IV. R Markdown (presentation)

Data Science Lab, University of Copenhagen

14 March, 2022

This document doesn't include much, because we will simply review some features along the way. You are encouraged to work with the exercises (which include explanations) and ask us questions either during the course or later.

The R Markdown cheat sheet

The exercise you will do with R markdown will cover the following:

- Open new markdown document, look at its structure. Knit!
- Change some text, headers, etc.
- R chunks: Knit, run code without knitting, good practice
- Options in R chunks
- Example: table1
- Example2: kable
- Other output formats: docx, pdf, latex (Using the dropdown menu by the Knit button)
- Good practice: See the list in the end of the document with exercises

some more text formatting

You can easily use *italic* text and **bold** text. You can also mark text as `inline code`.

Some simple commands

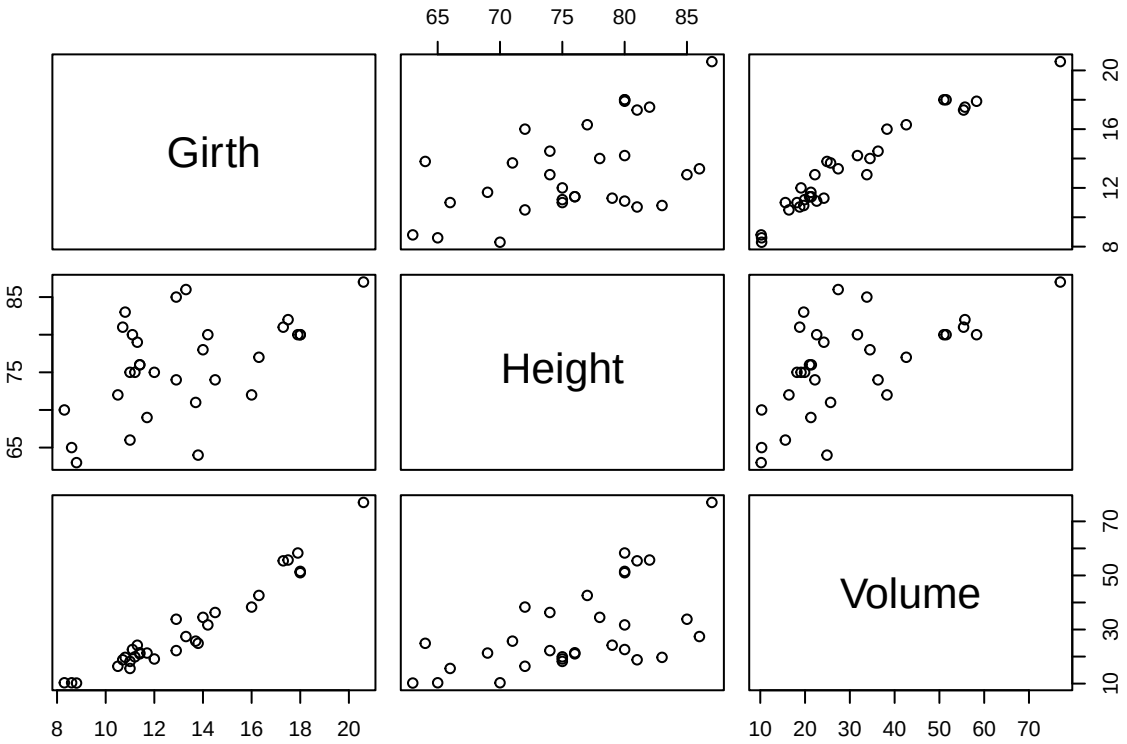
```
11 * 5
```

```
## [1] 55
```

```
sqrt(25)
```

```
## [1] 5
```

```
plot(trees)
```



Options in R chunks

It is possible to control if code and/or output is shown in the knitted document.

First, some code with the default settings (both code and output shown):

```
reg <- lm(Volume ~ Girth, data=trees)
summary(reg)$coefficients
```

```
##               Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) -36.943459   3.365145 -10.97827 7.621449e-12
## Girth         5.065856   0.247377  20.47829 8.644334e-19
```

Then exactly the same code, but now with the code suppressed. This done with the option `echo=FALSE` (not visible in the output). The easiest thing is to insert such options via the small wheel in the upper right corner of the R chunk.

```
##               Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) -36.943459   3.365145 -10.97827 7.621449e-12
## Girth         5.065856   0.247377  20.47829 8.644334e-19
```

Examples

Example 1: `table1`

The html output format plays well together with certain facilities for table generation. There is a function called `table1` which easily generates a table of statistics for variables of a dataset, possibly stratified after other variables in the dataset. The `table1` function is in a package with the same name.

We first (install and) load the package and import the **downloads** data. (I inserted an option such that

we don't get messages about loading of packages). Note: You may need to adjust the path for the file downloads.xlsx to import the dataset. You can do this manually or via the "Files" tab in the bottom right window of Rstudio.

```
# install.packages("table1")
library(table1)
library(readxl)
downloads <- read_excel("downloads.xlsx")
```

We then make an unstratified table (first) and a table stratified by machine name (second):

```
# Unstratified table
table1(~ size + time, data=downloads)
```

```
## Get nicer `table1` LaTeX output by simply installing the `kableExtra` package
```

	Overall
	(N=147035)
size	
Mean (SD)	4150 (88900)
Median [Min, Max]	0 [0, 14500000]
time	
Mean (SD)	0.954 (14.2)
Median [Min, Max]	0 [0, 1880]

```
# Stratified table
table1(~ size + time | machineName, data=downloads)
```

```
## Get nicer `table1` LaTeX output by simply installing the `kableExtra` package
```

	cs18	kermit	piglet	pluto	tweetie	Overall
	(N=16822)	(N=39157)	(N=41307)	(N=18396)	(N=31353)	(N=147035)
size						
Mean (SD)	5980 (100000)	4470 (103000)	3830 (98300)	3950 (77400)	3330 (46000)	4150 (88900)
Median [Min, Max]	0 [0, 6360000]	0 [0, 14500000]	0 [0, 14200000]	0 [0, 8670000]	0 [0, 4660000]	0 [0, 14500000]
time						
Mean (SD)	1.21 (26.8)	0.957 (13.0)	0.823 (8.51)	1.26 (17.1)	0.804 (9.20)	0.954 (14.2)
Median [Min, Max]	0 [0, 1750]	0 [0, 1380]	0 [0, 597]	0 [0, 1880]	0 [0, 1210]	0 [0, 1880]

Notice that the table1 function generates html-output only, and it cannot be used with MS Word or pdf output. Therefore, look at the corresponding html file to see the content.

Example 2: `kable`

Another option is the `knitr::kable` function that is compatible with html, pdf and latex. But it doesn't do everything for you. However, you can use tidyverse functions you will learn to make your own summary table.

```
# Unstratified table
downloads2 = downloads %>%
  summarize_at(c('size','time'), list(avg=mean,stdev=sd,median=median))
knitr::kable(downloads2)
```

size_avg	time_avg	size_stdev	time_stdev	size_median	time_median
4153.848	0.9539674	88941.97	14.22557	0	0

```
# Stratified table
downloads2 = downloads %>%
  group_by(machineName) %>%
  summarize_at(c('size','time'), list(avg=mean,stdev=sd,median=median)) %>%
  mutate(n = count(downloads, machineName) %>% pull(n))
knitr::kable(downloads2)
```

machineName	size_avg	time_avg	size_stdev	time_stdev	size_median	time_median	n
cs18	5979.865	1.2120939	100077.82	26.798180	0	0	16822
kermit	4470.019	0.9571091	103438.34	13.029852	0	0	39157
piglet	3828.645	0.8234826	98268.44	8.505271	0	0	41307
pluto	3946.811	1.2599522	77423.39	17.099850	0	0	18396
tweetie	3329.180	0.8039281	46049.85	9.204145	0	0	31353