cex = 1.5, # (3) enlarge point size within plot
     main = "Calcium concentration vs. mortality")
plot(mortality ~ hardness, data = water,
     cex.axis = 1.5, # enlarge number of the axis
     cex = 1.5, # enlarge point size within plot
     cex.lab = 1.5, # enlarge font size of axis labels
     main = "Calcium concentration vs. mortality")
```

- Looking at the above plot: Do you think the two variables `hardness` and `mortality` correlate? What function do you use to find out the correlation coefficient? Do they have a positive or a negative correlation coefficient? How do you interpret the correlation coefficient in your own words?

```
cor(x = water$hardness, y = water$mortality) # -0.6548486
cor.test(x = water$hardness, y = water$mortality)
# negative correlation of -0.65 with confidence interval of [-0.78, -0.48]:
# the higher the calcium concentration (hardness),
# the smaller the averaged annual mortality per 100.000 male
# inhabitants (mortality)
```

- In the `water` data set, can you graphically find out if there is a difference between the the two variables `hardness` and `mortality` conditional on the `location` (`North`, `South`).

```
plot(x = water$hardness, y = water$mortality,
     col = as.numeric(water$location),
     pch = 16, cex.axis = 1.5,
     cex = 1.5, cex.lab = 1.5)
library("graphics")
coplot(mortality ~ hardness | location, data = water, panel = panel.smooth)
```

- Add a legend to the above plot so that you can easily differentiate the locations (`North` or `South`) of the observations.

```r
plot(x = water$hardness, y = water$mortality,
     col = as.numeric(water$location),
     pch = 16, cex.axis = 1.5,
     cex = 1.5, cex.lab = 1.5)
legend("topright", legend = levels(water$location),
       col = c("black", "red"), pch = 16, cex = 1.5)
```

- Do a barplot of the variable `location` from the `water` data set.

```r
barplot(table(water$location))
```

- ADDITIONAL: Try if any of these following plotting functions can be applied to the data sets `perulung` or `ToothGrowth`.

```r
install.packages("graphics")
library("graphics")
?coplot
#
# install.packages("lattice")
library("lattice")
?xyplot
#
?interaction.plot
```

```r
# PERULUNG DATA SET
coplot(fev1 ~ height | sex, data = lung, panel = panel.smooth)
coplot(fev1 ~ height | respsymptoms, data = lung, panel = panel.smooth)

xyplot(fev1 ~ height | sex, data = lung)
xyplot(fev1 ~ height | respsymptoms, data = lung)

# ToothGrowth DATA SET
interaction.plot(ToothGrowth$dose,
                 ToothGrowth$supp,
                 ToothGrowth$len,
                 fixed = TRUE)
```

Exercise 15

- Download the .R file `ANOVA_with_chickwts.R` from the switch drive and have another look on how we applied the anova to the `chickwts` data set.
- Load the `ToothGrowth` data set into R and encode the numeric variable `dose` as a factor variable. Define the new factor variable as `dose.fac` with the three levels `low`, `med` and `high` and add it to the data frame of `ToothGrowth`.

```
data(ToothGrowth)
str(ToothGrowth)
head(ToothGrowth)
ToothGrowth$dose.fac <- factor(ToothGrowth$dose, levels = c(0.5, 1.0, 2.0),
                               labels = c("low", "med", "high"))
table(ToothGrowth$dose.fac)
```

- Visualize the variable `len` per `dose` level in a boxplot.

```
boxplot(ToothGrowth$len ~ ToothGrowth$dose.fac)
```

- With the help of the R-commands written in the `ANOVA_with_chickwts.R` file, apply a analysis of variance (ANOVA) to the data set `ToothGrowth`

```
# aov.mod <- aov(ToothGrowth$len ~ ToothGrowth$dose.fac)
aov.mod <- aov(len ~ dose.fac, data = ToothGrowth)
# What objects can we extract from a anova model?
objects(aov.mod)
#
summary(aov.mod)

# What are residuals?
ToothGrowth$residuals <- residuals(aov.mod)
```

```r
tapply(ToothGrowth$len, ToothGrowth$dose.fac, mean)
ToothGrowth[c(1:3),]
# Save residuals to an objects and check mean of residuals
aov.mod.resid <- residuals(aov.mod)
mean(aov.mod.resid)
round(mean(aov.mod.resid), 3)

par(mfrow=c(1,1))
qqnorm(aov.mod.resid)
qqline(aov.mod.resid, col = "red", lwd = 3, lty = 2)
# Shapiro-Wilk test (dependent on sample size --> limited use)
shapiro.test(aov.mod.resid)
# a <- rnorm(100, 20, 3)
# qqnorm(a)
# qqplot(a)
# shapiro.test(a)

# Bartlett Test
bartlett.test(ToothGrowth$len ~ ToothGrowth$dose.fac)

# Levene's Test
# install.packages("Rcmdr")
# library("Rcmdr")
# levene.test(ToothGrowth$len ~ ToothGrowth$dose.fac)

# Plot fitted against residual values
objects(aov.mod)
plot(fitted.values(aov.mod), residuals(aov.mod))

# Plot fitted against residual values
par(mfrow=c(1,2), pty="s", mar = c(1, 4, 1, 2))
plot(fitted.values(aov.mod), residuals(aov.mod))
abline(h = 0, col = "red", lwd = 3, lty = 2)
plot(aov.mod, which=1)

# Plot fitted against residual values
# Cut-off at 3 (y-axis)

# observations above 3 are regarded as having high
# influence to the analysis - have a closer look at them:
# outliers? delete them from the data set?
# why are these observations so influencial?
# everything below 3 is okay for the model
par(mfrow=c(1,1), pty="s", mar = c(5, 4, 4, 2))
plot(aov.mod, which=4)
# ToothGrowth[c(22, 23, 32),]

par(mfrow=c(1,3), pty="s")
plot(fitted.values(aov.mod), residuals(aov.mod))
abline(h = 0, col = "red", lwd = 3, lty = 2)
# Plot residuals against variables from the model
plot(ToothGrowth$len, residuals(aov.mod), ylab = "residuals")
plot(ToothGrowth$dose.fac, residuals(aov.mod),
     xlab = "ToothGrowth$dose.fac", ylab = "residuals")

par(mfrow=c(2, 2))
plot(aov.mod)

# # HOW TO RELEVEL FACTORS?
# # How to change the reference category of a factor variable?
# # Use the relevel(...) function
# # Make "sunflower" as reference category
# chickwts$feed <- relevel(chickwts$feed, "sunflower")
# levels(chickwts$feed)
# # Make "linseed" as reference category
# chickwts$feed <- relevel(chickwts$feed, "linseed")
# levels(chickwts$feed)
# chickwts$feed <- relevel(chickwts$feed, "casein")
# levels(chickwts$feed)

aov.mod <- aov(len ~ dose.fac, data = ToothGrowth)
```

```r
# aov.mod1 <- aov(len ~ dose.fac, data = ToothGrowth)
# aov.mod2 <- aov(ToothGrowth$len ~ ToothGrowth$dose.fac)
# summary(aov.mod1)
# summary(aov.mod2)

# DO NOT USE THIS COMMAND, OTHERWISE THE LINEAR FUNCTION WITHIN
# DUNNETT AND TUKEY DOES NOT WORK!
# --> specify the data at the end of the aov model
# aov.mod <- aov(ToothGrowth$len ~ ToothGrowth$dose.fac)
summary(aov.mod)
pairwise.t.test(ToothGrowth$len, ToothGrowth$dose.fac, p.adj = "none")
pairwise.t.test(ToothGrowth$len, ToothGrowth$dose.fac, p.adj = "bonferroni")

# install the package first (one time)
# install.packages("multcomp")
# load the library (every single time you use it!)
library("multcomp")
# compares always to baseline levels (here: casein) --> saves degrees of freedom
# make sure you saved the aov.mod as:
# aov.mod <- aov(len ~ dose.fac, data = ToothGrowth)
dunnett <- glht(aov.mod, linfct = mcp(dose.fac = "Dunnett"))
summary(dunnett)

library("multcomp")
# compares all factor levels
tukey <- glht(aov.mod, linfct = mcp(dose.fac = "Tukey"))
summary(tukey)
# summary(tukey)          # standard display
tukey.cld <- cld(tukey)   # letter-based display
# the cld(...) function sets up a compact letter display of all pair-wise comparisons
par(mfrow=c(1,1), mar=c(5,4,8,2))
plot(tukey.cld)
```

Exercise 16

- Download the .R file `LM_with_water.R` from the switch drive and have another look on how we applied the linear model to the `water` data set.

- Reuse these commands to fit a simple as well as multiple linear regression model to the data set of `perulung_ems`. Use `fev1` as your response variable y.

```r
lung <- read.csv("~/Dropbox/data/perulung_ems.csv", sep = ";")
head(lung)
str(lung)
lung$sex <- factor(lung$sex, levels = c("0", "1"))
levels(lung$sex) <- c("female", "male")
lung$respsymptoms <- factor(lung$respsymptoms, levels = c("0", "1"))

# MODEL 1
# mod.age <- lm(fev1 ~ age, data = lung)
mod.age <- lm(lung$fev1 ~ lung$age)
summary(mod.age)
coef(mod.age)
# Check model assumptions graphically
par(mfrow=c(2,2))
plot(mod.age)

# MODEL 2
# mod.height <- lm(fev1 ~ height, data = lung)
mod.height <- lm(lung$fev1 ~ lung$height)
summary(mod.height)
coef(mod.height)
# Check model assumptions graphically
par(mfrow=c(2,2))
plot(mod.height)

# MODEL 3
mod.age.height <- lm(fev1 ~ age + height, data = lung)
summary(mod.age.height)
coef(mod.age.height)
```

```r
# Check model assumptions graphically
par(mfrow=c(2,2))
plot(mod.age.height)

# MODEL 4
mod.age.height.sex <- lm(fev1 ~ age + height + sex, data = lung)
summary(mod.age.height.sex)
coef(mod.age.height.sex)
# Check model assumptions graphically
par(mfrow=c(2,2))
plot(mod.age.height.sex)

# MODEL 5
mod.age.height.sex.resp <- lm(fev1 ~ age + height + sex + respsymptoms,
                              data = lung)
summary(mod.age.height.sex.resp)
coef(mod.age.height.sex.resp)
# Check model assumptions graphically
par(mfrow=c(2,2))
plot(mod.age.height.sex.resp)

mod1 <- lm(lung$fev1 ~ lung$age)
mod2 <- lm(lung$fev1 ~ lung$height)
mod3 <- lm(fev1 ~ age + height, data = lung)
mod4 <- lm(fev1 ~ age + height + sex, data = lung)
mod5 <- lm(fev1 ~ age + height + sex + respsymptoms,
           data = lung)
summary(mod5)

# MODEL SELECTION
AIC(mod1, mod2, mod3, mod4, mod5)
round(AIC(mod1, mod2, mod3, mod4, mod5), 2)
# Which of the models is best?
par(mfrow=c(2,2))
plot(mod5)
```