Practical Exercises for **Day 3**

Sonja Hartnack, Terence Odoch & Muriel Buri

October 2017

Exercise 8

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

```r
# Read in .csv data
lung <- read.csv("C:\\Users\\Exercises\\data\\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

- What can go wrong?
- Identify different strategies for spotting these potential errors.
  - Logical errors
  - Spelling mistakes
- Import the data set `bacteria_plausibility_check.csv` to R.

```
# OPTION 1:
# install.packages("readr")
library("readr")
bac <- read_delim("~/Dropbox/data/bacteria_plausibility_check.csv",
                  ";", escape_double = FALSE, trim_ws = TRUE)
bac <- data.frame(bac)
# OPTION 2:
# Import .csv file with the help of the read.csv function
# Be sure to add sep = ";" so that we separate the columns.
bac <- read.csv("~/Dropbox/data/bacteria_plausibility_check.csv", sep = ",")
head(bac)
str(bac)
summary(bac)
```

- 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
```

- 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$hilo) <- c("hi", "hi", "lo")
summary(bac)
bac <- bac[-which(bac$week == 20), ] # dim(bac)
bac$trt[bac$trt == "drug++"] <- "drug+"
bac$trt[bac$trt == "penicillin+"] <- "drug+"
table(bac$trt) # We have wrong factor levels: drug++, penicillin+
bac$child_weight[bac$child_weight == 302.8] <- 30.28
summary(bac)
```

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

```
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 9B: Missing Values

- Check out the difference between the different missing values

```
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)
```

- 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)
```

- 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)
```

- 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
```

- 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
```

- 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))
```

- Create a vector and find the number of missing values and their position

```
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))
```

- 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
```

- What is the meaning of "NA" versus "NaN"?
- Replace the missing values in x1 with a 0, and check that no NAs are present try two different commands to coerce the NAs into 0

```
x1[is.na(x1)] <- 0
is.na(x1)
# or
ifelse(is.na(x1),0,x1)
```

Exercise 10

- Import the data set `water_errors.csv` to R: A data frame with 61 observations on the following 6 variables.
  - **location**: a factor with levels `North` and `South` indicating whether the town is as north as Derby.
  - **town**: the name of the town.

- **mortality**: averaged annual mortality per 100.000 male inhabitants.
- **hardness**: calcium concentration (in parts per million).
- **smoker**: If there are any smokers living in town.
- **num.of.cig**: In case, smokers live in town, what number of cigarettes do they smoke per day.

```r
# H2O_err <- read_csv("C:\\Users\\admin\\Dropbox\\data\\water_errors.csv")
# str(H2O_err)
# H2O_err <- data.frame(H2O_err)
# str(H2O_err)
# BEST SOLUTION how to read it in:
# Try to use the "read.csv(...)" function to read data in!
# use the separator sep=";" or sep="," - which ever works better.
H2O_err <- read.csv("C:\\Users\\admin\\Dropbox\\data\\water_errors.csv", sep=",")
str(H2O_err)

# H2O_err <- read_csv("~/Dropbox/201710_Makerere/03_Exercises/data/water_errors.csv")
H2O_err <- read.csv("~/Dropbox/data/water_errors.csv", sep=",")
H2O_err <- data.frame(H2O_err)
str(H2O_err)
head(H2O_err)
```

- Detect the errors in the imported data set `water_errors.csv` in R.

```r
str(H2O_err)
table(H2O_err$location) # Only one N and only one West observation.
table(H2O_err$town) # LIVERPOOL is in capital letter.
summary(H2O_err$mortality)
summary(H2O_err$hardness) # hardness of -2 does not make sense, two NA's
table(H2O_err$num.of.cig) # only one "zero" observation (wrong coding / level)
table(H2O_err$smoker, H2O_err$num.of.cig) # non-smokers who smoke more than 20?
```

- Find possible solutions in R how to handle these challenges.

```r
str(H2O_err)
which(H2O_err$location == "N") # 6th row
which(H2O_err$location == "West") # 9th row
```

```
H2O_err$location[H2O_err$location == "N"] <- "North"

H2O_err$location[H2O_err$location == "West"] <- NA # Option 1: Set to NA.

dim(H2O_err)

H2O_err <- H2O_err[-which(H2O_err$location == "West"), ] # Option 2: Remove from data.

dim(H2O_err)

# H2O_err$town[H2O_err$town == "LIVERPOOL"] <- "Liverpool"

# H2O_err <- H2O_err$town[-which(H2O_err$town == "LIVERPOOL"), ]

which(is.na(H2O_err$hardness))

H2O_err$hardness[which(is.na(H2O_err$hardness))] <- NA

H2O_err$hardness[which(H2O_err$hardness == -2)] <- NA

# H2O_err$hardness[which(H2O_err$hardness == -2)] <- 2

summary(H2O_err$hardness)

# Check levels of varibale num.of.cig

levels(H2O_err$num.of.cig)

table(H2O_err$num.of.cig)

# Change the zero level to none

H2O_err$num.of.cig[H2O_err$num.of.cig == "zero"] <- "none"

# Drop unused levels

H2O_err$num.of.cig <- droplevels(H2O_err$num.of.cig)

# levels(droplevels(H2O_err$num.of.cig))

table(H2O_err$num.of.cig)


which.F.morethan20 <- which(H2O_err$smoker == FALSE & H2O_err$num.of.cig == "more than 20")

H2O_err[which.F.morethan20, ]

# OPTION 1:

H2O_err$num.of.cig[which.F.morethan20] <- NA

# OPTION 2:

H2O_err$smoker[which.F.morethan20] <- TRUE

# check again, that we corrected it right

H2O_err[which.F.morethan20, ]

table(H2O_err$smoker, H2O_err$num.of.cig) # check again!


which(H2O_err$smoker == FALSE & H2O_err$num.of.cig == "more than 20")

which.T.none <- which(H2O_err$smoker == TRUE & H2O_err$num.of.cig == "none")

H2O_err[which.T.none, ]

H2O_err$smoker[which.T.none] <- FALSE
```

```
table(H2O_err$smoker, H2O_err$num.of.cig)
```

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

```
str(H2O_err)
levels(H2O_err$location)
H2O_err$location <- factor(H2O_err$location, levels = c("North", "South", NA),
                           exclude = NULL)
levels(H2O_err$smoker)
H2O_err$smoker <- factor(H2O_err$smoker, levels = c("FALSE", "TRUE"))
table(H2O_err$num.of.cig)
H2O_err$num.of.cig <- factor(H2O_err$num.of.cig,
                             levels = c("none", "less than 5", "5 to 20", "more than 20"),
                             ordered = TRUE)
table(H2O_err$num.of.cig)
levels(H2O_err$num.of.cig)
levels(H2O_err$num.of.cig) <- c("none", "1 to less than 5", "5 to 20", "more than 20")
table(H2O_err$num.of.cig)
levels(H2O_err$num.of.cig)
```