Practical Exercises for **Wednesday, July 24, 2019, Day 3**

Sonja Hartnack, Terence Odoch & Muriel Buri

July 2019

Exercise 1: Get to know `bacteria` data set

(a) Install package `MASS`. Load data set `bacteria`.

```
# install.packages("MASS")
library("MASS")
data(bacteria)
head(bacteria)
str(bacteria)
summary(bacteria)
# ?bacteria
```

(b) Describe in your own words what the data set `bacteria` contains.

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

```
summary(bacteria)
table(bacteria$week)
barplot(table(bacteria$week))
barplot(table(bacteria$trt))
table(bacteria$trt, bacteria$ap)
table(bacteria$trt, bacteria$y)
%
fisher.test(table(bacteria$trt, bacteria$y))
%
prop.table(table(bacteria$trt, bacteria$y))
prop.table(table(bacteria$trt, bacteria$y), margin = 1)
prop.table(table(bacteria$trt, bacteria$y), margin = 2)
%
plot(prop.table(table(bacteria$trt, bacteria$y)))
mosaicplot(~trt + y, data = bacteria)
barplot(prop.table(table(bacteria$y, bacteria$trt),margin=1), beside=TRUE)
```

```
barplot(prop.table(table(bacteria$trt, bacteria$y),margin=1), beside=TRUE)
barplot(prop.table(table(bacteria$y, bacteria$trt),margin=1), beside=FALSE)
barplot(prop.table(table(bacteria$trt, bacteria$y),margin=1), beside=FALSE)
?barplot
```

(d) Select only observations collected during the second week.

```
subset(bacteria, week == 2)
ss <- subset(bacteria, week == 2)
summary(ss)
# Check if we only have observations of week 2.
table(bacteria$week)
table(ss$week)
```

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

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

(f) 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"
```

```r
levels(bacteria$trt.new) <- c("placebo", "treated")
# Do another table to check if you did everything correctly:
table(bacteria$trt.new)

### Easiest solution:
bacteria$trt.new <- bacteria$trt
summary(bacteria)
levels(bacteria$trt.new) <- c("placebo", "treated", "treated")
summary(bacteria)

## OPTION 1:
bacteria$trt.new <- bacteria$trt
summary(bacteria)
# "drug"
levels(bacteria$trt.new)[2] <- "treated"
levels(bacteria$trt.new)
# "drug+"
levels(bacteria$trt.new)[3] <- "treated"
levels(bacteria$trt.new)

## OPTION 2:
bacteria$trt.new <- bacteria$trt
summary(bacteria)
bacteria$trt.new <- as.character(bacteria$trt.new)
summary(bacteria)
# change the levels
bacteria$trt.new[bacteria$trt.new == "drug"] <- "treated"
bacteria$trt.new[bacteria$trt.new == "drug+"] <- "treated"
bacteria$trt.new[bacteria$trt.new == "placebo"] <- "placebo"
summary(bacteria)
# change it back to factor
bacteria$trt.new <- as.factor(bacteria$trt.new)
summary(bacteria)
```

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

```r
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 2: Data 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.

```r
bac <- read.csv("data/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.

```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.

```r
levels(bac$y) <- c("n", "y", "n", "y")
bac$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)
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$trt) <- c("drug", "drug+", "drug+", "drug", "placebo")
# levels(bac$trt) <- droplevels(bac$trt)
table(bac$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
str(bac)
# 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)
```

Exercise 3: Missing values

(a) Check out the difference between the different missing values.

```r
y1 <- c(2, 4, 3, NA, 6, 1)
y2 <- c("diseased", "healthy", NA, "NA")
y3 <- c(1, "NA", 0, 1, NaN)
#
is.na(y1)
```

```
which(is.na(y1))
is.na(y2)
which(is.na(y2))
is.na(y3)
which(is.na(y3))
is.nan(y3)
```

```
y4 <- c(1, NA, 0, 1, NaN)
# finds the NaN
is.nan(y4)
# finds the NA and the NaN
is.na(y4)
# this vector is a character and NA as well as NaN cannot be detected
y5 <- c(1, "NA", 0, 1, NaN)
```

(b) Create a vector with missing values and determine the mean and median.

```
myvector <- c(1:3, NA, NA, 1:3)
mean(myvector)
mean(myvector, na.rm=TRUE) # calculates c(1, 2, 3, 1, 2, 3)
median(myvector, na.rm=TRUE)
```

(c) If `x <- c(22,3,7,NA,NA,67)` what will be the output for the R statement `length(x)`?

```
x <- c (22, 3, 7, NA, NA, 67)
length(x)
```

(d) If `x <- c(NA, 3, 14, NA, 33, 17, NA, 41)` which line of R code removes all occurrences of NA in x.

```
x <- c(NA,3,14,NA,33,17,NA,41)
x[!is.na(x)]
x[is.na(x)]
x[which(is.na(x))] <- 0
```

(e) If `y <- c(1, 3, 12, NA, 33, 7, NA, 21)` what R statement will replace all occurrences of NA with 11?

```
y <- c(1,3,12,NA,33,7,NA,21)
y[y=="NA"] <- 11
y[is.na(y)] <- 11
y[y==11] <- NA
```

(f) If `x <- c(34, 33, 65, 37, 89, NA, 43, NA, 11, NA, 23, NA)` then what will count the number of occurrences of NA in `x`?

```
x <- c(34,33,65,37,89,NA,43,NA,11,NA,23,NA)
sum(x=="NA")
sum(x == "NA", is.na(x))
sum(is.na(x))
mean(is.na(x))
```

(g) Create the vector `x1`. Then, find again the number of missing values and their position.

```
x1 <- c(rnorm(10,5,2), NA, 5:12, NA, 6, 7.5, NA)
```

```
x1 <- c(rnorm(10,5,2), NA, 5:12, NA, 6, 7.5, NA)
is.na(x1)
summary(x1)
sum(is.na(x1))
which(is.na(x1))
```

(h) Now, create the vector `x2` and assess the difference to `x1`.

```
x2 <- c(rnorm(10,5,2), NA, 5:12, NA, 6, 7.5, NA, log(-2))
```

```
x2 <- c(rnorm(10,5,2), NA, 5:12, NA, 6, 7.5, NA, log(-2))
x2
is.na(x2)
summary(x2)
sum(is.na(x2))
which(is.na(x2))
```

(i) What is the meaning of "NA" versus "NaN"?

(j) Replace the missing values in `x1` with a 0. Check then that the NAs are no longer present. Try two different commands to coerce the NAs into 0.

```r
x1 <- c(rnorm(10,5,2), NA, 5:12, NA, 6, 7.5, NA)
x1[is.na(x1)] <- 0
is.na(x1)
# or with the ifelse statement
x1 <- c(rnorm(10,5,2), NA, 5:12, NA, 6, 7.5, NA)
ifelse(is.na(x1), 0, x1)
is.na(x1)
```