

PRACTICAL 2:

Descriptive Statistics (Chapter 3)

In this practical you will become familiar with some more functions of R. You will use the data that you also used during last week's practical 1 to perform some first analyses involving descriptive statistics.

Last week you submitted answers to Practical 1. If you go to Nestor, you can see under Practical 1 that there is a grade (0-3) and we have also added the example answer file with grading details in PDF under R Practicals on Nestor. If you received 3 points, you can immediately move on to Part A of practical 2 below. In case you received 1 or 2 points, please take a few minutes to compare your file to the answer file. If you forgot to change some things in the data file, e.g. if you have not turned Participant into a factor, please make sure to correct those mistakes before continuing to work on that dataset (your code for that can be added to your answers for Practical 2).

Please note that we want you to report on your answers! We know that all of you are very well capable of copying code, but we also want to make sure you know and understand what you have done. So, from now on we want you to add one or two sentences to report on the results you obtained.

Part A

1. CREATE A MARKDOWN FILE and OPEN THE DATA IN R

- a. Start a new Markdown file, name it Prac2[yourinitials].Rmd, and save it in the folder where you want to work from (e.g. the 'R Practicals' folder we used last week). Remember that you need to save your Markdown in the folder where you also save(d) the data you will be using.
- b. Delete unnecessary text and add an informative heading for the sections and questions using #-signs (e.g., '##Part A' followed by '###1. Open data', '####1a' and so on to return an output similar to the one in Figure 2 below).
- c. Open the RDS file we saved last week by using the following code:

```
> Practical1 <- readRDS(file="Practical1.rds")
```

As was also mentioned in Practical 1, please remember that you want both code and answers to be added to the .Rmd file, so you will add everything in your .Rmd file (and not in the Console window!). In order to execute your line of code, you have to select it and run it by either pressing Ctrl+R, by clicking the run button (🏃), or by pressing the run current chunk button next to your line of code (▶) (also see Figure 1). They will all provide the same results, i.e. the answer to your code, so choose the one you find most convenient.

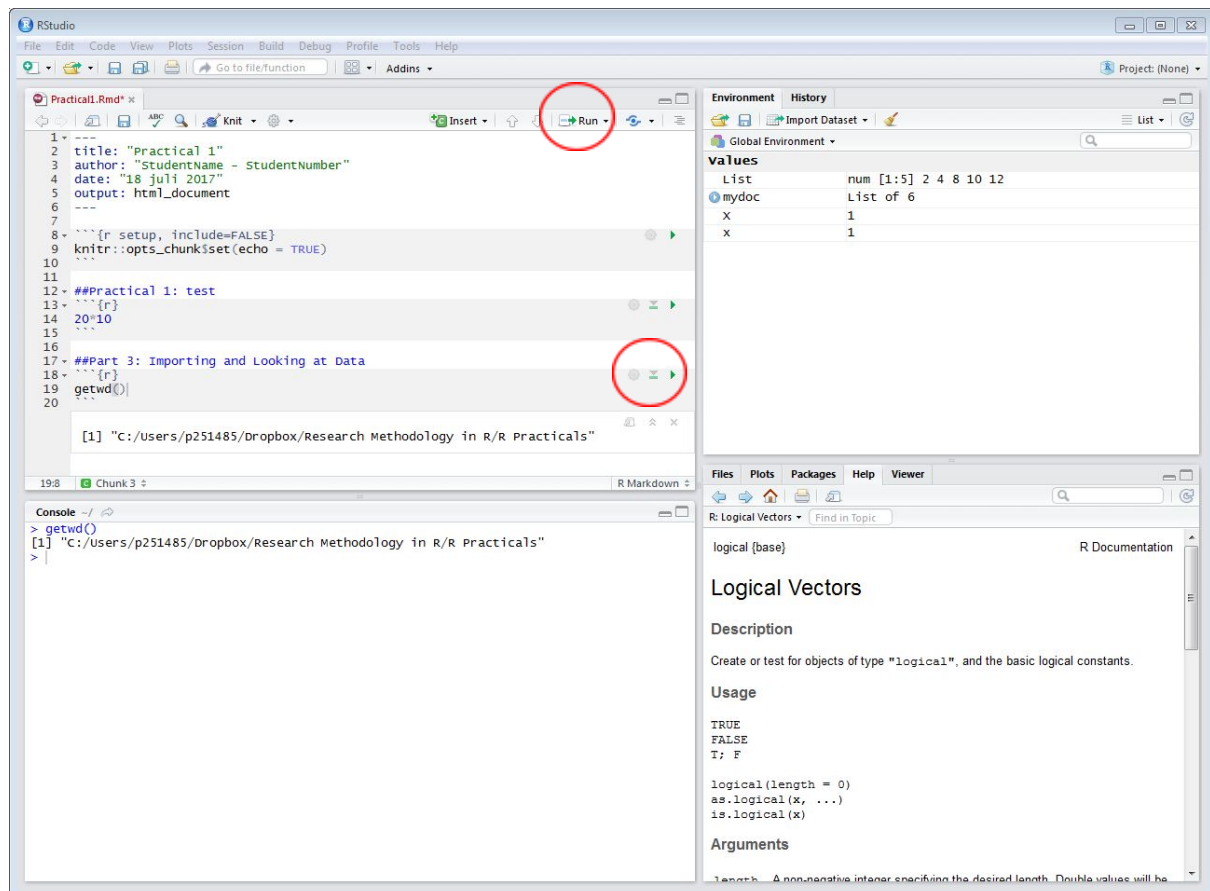






Figure 6. R Markdown script can be executed in various ways. You can select the code and press Ctrl+R, you can click the run button (), or you can click on the 'run current chunk button' next to your line of code ().

d. Finally, remember to save the .Rmd file () and to press 'knit' () to create a PDF or HTML file. It should resemble the picture below.

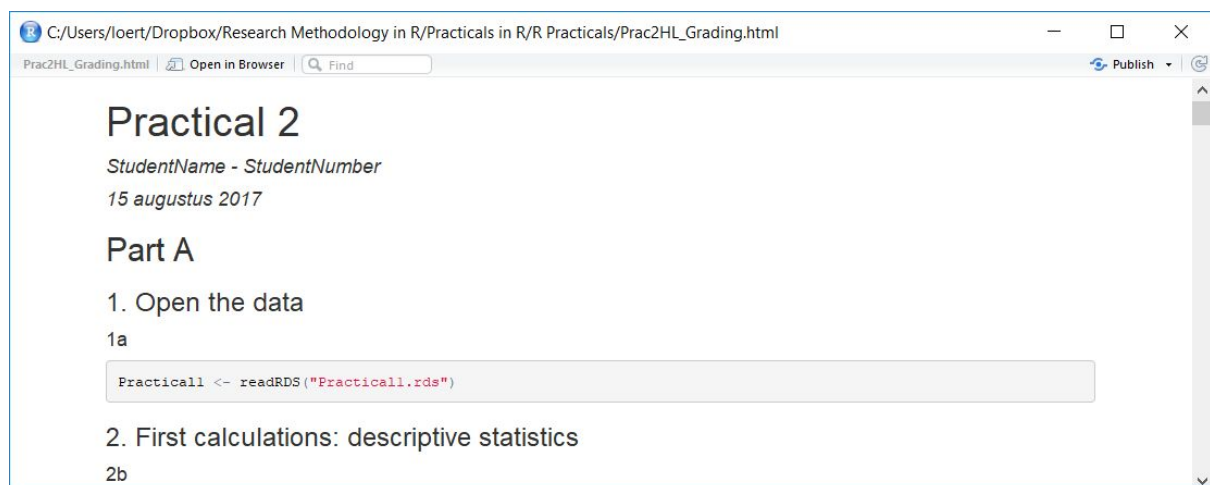


Figure 7. Markdown output practical 2.

2. FIRST CALCULATIONS: DESCRIPTIVE STATISTICS

a. First, add an informative heading.

b. During this first step, we want to find the frequency of occurrence of each age. This can be done creating a `table()` for the variable `age`. Do note that, R will need to know to which data set the variable `age` belongs and we do that using the dollar sign `$`. So, in this case our dataset is called 'Practical1' and it contains `participant`, `age`, `gender` and `profsc` as variables. By writing 'Practical1\$age', we are telling R that we want to work with the variable `age` from the file Practical 1.

So, use `table()` in combination with the above instructions and find out which age occurs most often. Do not forget to answer your question as text below your code in the Markdown file.

3. MORE DESCRIPTIVES: EXPLORING FREQUENCIES

a. Now we want to find the mean, the median, the mode, the range and the standard deviation of the Proficiency Score in your data. The following functions can be used to calculate these:

```
- mean(File$VariableName)
- median(File$VariableName)
- which.max(tabulate(File$VariableName))
- range(File$VariableName)
- sd(File$VariableName)
```

Unfortunately, there is no built-in function to calculate modes in R. Therefore, one option is to use a function that tabulates the scores and subsequently takes the maximum score. You do not have to understand this formula at this point, but it is important to realize that you will repeatedly come across problems like these. We were definitely not the first to find out that R does not have a built-in mode-function and there is no need to reinvent the wheel. Often, you can find the code you need through a quick (or sometimes long and disappointing) google search.

To make sure you understood what you have done, briefly report on your findings in the Markdown file. Please keep in mind that we use a dot (and not a comma) to report decimals (e.g. 0.5) and that you do not always have to report all decimals (use common sense or google!).

b. Also find and report the minimum age, the maximum age, the mean age and the standard deviation. Minimum and maximum can be calculated as follows:

```
- min(File$VariableName)
- max(File$VariableName)
```

OR using:

```
- range(File$VariableName)
```

c. What is the most frequently occurring proficiency score?

4. GETTING TO KNOW THE DATA : RELATIONSHIPS

a. Imagine that the researcher who gathered this data is interested in finding out whether the age at which someone started learning an L2 is related to the proficiency they obtained in that language. A scatter plot is normally a good place to start examining such a question. Use the formula below to create such a scatterplot in which Variable1 will be plotted along the x-axis (horizontal) and Variable2 along the y-axis (vertical).

```
- plot(File$Variable1, File$Variable2)
```

Your labels are now not really nice, but you can easily change that using `xlab` and `ylab` as in the following below. Additionally, this formula add a title above the graph (using `main`):

```
- plot(File$Variable1, File$Variable2, xlab="label x-axis", ylab="label y-axis",  
main="Title")
```

b. Type a caption below the scatterplot (e.g. Figure 1. Scatterplot showing the relationship between Age of Acquisition and Proficiency).

c. At face value, do you think there is a relationship between the two variables?

5. GETTING TO KNOW THE DATA: COMPARING GROUPS

a. Now we want to find out the mean proficiency scores and the standard deviations for the male and the female subjects. There are multiple ways on how to get to the answer to this question, but one easy-to-use function in R is the `aggregate()` function below.

```
- aggregate(Score~Factor, File, mean)
```

Using the above formula, you will get means for the Score (proficiency in this case) grouped by Factor (gender in this case) as the tilde (~) basically means 'depends on'. In Chapter 4 of Part I, we will discuss this dependency relationship in more depth. Try to fill in the above formula and report on your results.

b. Which group has higher proficiency scores, the male or the female participants?

c. Which group scored more homogeneously? You can use the formula above to answer this question, but you need to change the value that will be reported by the formula.

d. Now create a boxplot of the proficiency scores of the female and male participants separately. We again use a formula where the dependent variable depends on the independent variable using the tilde-operator (~). Try to adapt the formula below to create the boxplot you want and remember that you have to use exactly those names that are used in the Practical1 file for your variables. You can use `str(File)` or `names(File)` to check the column names in your file .

```
> boxplot(Variable on the y-axis ~ Variable on the x-axis , data=Your dataset, main="Title  
of your boxplot", xlab="Label for the x-axis", ylab=" Label for the y-axis ")
```

Don't forget to add a caption below the boxplot!

e. Compare your boxplot with the descriptive statistics. Can you find out how the boxplot is built up and what the different points of the boxplot signify? Explain this in your Markdown file (hint: there are 4 quartiles). You can also use Google to find out the answer.

Part B

1.

a. Enter the following 4 datasets in R, and provide the mean, the mode, the median, the range and the standard deviation. As there is very little data in these datasets, you can enter them manually using the `c` command we also used in Practical 1 to create our vector called List.

a. 3, 4, 5, 6, 7, 8, 9

b. 6, 6, 6, 6, 6, 6, 6

c. 4, 4, 4, 6, 7, 7, 10

d. 1, 1, 1, 4, 9, 12, 14

You can use the formulas we used before (e.g., `mean(File$VariableName)`), but you can also find packages that help you to report on several descriptive statistics for multiple variables at once (except for the mode again), such as the `describe(File$VariableName)` function from the “psych” package (Revelle, 2017). If you choose this latter option, remember to install the package once (this can be done through the Console) and use the `library(psych)` code to load the package. Only after loading it will you be able to use the functions from the package, so this latter code should be added to your Markdown file.

b. Do you agree with R’s calculation of the mode for variable ‘a’?

Part C

We will use a large sample of data containing information on the motivation to learn French and the score on a French Proficiency test. The other two columns reflect the amount of education and whether the participant has access to internet. The data can be downloaded from Nestor either as a ready-to-use R data file (Practical2C.RDS) with correct variable types and structures or as the original csv file (Data-Practical2C.CSV) in which variable types and labels still have to be added and altered. Please use the CSV-file in case you are a ReMa-student or if you are up for a challenge. In case you already had some difficulty, feel free to use the RDS file and continue with step 1b.

1. LOAD THE FILE INTO R

a. Save the CSV file in the folder you are currently working in and load it using the code we also used in Practical 1. Check the structure and the data itself and make sure all variables are in the correct format and that the labels below are added to the levels of the variables Motivation, Education, and Internet. Do remember that we are now dealing with a few ordinal variables and therefore we should use `ordered()` instead of `factor()` (see Practical1 for examples and details).

:

- Motivation: 1=very low; 2 =low; 3=neutral; 4=high; 5=very high;
- Education: 1=did not complete high school; 2=high school degree, 3=some college; 4=college degree; 5=post-undergraduate;
- Internet: 0=no; 1=yes; 8=does not know; 9=no answer.

b. Save the RDS file in the folder you are currently working in (if you have completed step 1a) and open it using the code we used before. Check the structure of the file (`str()`) and look at the data itself to get a feel of the data.

2. DESCRIPTIVE STATISTICS:

a. Find out the mean proficiency score, the median, mode and standard deviation for the group of students as a whole and then for the different motivation groups. In order to do this for the different motivation groups, we have to split up the file by Motivation and there are multiple ways to do that, but the simplest might be to use the aggregate formula we used before as well.

- `aggregate(Score~Factor, File, mean)`

Another option that we will use more in subsequent practicals (and that might be necessary here to find the modes) is to select data based on one level of a factor. You can, for example, select a HM (high motivation) group as follows:

```
> High <- Practical2C$Proficiency[Practical2C$Motivation == 'high']
```

By using the above formula, you are asking R to create a subset called 'High' that consists of the variable Proficiency from the file Practical2c, but only the scores for the data points for which motivation equals (==) 'high'.

Not sure what the names of the levels were? Check them as follows:

```
> levels(Practical2C$Motivation)
```

Once you have calculated all scores, it is important to also report them in an informative way also for us to be able to see that you understood what you have done. In your Markdown file, you can copy paste the following text with vertical bars to create a table. The only thing you have to do is replace the x's by the values you calculated:

```
value | Overall | very low | low | neutral | high | very high
- | - | - | - | - | - | -
mean | x | x | x | x | x | x
mode | x | x | x | x | x | x
median | x | x | x | x | x | x
sd | x | x | x | x | x | x
```

Now put an informative caption above the table. Note that for APA-style papers, captions for figures go below the figure, but captions for tables go above the table.

b. Make a boxplot with the different Motivation groups. Judging from the boxplot, do you think the groups will differ from one another? Report on your findings.

3. CHECKING THE NORMAL DISTRIBUTION:

a. To check whether proficiency has a normal distribution, create a histogram using the following code (where `prob=TRUE` makes sure to plot probabilities that together sum up to 1 instead of the raw frequencies that together sum up to the total number of occurrences in the data set):

```
> hist(FileName$Variable, prob=TRUE, xlab="Label for x-axis")
```

b. However, we also want the values for skewness and kurtosis to determine whether proficiency is normally distributed. We can use the "psych" package and its `describe()` function to check those values. If you have not used this package before to calculate descriptives, please use `install.packages("psych")` in the Console once to install the package. Remember to load the package in your Markdown file using `library()` if you have not already done so. Do these values deviate from zero and, if so, are they positive or negative? What does that tell you about the Skewness and Kurtosis (also try to link these values to the shape of the histogram you just created)?

c. Add a distribution curve to your histogram by adding the following line directly beneath your histogram function:

```
> curve(dnorm(x, mean=mean(FileName$VariableName), sd=sd(FileName$VariableName)), add=TRUE)
```

By looking at the histogram, does the data seem to follow a normal distribution?

d. Usually, you want to know whether your different groups are normally distributed by themselves. Carry out step 3a – 3c for the different motivation groups. HINT: use the groups we created for motivation under 2a of Part C.

4. PLOTTING THE DATA:

Statistics can be useful to get answers to specific questions, but you need to know your data before you can interpret your statistics. To get a good picture of the data let's create a grouped boxplot that displays education on the x-axis, proficiency on the y-axis and that is grouped by motivation.

a. We need the package *ggplot2* in order to create the boxplot, so install the package in the console and load it in your Markdown file like we did with the *psych* package.

Now fill in the correct names and use the following function following the explanation above on where to put each variable (you do not have to fully understand it, but do give it a try!):

```
> ggplot(data = Datafile, aes(x = VariableOnXAxis, y = VariableOnYAxis, fill=
GroupingVariable)) + geom_boxplot(aes(fill = GroupingVariable), width = 1) + theme_bw()
```

For a first try this does look nice! However, this plot is not yet exactly what we want. First of all, let's alter the labels by adding the following directly after the formula for the graph:

```
> + ggtitle("Title") + labs(x="NameXAxis",y="NameYAxis", fill="NameGroupingVariable")
```

Next we want to alter the text at the x-axis because it is not readable now. This is done by replacing `+ theme_bw()` by the following code (again, no need to fully understand this, but do try to link the changes you just made to the text in the formula):

```
> theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))
```

Of course you can change colours, angles and much more, try experimenting with this yourself! And do not forget to add an informative caption!

- When everything is finished and double checked, UPLOAD a PDF version of the HTML file as well as your Rmd. file to Nestor (in the assignment) and SUBMIT it.

Revelle, W. (2017). Psych: Procedures for Personality and Psychological Research, Northwestern University, Evanston, Illinois, USA, <https://CRAN.R-project.org/package=psych> Version = 1.7.5.