Laboratory Session : April 1, 2020

Exercises due on : April 15, 2020

Exercise 1 - vectors and data frames

- The following table gives the volume, area, length and maximum and mean depths of some Scottish lakes[1]. Create vectors, holding the lake's name and all the parameters and build a dataframe called `scottish.lakes` from the vectors

1 evaluate the highest and lowest volume and area lake

2 order the frame with respect to the area and determine the two largest area lakes

3 by summing up the areas occpupied by the lakes, determine the area of Scotland covered by water

Loch	Volume [km^3]	Area [km^2]	Length [km]	Max. depth [m]	Mean depth [m]
Loch Ness	7.45	56	39	230	132
Loch Lomond	2.6	71	36	190	37
Loch Morar	2.3	27	18.8	310	87
Loch Tay	1.6	26.4	23	150	60.6
Loch Awe	1.2	39	41	94	32
Loch Maree	1.09	28.6	20	114	38
Loch Ericht	1.08	18.6	23	156	57.6
Loch Lochy	1.07	16	16	162	70
Loch Rannoch	0.97	19	15.7	134	51
Loch Shiel	0.79	19.5	28	128	40
Loch Katrine	0.77	12.4	12.9	151	43.4
Loch Arkaig	0.75	16	19.3	109	46.5
Loch Shin	0.35	22.5	27.8	49	15.5

Exercise 2

- install the `DAAG`[2] and the `tibble`[3] packages:

- `install.packages(c('DAAG','tibble'), type='source')`

- after having loaded the library, get information on the package content and on the `ais` data frame

- create a `tibble` from the `ais data.frame` and perform the following analysies:

1) create a table grouping the data by gender and by sport; produce a barplot with the table adding a legend

2) determine if any of the columns holds missing values

   Hint: the function `any(condition)` applied on a vector returns a 1-dim vector with `TRUE` if `condition` is verified for any of the elements of the vector:

```
vc <- c(1,3,5,-2,4)
any(vc < 0)
[1] TRUE
```

3) produce `boxplots` of the main blood variables ('red blood cell counts', 'white blood cell counts', 'hematocrit' and 'hemaglobin concentration'), for different kind of sports

4) make some scatter plot correlations of the same blood variables using different colors and symbols for the two genders in the sample

Exercise 3

- download the latest update on the COVID-19 Virus infection fom the European Centers for Disease Control[4] using the following code:

```
needed_packages <- c('lubridate', 'readxl', 'curl')

already_installed <- needed_packages %in% installed.packages()
for (pack in needed_packages[!already_installed]) {
    message(paste("To be installed:", pack, sep=" "))
    install.packages(pack)
}

library(lubridate)
library(readxl)
library(curl)

url <- "https://www.ecdc.europa.eu/sites/default/files/documents/"
fname <- "COVID-19-geographic-disbtribution-worldwide-"
date <- lubridate::today() - 1
ext = ".xlsx"
target <- paste(url, fname, date, ext, sep="")
message("target: ", target)
tmp_file <- tempfile("data", "/tmp", fileext=ext)
tmp <- curl::curl_download(target , destfile=tmp_file)

covid <- readxl::read_xlsx(tmp_file)
```

1) examine the loaded tibble structure

2) create a sub-tibble containing only the last day (hint: apply a selection on `dateRep` column) and produce a table with all the countries with number of deaths or number of new cases greater than 200

   Himt: use `order(vector)` and `order(-vector)` to arrange the tibble in ascending or descending order, respectively

3) select the top 10 countries, in terms of cases, and plot the total number of cases as a function of time. Plot the total number of deaths as a function of time. In order to compare the different curves, normalize the first date-time plot to the same t_0 value.

Bibliography

[1] Lakes of Scotland: https://en.wikipedia.org/wiki/List_of_lochs_of_Scotland

[2] DAAG: Data Analysis and Graphics Data and Functions
https://cran.r-project.org/web/packages/DAAG/index.html

[3] Tibble: Simple Data Frames https://cran.r-project.org/web/packages/tibble/index.html

[4] European Centers for Disease Control https://www.ecdc.europe.eu/en