# Practical Exercises for **Day 3**

Sonja Hartnack, Terence Odoch & Muriel Buri

October 2018

## Exercise 5

(a) How many levels has the factor variable `trt` from `bacteria`?

```
str(bacteria)
head(bacteria$trt)
table(bacteria$trt)
levels(bacteria$trt)
nlevels(bacteria$trt)
```

(b) Define a new variable `trt.new` in which you combine the levels `drug` and `drug+` into one single level and label it as `treated`. The new variable `trt.new` should in the end have two levels: `placebo` and `treated`.

```
table(bacteria$trt)
# OPTION 1:
# Test how many levels are in the variable "trt"?
levels(bacteria$trt)
bacteria$trt.new <- bacteria$trt
# Overwrite the levels "placebo", "drug", "drug+" with new
# levels called "placebo", "drug", "drug" --> combine "drug" and "drug+"
levels(bacteria$trt.new) <- c("placebo", "drug", "drug")
# Do table for variable "trt" and "trt.new" to see if you combined correctly
table(bacteria$trt)
table(bacteria$trt.new)
# Rename the levels from "placebo", "drug" to "placebo", "treated"
levels(bacteria$trt.new) <- c("placebo", "treated")
# Do another table to check if you did everything correctly:
table(bacteria$trt.new)
```

(c) Do summary statistics for `placebo` and `treated` group.

```
summary(bacteria)
table(bacteria$trt.new)
barplot(table(bacteria$trt.new))
table(bacteria$trt.new, bacteria$ap)
table(bacteria$trt.new, bacteria$y)
plot(table(bacteria$trt.new, bacteria$y))
fisher.test(table(bacteria$trt.new, bacteria$y))    # Fisher's exact test
chisq.test(table(bacteria$trt.new, bacteria$y))     # Chi-squared test
# odds ratio: (a*d)/(b*c)
(12*93)/(31*84)
```

## Exercise 6

(a) Load data set `ToothGrowth`.

```
data(ToothGrowth)
str(ToothGrowth)
head(ToothGrowth)
summary(ToothGrowth)
```

(b) Do summary statistic (numerically and graphically).

```
# NUMERICAL statistics
# supplement
table(ToothGrowth$supp)
tapply(ToothGrowth$len, ToothGrowth$supp, mean)
tapply(ToothGrowth$len, ToothGrowth$supp, median)
tapply(ToothGrowth$len, ToothGrowth$supp, sd)
# dose
tapply(ToothGrowth$len, ToothGrowth$dose, mean)
tapply(ToothGrowth$len, ToothGrowth$dose, median)
tapply(ToothGrowth$len, ToothGrowth$dose, sd)
# GRAPHICAL exploration
# length
```

```
hist(ToothGrowth$len)
boxplot(ToothGrowth$len, ylab = "tooth length",
        main = "Boxplot of the tooth length")
# supplement
barplot(table(ToothGrowth$supp), legend.text = TRUE)
boxplot(ToothGrowth$len ~ ToothGrowth$supp, , ylab = "tooth length",
        main = "Boxplot of the tooth length separately plotted per supplement")
# dose
barplot(table(ToothGrowth$dose), legend.text = TRUE)
# display tooth length increase in dosage
boxplot(ToothGrowth$len ~ ToothGrowth$dose, , ylab = "tooth length",
        main = "Boxplot of the tooth length separately plotted per dose")
# install.packages("graphics")
library("graphics")
coplot(len ~ dose | supp, data = ToothGrowth, panel = panel.smooth,
       xlab = "ToothGrowth data: length vs dose, given type of supplement")
```

(c) Define additional column `dose.factor` by converting the numeric variable `dose` into a factor variable.

```
class(ToothGrowth$dose)
str(ToothGrowth$dose)
table(ToothGrowth$dose)
# ToothGrowth$dose.factor <- as.factor(ToothGrowth$dose)
ToothGrowth$dose.factor <- factor(ToothGrowth$dose, levels = c("0.5", "1", "2"))
class(ToothGrowth$dose.factor)
table(ToothGrowth$dose.factor)
```

(d) Are the tooth length measurements normally distributed within the treatment (`supp`: VC or OJ) and within in the different doses (`dose`: 0.5, 1, 2)?

```
# supp: VC, OJ
sub.OJ <- subset(ToothGrowth, supp == "OJ")
sub.VC <- subset(ToothGrowth, supp == "VC")
# graphically
par(mfrow = c(1,2))
qqnorm(sub.OJ$len, main = "Normal Q-Q Plot for OJ")
```

```r
qqline(sub.OJ$len)
qqnorm(sub.VC$len, main = "Normal Q-Q Plot for VC")
qqline(sub.VC$len)
# with a statistical test
shapiro.test(sub.OJ$len)
shapiro.test(sub.VC$len)
# dose: 0.5, 1, 2
sub.0.5 <- subset(ToothGrowth, dose.factor == "0.5")
sub.1 <- subset(ToothGrowth, dose.factor == "1")
sub.2 <- subset(ToothGrowth, dose.factor == "2")
# graphically
qqnorm(sub.0.5$len)
qqline(sub.0.5$len)
qqnorm(sub.1$len)
qqline(sub.1$len)
qqnorm(sub.2$len)
qqline(sub.2$len)
# with a statistical test
shapiro.test(sub.0.5$len)
shapiro.test(sub.1$len)
shapiro.test(sub.2$len)
```

## Exercise 7

(a) Import the data set `perulung_ems.csv` (taken from Kirkwood and Sterne, 2nd edition) into R.
Data from a study of lung function among children living in a deprived suburb of Lima, Peru.
Variables:

- `fev1`: in liter, "Forced Expiratory Volume in 1 second" measured by a spirometer. This is the maximum volume of air which the children could breath out in 1 second
- `age`: in years
- `height`: in cm
- `sex`: 0 = girl, 1 = boy
- `respsymp`: respiratory symptoms experienced by the child over the previous 12 months

(b) What *delimiter* do you need to choose?

```
lung <- read.csv("perulung_ems.csv", sep = ";")
head(lung)
str(lung)
```

(c) Do all variables have the correct data type (numeric, integer, factor)? If not, do correct and / or define them.

```
head(lung)
str(lung)
lung$sex <- factor(lung$sex, levels = c("0", "1"))
# levels(lung$sex) <- c("female", "male")
# levels(lung$sex)[levels(lung$sex)=="0"] <- "female"
# levels(lung$sex)[levels(lung$sex)=="1"] <- "male"
# tapply(lung$fev1, lung$sex, mean)
lung$respsymptoms <- factor(lung$respsymptoms, levels = c("0", "1"))
```

```
library(usdm)
# check for multicollinearity by using variance inflation factors
# cerate a dataframe just with the three continuous/numeric variables fevs, age and height
try.vif <- lung[,c("fev1","height","age")];
# perform scatterplots for these three variables
pairs(try.vif)
# get the three VIF, as a rule of thumb they should be < 3
vif(try.vif)
```

(d) Check for heteroscedascity or homogeneity of variances

```
?bartlett.test
data("chickwts")
bartlett.test(weight ~ feed, data = chickwts)
```

## Exercise 8

Apply the summary statistics to the `perulung_ems` and `ToothGrowth` data set.

```r
# Read in .csv data
lung <- read.csv("perulung_ems.csv", sep = ";")
head(lung)
str(lung)
summary(lung)
lung$sex <- factor(lung$sex, levels = c("0", "1"))
levels(lung$sex) <- c("female", "male")
lung$respsymptoms <- factor(lung$respsymptoms, levels = c("0", "1"))
# Continuous and factor
tapply(lung$height, lung$sex, mean)
tapply(lung$height, lung$respsymptoms, mean)
# Factor and factor
table(lung$respsymptoms, lung$sex)
prop.table(table(lung$respsymptoms, lung$sex))
# Continuous and factor
tapply(lung$age, lung$sex, mean)
tapply(lung$age, lung$respsymptoms, mean)
# Continuous and factor
tapply(lung$fev1, lung$sex, mean)
tapply(lung$fev1, lung$respsymptoms, mean)
# Continuous and continuous
pairs(lung)
cor.test(lung$fev1, lung$age, method = "pearson")
cor.test(lung$fev1, lung$height, method = "pearson")
# ToothGrowth
summary(ToothGrowth)
table(ToothGrowth$supp)
tapply(ToothGrowth$len, ToothGrowth$supp, mean)
tapply(ToothGrowth$len, ToothGrowth$supp, median)
tapply(ToothGrowth$len, ToothGrowth$supp, sd)
table(ToothGrowth$dose)
tapply(ToothGrowth$len, ToothGrowth$dose, mean)
tapply(ToothGrowth$len, ToothGrowth$dose, median)
tapply(ToothGrowth$len, ToothGrowth$dose, sd)
```

## Exercise 9A: Plausibility Checks

(a) What can go wrong?

(b) Identify different strategies for spotting these potential errors.

- Logical errors
- Spelling mistakes

(c) Import the data set `bacteria_plausibility_check.csv` to R.

```
bac <- read.csv("bacteria_plausibility_check.csv", sep = ",")
head(bac)
str(bac)
summary(bac)
```

(d) Detect the **six** errors in the imported data set `bacteria_plausibility_check.csv` in R.

```
str(bac)
table(bac$y) # We have wrong factor levels: 0, 1
table(bac$ap)
table(bac$hilo) # We have a spelling mistake: Hi.
table(bac$week) # There's only ONE observation in week 20.
table(bac$ID)
table(bac$trt) # We have wrong factor levels: drug++, penicillin+
summary(bac$child_weight) # child weight of 302.8 kg is impossible --> comma
```

(e) Find possible solutions in R how to handle these challenges.

```
bac$y[which(bac$y == 0)] <- "n"
# bac$y[bac$y == 0] <- "n"
bac$y[which(bac$y == 1)] <- "y"
# Delete the unused levels with the function droplevels(...)
bac$y <- droplevels(bac$y)
bac$hilo[bac$hilo == "Hi"] <- "hi"
bac$hilo[which(bac$hilo == "Hi")] <- "hi"
levels(bac$hilo) <- c("hi", "hi", "lo")
summary(bac)
```

```r
bac <- bac[-which(bac$week == 20), ] # dim(bac)
bac$trt[bac$trt == "drug++"] <- "drug+"
bac$trt[bac$trt == "penicillin+"] <- "drug+"
table(bac$trt) # We have wrong factor levels: drug++, penicillin+
levels(bac$trt) <- c("drug", "drug+", "drug+", "drug", "placebo")
table(bac$trt) # We only have three factor levels left.
bac$child_weight[bac$child_weight == 302.8] <- 30.28
summary(bac)
```

(f) Do all variables have the correct data type (numeric, integer, factor)? - If not, do correct / define them.

```r
bac$y <- factor(bac$y, levels = c("n", "y"))
bac$hilo[bac$hilo == "Hi"] <- "hi"
bac$ID <- factor(bac$ID)
bac$trt <- factor(bac$trt)
```