**Learn R as a language**

**Written by Aphalo, P.**

**Viivi Lindholm**

**Learning diary R**

CONTENTS

[1 R: The language and the program 4](#_Toc132584905)

[1.2 Computer programming 4](#_Toc132584906)

[1.3 R 4](#_Toc132584907)

[1.4 Reproducible data analysis 4](#_Toc132584908)

[1.5 Finding additional information 5](#_Toc132584909)

[2 The R language: “Words” and “sentences” 5](#_Toc132584910)

[2.2 Natural and computer languages 5](#_Toc132584911)

[2.3 Numeric values and arithmetic 5](#_Toc132584912)

[2.4 Logical values and Boolean algebra 6](#_Toc132584913)

[2.5 Comparison operators and operations 6](#_Toc132584914)

[2.6 Sets and set operations 6](#_Toc132584915)

[2.7 Character values 7](#_Toc132584916)

[2.8 The ‘mode’ and ‘class’ of objects 7](#_Toc132584917)

[2.9 ‘Type’ conversions 7](#_Toc132584918)

[2.10 Vector manipulation 7](#_Toc132584919)

[2.11 Matrices and multidimensional arrays 7](#_Toc132584920)

[2.12 Factors 7](#_Toc132584921)

[2.13 Lists 7](#_Toc132584922)

[2.14 Data frames 8](#_Toc132584923)

[2.15 Attributes of R objects 8](#_Toc132584924)

[2.16 Saving and loading data 8](#_Toc132584925)

[2.17 Looking at data 8](#_Toc132584926)

[2.18 Plotting 8](#_Toc132584927)

[3 The R language: “Paragraphs” and “essays” 9](#_Toc132584928)

[3.2 Writing scripts 9](#_Toc132584929)

[3.3 Control of execution flow 9](#_Toc132584930)

[3.4 Apply functions 10](#_Toc132584931)

[3.5 Functions that replace loops 10](#_Toc132584932)

[3.6 Object names and character strings 10](#_Toc132584933)

[3.7 The multiple faces of loops 10](#_Toc132584934)

[4 The R language: Statistics 10](#_Toc132584935)

[4.2 Statistical summaries 10](#_Toc132584936)

[4.3 Distributions 11](#_Toc132584937)

[4.4 “Random” sampling 11](#_Toc132584938)

[4.6 Model fitting in R 11](#_Toc132584939)

[4.7 Fitting linear models 11](#_Toc132584940)

[4.8 Generalized linear models 11](#_Toc132584941)

[4.9 Non-linear regression 11](#_Toc132584942)

[4.10 Model formulas 11](#_Toc132584943)

[4.11 Time series 12](#_Toc132584944)

[4.12 Multivariate statistics 12](#_Toc132584945)

[5 The R language: Adding new “words” 12](#_Toc132584946)

[5.2 Packages 12](#_Toc132584947)

[5.3 Defining functions and operators 12](#_Toc132584948)

[5.4 Objects, classes, and methods 12](#_Toc132584949)

[5.5 Scope of names 13](#_Toc132584950)

[6 New grammars of data 13](#_Toc132584951)

[6.4 Replacements for data.frame 13](#_Toc132584952)

[6.5 Data pipes 13](#_Toc132584953)

[6.6 Reshaping with ‘tidyr’ 13](#_Toc132584954)

[6.7 Data manipulation with ‘dplyr’ 13](#_Toc132584955)

[7 Grammar of graphics 14](#_Toc132584956)

[7.3 Introduction to the grammar of graphics 14](#_Toc132584957)

[7.4 Geometries 14](#_Toc132584958)

[7.5 Statistics 14](#_Toc132584959)

[7.6 Flipped plot layers 14](#_Toc132584960)

[7.7 Facets 14](#_Toc132584961)

[7.8 Scales 14](#_Toc132584962)

[7.9 Adding annotations 15](#_Toc132584963)

[7.10 Coordinates and circular plots 15](#_Toc132584964)

[7.11 Themes 15](#_Toc132584965)

[7.12 Composing plots 15](#_Toc132584966)

[7.13 Using plotmath expressions 15](#_Toc132584967)

[7.14 Creating complex data displays 15](#_Toc132584968)

[7.15 Creating sets of plots 15](#_Toc132584969)

[7.16 Generating output files 16](#_Toc132584970)

[8 Data import and export 16](#_Toc132584971)

[8.4 File names and operations 16](#_Toc132584972)

[8.5 Opening and closing file connections 16](#_Toc132584973)

[8.6 Plain-text files 16](#_Toc132584974)

[8.7 XML and HTML files AND 8.8 GPX files 16](#_Toc132584975)

[8.9 Worksheets 16](#_Toc132584976)

[8.10 Statistical software 16](#_Toc132584977)

[8.11 NetCDF files 17](#_Toc132584978)

[8.12 Remotely located data AND 17](#_Toc132584979)

[8.13 Data acquisition from physical devices AND 17](#_Toc132584980)

[8.14 Databases 17](#_Toc132584981)

[MY OWN THOUGHTS ABOUT THE BOOK 18](#_Toc132584982)

# 1 R: The language and the program

## 1.2 Computer programming

Computer programming and problem solving has two main steps, first one being forming the concept and design and the second being translation to a language. Abstraction is in a main role in all programming, no matter which program is used.

## 1.3 R

R can be considered as a program but also as a language as it behaves as one in every way, although there is no standardized grammar. It is clear and simple and can be used in many devices. Only the capacity of the device is defining the usage and size of the dataset.

The language R uses is very similar to earlier program called S. R has many advantages, like being free and open-source, and constantly developed by multiple users. Especially these extensions are the key element why R is so adaptable. All the codes can be easily repeated by any other human or machine.

R do not have any separate menu where user could choose what to do, but instead the user must type the command on console. R is also commonly used with some other program, like RStudio, to make the use easier.

Textual, language-kind of approach might feel more difficult in the beginning, but after learning the language it gives almost endless opportunities to handle the data, unlike menu-driven programs which have very limited options.

RStudio is a very convenient way to use R, as it has integrated warning system for errors, help function for finding excess information about the certain function of coding, and many other function which makes it easy to use. Also the user interface if more or less similar, no matter the current operating system or even if used via web browser.

## 1.4 Reproducible data analysis

Being able to reproduce the analysis when needed is crucial element in research. In RStudio, one gets a nice visible workflow which is easy to follow and later get back into. This is a feature that is not achieved in graphical user interphase.

In R, one does not get any separate file which includes only results, but rather it makes a script. By running the script code using the same data one gets both the steps of the workflow and the same original results extracted.

## 1.5 Finding additional information

There are multiple ways for searching information and help while using R. RStudio has built-in help pages for additional information and instructions, and they have some examples as well. The same information is often available as a separate file (PDF or other). A new thing for me is the citation command, that makes the automatic reference information when using some packages or R itself!

Internet forums are very useful source of answers, although sometimes finding the correct piece of information requires some extra effort. R has many users around the globe, and most errors and problems have happened at least once before ones own. Therefore, there is a very high probability of finding applicable information and answers to almost any question one can imagine.

Example: I had the need of dividing my dataset into subsets in a certain manner. First I searched information from online (StackOverflow and other forums) and got some commands I could use. Then I searched for additional information about the usage of suggested codes/commands by using the ?-command in RStudio. This gave me enough information to keep going with the data processing. I have never commented or asked for help myself, but rather search if someone else have had problems and answers that can be applied in my case.

# 2 The R language: “Words” and “sentences”

## 2.2 Natural and computer languages

R language has similarities with both natural languages and mathematical language. Unlike natural languages, R is not very forgiving what comes to slightest errors. It will not let you continue without making corrections, and therefore the change of misunderstanding is not present. R also follows operator precedence rules, the typical order of mathematical calculations. This is what one should take into account while writing commands in R.

## 2.3 Numeric values and arithmetic

Mathematical functions are the main elements of using R, and many similar rules apply as in mathematic. The used language must be precise: small letters (for example d) are not red as equal as similar capital letters (D) and they contain a completely different meaning and value, because they are not presenting letters as letters, but rather symbols for vectors or random variables. Even there is some strict rules in the logic of R language, many things can be done in multiple ways.

The use of letter L as a symbol of integer number was also something new to me. I use R mostly for quite simple statistics, so this kind of interesting details have not reached my knowledge. The concept of floating numbers and integers/doubles etc. seems a bit abstract to me and is hard to understand. I read the chapter several times, but I still don’t quite get it. I’m not sure if it’s about the language or my non-math brain overall.

<- operator is very commonly used in R. Vectors are very fundamental parts of code, and with <- one can give a name to a sting, value, dataset, or calculation. Concept of vector and how to use them is very familiar to me. Append was a new command, and before I have always just used c()-command because I haven’t needed function to insert a vector inside another, but probably in the future there will be a chance to use it!

On the page 25 is an interesting paragraph about some common commands. Most of them I knew as I have used R for some years, but the use of pattern argument and ^ and $ signs as a searching tool was completely new thing to me!

## 2.4 Logical values and Boolean algebra

TRUE/FALSE are considered logical, or Boolean, values. This was hard for me to follow and I honestly don’t have slightest clue what’s going on in this chapter. I really need some concrete examples to understand this whole concept. I don’t know, or at least remember, what are Boolean numbers etc. and I don’t understand what the take-home-message of the examples is. I feel like there was something wrong with my understanding here, maybe I’m tired or this just is too hard for me.

Commands any() and all() were new to me, but rather simple to understand. Any() will search and give a value if even a single value in dataset is as the given value, but all() requires all values to be equal to given value to be true.

## 2.5 Comparison operators and operations

Comparison signs <, > and = are familiar from basic math, and in my opinion their use in R does not differ, no matter if logical values or arithmetic values are used. This chapter was easier to follow.

## 2.6 Sets and set operations

I found this chapter interesting, as these operations like %in%, setdiff() and intersect() are very useful when organizing datasets in real life. I have constant struggle with these basic operations, so this sheds some light on it. I found the concrete example of fruits and shopping useful and understandable.

## 2.7 Character values

Both quotation marks, “ and ’, can be used as delimiters of character constant. This allows

to use the other type when coding vectors that has quotation marks in it.

## 2.8 The ‘mode’ and ‘class’ of objects

I did not know that such a function as mode() exists, but I have missed it already so many times! I have always used functions starting with is. to find out what is the mode of my data!

## 2.9 ‘Type’ conversions

The function as. changes the type of a sting/data to required form. This function is very familiar to me, unlike the other two functions. I believe there are times and places one could use those, but probably not very often while doing very basic stuff as I do. The use of function sprint() required a moment to understand properly, but after a while it was all clear.

## 2.10 Vector manipulation

Vector manipulation is probably one of the most important tasks one needs to handle while using R, at least in my opinion. The hard part is to memorize which brackets, if any, are used in which cases to dig out the information required. The playground tasks with selector and indexes explained well how vector manipulation helps to extract information.

## 2.11 Matrices and multidimensional arrays

Handling matrices and arrays is quite similar to vectors. Function t() that transposes the matrix is new to me indeed and comes in handy! All the other functions seemed more or less familiar to me. Only the very last part with matrix multiplications was a little unclear.

## 2.12 Factors

How can it be that I have never heard about this function before! I feel like this should be some very basic stuff when teaching how to handle the data in R, and with this function I would have avoided some extra effort while doing my tasks. Paragraph was easy to follow.

## 2.13 Lists

I’m not sure in which context there are used or why they are useful? This part was not easy to understand and I’m not sure if I got the full message, even I tried to read it several times. Maybe it would be good to give some concrete examples in which context these functions are useful? This doesn’t apply only for this particular case but also the others.

For people like me who find abstract numbers and math in general challenging benefit from real-life examples. Some parts of the book are easier to understand because I have some background and context to them, and I can think how I could have used them in my own experiments etc. This is something that has been challenging for me for years, and I guess some people just can handle abstract things with more ease than others.

## 2.14 Data frames

I don’t know why that list thing feels somehow unclear but data frame which literally is kind of a list seems all clear. Maybe I was tired while reading the previous chapter or something, but the concept of data frame is much more familiar with me and I have used and handled them many times. Everything in this chapter was more or less easy to understand, and there were no big surprises.

There were many commands introduced in this chapter, but I picked only few here. I normally prefer using $ instead of double square brackets as it seems simpler to me. I also rarely use the command subset() but rather make “a subset” by using []. I haven’t heard about commands with() or within() before.

## 2.15 Attributes of R objects

I guess this is also kind of a “nice to know” thing, even though I don’t find it very useful in my own work. I didn’t know that one can store or erase attributes like this, but it’s hard to imagine any situation where I would use it myself.

## 2.16 Saving and loading data

The information of this whole chapter was somewhat familiar to me, yet many of these things I forget easily so it was good to read again. It is essential to know how to load and save datasets, which is why I think this section should be maybe even earlier in the book.

## 2.17 Looking at data

Again, this was very trivial in basic use of R but it’s always a good idea to refresh memories by reading things again.

## 2.18 Plotting

Plotting itself is not anything new, but as a RStudio user I didn’t know that I could use commands like pdf() etc. to produce plots to certain formats! I usually use the plotting device in RStudio to observe the plot, and then I export it there when needed.

# 3 The R language: “Paragraphs” and “essays”

## 3.2 Writing scripts

When I first started using R some years ago, I had a tendency to write whatever came to my mind, test things and re-write something randomly, making my code very hard to read and almost impossible to get the same results as I had had before. Nowadays I have much more systematic way to produce code, I make script “cleanups” regularly and also use # to explain (to myself and possible to others reading the code) what I have done and what my intentions have been to make sure that my analysis can be executed again just like I have done, and I can use the same code to other things if I need to.

## 3.3 Control of execution flow

It took several attempts to read the beginning of this chapter before I even understood what’s going on here, but when I got to *if-else* part I finally got what the whole chapter was about. Sometimes it sure is because of the language that I’m not speaking as a native (and especially this kind of special vocabulary) but also because I couldn’t get any context in first few paragraphs. I have used *if-else* commands many times. I also understand what switch() does but the concrete use of that command still remains a mystery. What kind of dataset it’s good for?

Loops are difficult to many people – me definitely not being an exception to this – and it would be very important to explain every single character and step of the code to make sure that the reader understands. At first I thought I was following but then all the sudden it changed with some chunk of code (p. 114) that I have no idea what is going on, and nowhere is explained for what the letter *i* or text *along with* or other things stand for. Yes, I can google it (and I did) but I think it’s not very handy that the reader has to use external source to understand the text.

Similar problems occur every time I try to find help with R or similar subjects: majority of people trying to help/teach with programming and statistics are just too deep into it and naturally understand mathematics etc. that they don’t understand or remember anymore how little beginners actually know, and how differently people handle this kind of abstract information. Same often applies to help pages of R commands: they have been written by someone who can’t see why someone don’t think as they do. For me most of the R help pages are gibberish and it’s really frustrating when I’m trying to solve the problem or learn why certain things happen. They should explain things like they would explain to their grandmother. It’s better to rather oversimplify things than assume others to have similar way of thinking and understanding.

## 3.4 Apply functions

I have seen these many times but don’t remember using them by myself. Some things were easy to understand but some still remain unclear. At least I learned the basic principles of these functions.

## 3.5 Functions that replace loops

Most of these functions are familiar to me. I avoid using loops because I still quite don’t understand how to make them, so many of these functions are handy.

## 3.6 Object names and character strings

Once again I wish there was some real life examples where one could use assign() command. Is this something beyond normal R use or something everyone making basic statistics/analysis should know? I found doing the playground task difficult. Some examples were told but I still can’t think any case where this would be necessary or any way useful for me.

## 3.7 The multiple faces of loops

“More advanced aspects of the R language” for sure are over and beyond my capacity as already the easier parts of this part of the book (I mean the chapter 3 and all the parts of it) felt too difficult to handle. Therefore I read this part rather quickly and didn’t put much effort to this specific section.

I wonder why this chapter 3 in total was so difficult to me. I tried my best to understand and don’t consider myself any more or less incapable of learning in general than anyone else. I also avoided reading while I was tired, but it didn’t help. The previous part (chapter 2) felt all okay compared to this part of the book. I guess it might be something related to how we handle and understand information and if we already have some real-life examples where we can compare the new information.

# 4 The R language: Statistics

## 4.2 Statistical summaries

This part was very familiar to me and I have used all the commands mentioned in this chapter. These are some commands I use almost every time I do any kind of analysis.

## 4.3 Distributions

I didn’t know that the random numbers pulled by computer are actually not random but pseudo random. Otherwise use of pnorm(), rnorm() and qnorm() are familiar from earlier statistics courses.

## 4.4 “Random” sampling

This is also something I have learned from previous courses. I haven’t used it in any real-life data-analyses, but in the course we used this a lot.

**4.5 Correlation**

Cor() and cor.test() are some commands I have used several times when doing any sort of data-analysis. I have also used several methods, but mostly Pearson’s.

## 4.6 Model fitting in R

I have regularly used and will use all of the mentioned commands. I guess linear model is the most common one when doing some rather simple analysis like I do. I still have a lot to learn about fitting models etc. so I’m looking forward what kind of things are mentioned in this book.

## 4.7 Fitting linear models

This chapter was good for me and deepened my understanding about fitting linear models. I knew already most of the things, but there were new pieces of information here and there. ANCOVA was something I have never heard of before.

## 4.8 Generalized linear models

Also generalized linear models are very familiar to me, as it’s been taught in some previous courses and I have used it myself too.

## 4.9 Non-linear regression

I have heard about non-linear models but I guess I have never use it myself. It’s good to know the basics but I think I’ll need to read a bit more if I’m about to use this kind of regressions someday. I must admit that I didn’t go very deep into this specific chapter.

## 4.10 Model formulas

First I wasn’t even sure what was the text about, but then after a while and seeing more examples I got it and it’s familiar from some statistics courses and my own analyses.

I haven’t used so much nesting of factors in formulas as I have mostly stuck with rather simple setups, but I have done at least some analysis with the help of 4lme when working on with my master’s thesis data. This is anyway something I need to deepen my knowledge more to properly understand it. I still think most of the things were a bit too high for my level.

## 4.11 Time series

Time series is something I have done before in one task some years back! I had time series data *with* replicates and it took me quite a while to understand how to handle and analyze such data. I haven’t tried the ts() command though. Stl() command looked interesting!

## 4.12 Multivariate statistics

I’ve done PCA’s, MDS’s and cluster analyses several times and it’s always somehow exciting to do it to a new data set! I find plotting those a very informative way to observe data and see the relationships of the variables. I have mostly used it to plant survey data, but probably will try soon with bird data I collected from Laos. Can’t wait to see how it looks!

The MDS example was very informative and I found it clever to use the actual distance of the measurement points to show how the data behaves. Often the distance between observations is not something you could measure with a ruler, but this time it is and it’s easier to understand.

Even I have done these it’s sometimes still not easy to understand how the analyses work and what to do if they give some errors. Also some of the commands used in the examples were bit different to those I have used.

# 5 The R language: Adding new “words”

## 5.2 Packages

I think in practice in most cases when I try to analyze something I end up checking methods by googling and just install any packages that are recommended and seem sensible to execute the task. It’s anyway fun to know a bit more of the background of the packages.

## 5.3 Defining functions and operators

Parts of this chapter were a bit advanced compared to things I do, but some parts were familiar and useful. I have made some functions but I have never tried to do one with the na.omit kind of argument. It’s nice to see step by step how more complex operations are constructed.

## 5.4 Objects, classes, and methods

I really don’t know any occasion where I would need to change the class of an object. It sure might be good to know that it’s possible to do but I’ll leave it to that level.

## 5.5 Scope of names

I have sometimes faced some problems when two packages are sharing same names for commands. It’s rather rare situation but might happen when many packages are used for working on with a project.

# 6 New grammars of data

## 6.4 Replacements for data.frame

Once again, I’m not sure how useful this information is for a very basic user who mostly does simple analyses etc. I understand that those commands introduced in this chapter allow handling very large chunks of data more easily and maybe some special properties that data.frame cannot provide, but maybe they are somewhat more advanced stuff compared to my use of R.

## 6.5 Data pipes

I have heard about the concept of pipe in this context, but I had no other previous knowledge about that. Now when I saw the examples I remember I have seen similar structures (especially %>% and dotpipe) in Stack Overflow and similar websites when I have search for a solution to problems I have faced. I also understand now the basic principle of the use of pipes.

## 6.6 Reshaping with ‘tidyr’

I have been sometimes in situation where I have to “transpose” some wide form data to long one and to be fair it’s horrible, because back then I didn’t know the methods to do it efficiently.

## 6.7 Data manipulation with ‘dplyr’

What is happening in this mutate() command here?

…str\_extract(part, "^[:alpha:]\*"), part\_dim = str\_extract(part, "[:alpha:]\*$"))…

I didn’t find any explanation for that alpha thing. Maybe I have missed something in previous chapters but would be quite useful anyway to tell step by step what is done in the code and what each section/word stands for. Otherwise things were pretty clear in this chapter.

I have also found the use of aggregate() relatively smooth and easy but I guess everyone got their preferences. The nice thing about R is that it allows you to do the same thing in several ways.

I don’t remember using joining functions in R but in QGIS several times. I think it’s nice to know also how to do it in R as probably I’ll need to use it at some point.

# 7 Grammar of graphics

## 7.3 Introduction to the grammar of graphics

Finally a step-by-step guide to use ggplot! Many times I have tried to search some help to plot my data but often the plotting instructions have been rather complex with no explanation what happens in each step. As I didn’t have any base knowledge, it has been very difficult to get help from those instructions from help pages. Now it’s much easier! This is very useful information for me and definitely will benefit from it.

## 7.4 Geometries

I found most of the introduced features very useful, although there were some that might be higher stuff than the thing I do. Nevertheless, it’s good to know these, and the functions were nicely presented and easy to follow. This is exactly what and how I need to learn.

## 7.5 Statistics

It was nice to see more types of plots and opportunities ggplot has. I think chapter was easy to follow and the examples were good and clear, and even I haven’t used most of the plot types myself I have at least seen them in different contexts before. It’s also good that book offers at least some pictures to show for the reader, because this time I was also reading from my phone and had no opportunity to try to make those plots. Sometimes reader might also face some different issues with R or computer, which is why it’s good to have pictures included in the book.

## 7.6 Flipped plot layers

I might find use for these in the future, but maybe at this point it was just nice to know that the plots can be flipped and get the general idea how to do it.

## 7.7 Facets

Facet as a term was new to me in this context. I saw that they were already used in previous examples, but I didn’t know how they were used exactly.

## 7.8 Scales

Pretty much similar thoughts as with some previous chapters: nice to get deeper into this and see all the opportunities available. Also following up the text was mostly easy and the examples provided were good.

## 7.9 Adding annotations

The basic idea of adding annotations seems simple, although I rarely have seen them. Or maybe I just don’t know yet which ones are actually annotations and which are not?

## 7.10 Coordinates and circular plots

It’s funny how often pie charts are used in common media even though they are difficult to properly read and understand. Could it be even the purpose in some cases?

Anyway, circular plots are not very useful in most cases but it’s of course very important to know that it’s possible to make them as sometimes they still might be the best way to present the information.

## 7.11 Themes

It was quite fun to see some themes that are available for plotting! In scientific publications plots look usually rather boring but they serve their purpose, but for example in different reports or other similar occasions it’s always nice to get some color and structure to the plots.

## 7.12 Composing plots

This is definitely something I would have needed in my previous master’s thesis. I had several individual plots which I needed to put next to each other as a group in a page. It was horrible to try to assemble them so that they looked uniform in plot and font size and using PowerPoint didn’t feel (and for sure was not) very sophisticated method to do that. At least now I know better, as the ggplot method looked very simple.

## 7.13 Using plotmath expressions

This might need a few tries to get some successful results if I tried to do these by myself from the memory. Probably I’ll come back to instructions in many other cases too, but at least this felt somewhat more difficult.

## 7.14 Creating complex data displays

I must admit that the previous plot and exercises have been indeed complex enough for me. I’m anyway satisfied that after all I’ve been able follow the instructions so far and have done the plots successfully.

## 7.15 Creating sets of plots

This was a good chapter with especially useful information! It’s good to know different methods how to save and be able to later use the data, although I probably stick with some of the simplest methods at least in the beginning.

## 7.16 Generating output files

Making output files instead of always relying on RStudio’s export menu is definitely a good practice to learn in the future. I’m lazy and use export menu almost every time, but would be better to learn new routines as they are very easy to use as well.

# 8 Data import and export

## 8.4 File names and operations

Most of the time I use setwd() and getwd() commands when importing data. I think I haven’t even tried the other ones introduced here, or if I have, I don’t remember that. It’s easy to stick something you know and what have been working well, but maybe if I have some different and/or bigger data sets in future the other ones might be handier.

## 8.5 Opening and closing file connections

This was a totally new thig to me, and I didn’t quite understand the benefit of this method.

## 8.6 Plain-text files

I have used read.csv() and write.csv() many times, but others were not very familiar. At least in some course I have tried the read\_delim() also. I often use the menu in RStudio which shows also the preview of the file imported, so one can easily see and select which separator has been used.

## 8.7 XML and HTML files AND 8.8 GPX files

Well, I think once again it’s good to know that this is possible, but I’m not very sure how and why to use these commands for my own purposes as I’m not planning to do any super advanced coding in my near future.

## 8.9 Worksheets

Most of the time I have used commands from the readxl package, mostly just read\_excel, but sometimes also xlsx package. I have also converted the data first to csv-file and then opened in R. The package readODs was new to me.

## 8.10 Statistical software

I have never tried to import data from other statistical programs to R, so it was quite interesting to see how it’s done. I haven’t even used many other programs before R (or Excel of course). SPSS is familiar to me in some extent because we were learning it in my previous degree in university of applied sciences, but it’s been many years since the last time I have tried it and to be honest I don’t miss using it. There is still some people using it so it’s good to know how to exchange data between different programs.

## 8.11 NetCDF files

I have never heard about this file type before, but it’s no wonder as it seems to be very specific to certain scientific fields. Nice to know, but probably not very usable on my own field so I read it rather quickly through.

## 8.12 Remotely located data AND

## 8.13 Data acquisition from physical devices AND

## 8.14 Databases

My use of R has been so simple and my data sets have been so small so I haven’t really had to think about these, but I can easily imagine in case of a very big data set that these are very useful or even essential pieces of information.

# MY OWN THOUGHTS ABOUT THE BOOK

I did it! I told you about my difficulties to read, and as I knew that the information provided by this book was quite “heavy” in sense that I’m not very skilled with R or statistics, I’m very pleased and surprised that I got through this *relatively* easily! I wrote basically just my own thoughts and experiences.

It was of course easiest to follow the chapters that had something familiar and/or something that I had special interest in. I liked chapter seven the most with plotting and visible examples. I’m very visual person and learning math has always been quite challenging to me because of the abstract nature of it. Plotting makes the numbers “visible” and I find it fun to adjust the plots and see how things look in a picture.

Then a few words about the book itself. I was sometimes a bit confused what is the main purpose and what is the audience of the book. I have a few years background using R in rather simple statistics (and as I said, I’m not especially skilled with this, even I have interest towards these things) and sometimes the examples in chapters felt quite easy, but many of them felt very difficult and much more advanced. Sometimes I felt really frustrated if I didn’t understand some new concepts or other thing. I would like to tell what kind of things would make it easier to understand, but I don’t know it myself so unfortunately I can’t give any constructive feedback about that. Anyway, it was very good that there were plenty of examples, plenty of code to play etc. I didn’t do every single exercise but most of them yes and they worked. Sometimes I struggled to find the answer or do the task, so it would be nice to provide answers/model examples in some form. Maybe there could be an “answer page” in R which would be available for the reader when needed?

A very good thing about the book was also a clear division into chapters that were small enough. It’s exhausting to read very long chapters and try to kind of wrap the information after reading. Shortish chapters and informative headers made it possible to get familiar with concepts by reading the headers first and then thinking what I knew about the topics already.

Overall I’m happy that I read this and I will also download the book for further use, as there were multiple good examples how to do different things with R, and many of them I will need in my future analyses as well!