Practical Exercises for **Exercise Collection**

Sonja Hartnack, Terence Odoch & Muriel Buri

October 2018

Exercise 15

(a) Open the .R file `ANOVA_with_chickwts.R` from your RCourse folder and have another look on how we applied the anova to the `chickwts` data set. Check line for line.

(b) Load the `ToothGrowth` data set into R and encode the numeric variable `dose` as a factor variable. Define the new factor variable as `dose.factor` with the three levels `low`, `med` and `high` and add it to the data frame of `ToothGrowth`.

```
data(ToothGrowth)
```

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

(c) Visualize the variable `len` per `dose.factor` level in a boxplot.

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

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

```r
#
summary(aov.mod)

# What are residuals?
ToothGrowth$residuals <- residuals(aov.mod)
tapply(ToothGrowth$len, ToothGrowth$dose.factor, 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.factor)

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

# 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.factor, residuals(aov.mod),
     xlab = "ToothGrowth$dose.factor", 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.factor, data = ToothGrowth)

# aov.mod1 <- aov(len ~ dose.factor, data = ToothGrowth)
# aov.mod2 <- aov(ToothGrowth$len ~ ToothGrowth$dose.factor)
# 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.factor)
summary(aov.mod)
pairwise.t.test(ToothGrowth$len, ToothGrowth$dose.factor, p.adj = "none")
pairwise.t.test(ToothGrowth$len, ToothGrowth$dose.factor, 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.factor, data = ToothGrowth)
dunnett <- glht(aov.mod, linfct = mcp(dose.factor = "Dunnett"))
summary(dunnett)

library("multcomp")
# compares all factor levels
tukey <- glht(aov.mod, linfct = mcp(dose.factor = "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
```

```r
par(mfrow=c(1,1), mar=c(5,4,8,2))
plot(tukey.cld)
```

Exercise 16

(a) Reuse the commands from the lecture slides to fit a simple as well as a multiple linear regression model to the data set of `perulung_ems`. Use `fev1` as your response variable y.

(b) Check the model assumptions.

(c) Which model is best?

```r
lung <- read.csv("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)
# 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)
```

```r
# Which of the models is best?
par(mfrow=c(2,2))
plot(mod5)
```

Exercise 17

(a) Load the `ToothGrowth` data set and run the following four linear regression models.

```r
data(ToothGrowth)
ToothGrowth$dose.factor <- factor(ToothGrowth$dose, levels = c(0.5, 1.0, 2.0),
                                  labels = c("low", "med", "high"))
mod1 <- lm(len ~ dose.factor, data = ToothGrowth)
mod2 <- lm(len ~ supp, data = ToothGrowth)
mod3 <- lm(len ~ dose.factor + supp, data = ToothGrowth)
mod4 <- lm(len ~ dose.factor*supp, data = ToothGrowth)
```

(b) Have a look at the summary of these models.

```r
mod1 <- lm(len ~ dose.factor, data = ToothGrowth)
mod2 <- lm(len ~ supp, data = ToothGrowth)
mod3 <- lm(len ~ dose.factor + supp, data = ToothGrowth)
# mod4 <- lm(len ~ dose.factor + supp + dose.factor:supp, data = ToothGrowth)
mod4 <- lm(len ~ dose.factor*supp, data = ToothGrowth)
summary(mod1)
summary(mod2)
summary(mod3)
summary(mod4)
# Check model assumptions
par(mfrow=c(2, 2))
plot(mod1)
plot(mod2)
plot(mod3)
plot(mod4)
```

(c) How do you interpret the model coefficients?

(d) Which model is best?

```
AIC(mod1, mod2, mod3, mod4)
# t.test(ToothGrowth$len ~ ToothGrowth$supp) # not significant
# mod4 is the best model, because it has the smallest AIC.
# THE SMALLER THE AIC, THE BETTER THE MODEL!
```