*Back cover text*

Learning a computer language like R can be either frustrating, fun or boring. Having fun requires challenges that wake up the learner’s curiosity but also provide an emotional reward on overcoming them. The book is designed so that it includes smaller and bigger challenges, in what I call playgrounds, in the hope that all readers will enjoy their path to R fluency. Fluency in the use of a language is a skill that is acquired through practice and exploration. Although rarely mentioned separately, fluency in a computer programming language involves both writing and reading. The parallels between natural and computer languages are many but differences are also important. For students and professionals in the biological sciences, humanities and many applied fields, recognizing the parallels between R and natural languages should help them feel at home with R. The approach I use is similar to that of a travel guide, encouraging exploration and describing the available alternatives and how to reach them. The intention is to guide the reader through the R landscape of 2020 and beyond.

What will you find in this book?

* R as it is currently used.
* Few prescriptive rules, mostly the author’s preferences together with alternatives.
* Explanation of the R grammar emphasizing the “R way of doing things”.
* Tutoring for “programming in the small” using scripts.
* The grammar of graphics and the grammar of data described as grammars.
* Examples of data exchange between R and the foreign world using common file formats.
* Coaching for becoming an independent R user, capable writing original code and of solving future challenges.

What makes this book different to others?

* Tries to break the ice and help readers from all disciplines feel at home with R.
* It does not make assumptions about what the reader will use R for.
* It attempts to do only one thing well: guide readers into becoming fluent in the R language.

**Pedro J. Aphalo** is a PhD graduate from the University of Edinburgh, currently lecturer at the University of Helsinki. A plant biologist and agriculture scientist with a passion for data, electronics, computers and photography in addition to plants. A user of R for 25 years, who first organized an R course for MSc students 18 years ago and the author of 13 R packages currently in CRAN.

Chapter 1

R: the language and the program

In this chapter you will learn some facts about the history and design aims behind the R language, its implementation in the R program, and how it is used in actual practice when sitting at a computer. You will learn the difference between typing commands interactively, reading each partial response from R on the screen as you type versus using R scripts to execute a ``job'' which saves results for later inspection by the user. You will also learn advantages and disadvantages of textual command languages such as R compared to menu-driven user interfaces as frequently used in other statistics software and occasionally also with R. You will also learn about the role of textual languages in the very important question of reproducibility of data analyses.

Chapter 2:

The R language: ``words'' and ``sentences''

For those not familiar with computer programming, the best first step in learning the R language is to use it interactively by typing textual commands. This chapter teaches not only the syntax and grammar rules, but also gives you a glimpse at the advantages and flexibility of this approach to data analysis. In the first part of the chapter you will use R to do everyday calculations. This easy start will give you a chance to focus on learning how to issue textual commands at the command prompt. Later in the chapter, you will gradually focus more on the R language and its grammar and less on how commands are entered. By the end of the chapter you will be familiar with most of the kinds of ``words'' used in the R language and you will be able to write simple ``sentences'' in R.

Chapter 3:

The R language: ``paragraphs'' and ``essays''

As soon as a data analysis stops being trivial, describing the steps followed through a system of menus and dialogue boxes becomes extremely tedious. When the same sequence of commands needs to be applied to different data sets or when an earlier analysis needs to be reproduced, scripts make both implementation and validation of such requirements easy. In this chapter you will learn how to write and use R scripts, starting from an extremely simple script and progressing towards more complex ones. Related groups of R statements within a script are similar to “paragraphs” while a whole script describing the application of a data processing algorithm to data is similar to an “essay” in natural language.

Chapter 4:

The R language: Statistics

This chapter gives the reader a quick introduction to statistics in base R. Although, many of R’s functions are specific to given statistical procedures, they use a common approach to model specification and when returning the computed values. This approach can be considered a part of the R language. Here you will learn the approaches used in R for calculating statistical summaries, generating (pseudo-)random numbers, sampling, fitting models and carrying out tests of significance. We will use correlation, *t*-test, linear models, generalized linear models, non-linear models and some simple multivariate methods as examples. The aim is teaching how to specify models, contrasts and the data used, and how to access different components of the objects returned by the corresponding fit and summary functions.

Chapter 5:

The R language: adding new ``words''

In earlier chapters we have only used base R features. In this chapter you will learn how to expand the range of features available. In the first part of the chapter you will learn how to use existing packages such as those available for download from CRAN, the comprehensive R archive network, and how they be used to expand the functionality of R. In the second part you will learn how to define new functions, operators and classes. We will not consider the important, but more advanced question of packaging functions and classes into new R packages. Use of packages is equivalent to importing ready defined “words”, while defining new functions, operators and classes is similar to defining new “words” such verbs and nouns.

Chapter 6:

New grammars of data

Base R and the recommended extension packages (installed by default) include many functions for manipulating data. The R distribution supplies a complete set of functions and operators that allow all the usual data manipulation operations. These functions have stable and well described behaviour, so they should be preferred unless some of their limitations justify the use of alternatives defined in contributed packages. In the present chapter we aim at describing the new syntaxes introduced by the most popular of these contributed R extension packages aiming at changing (usually improving one aspect at the expense of another) in various ways how we can manipulate data in R. These independently developed packages extend the R language not only by adding new `words' to it but by supporting new ways of meaningfully connecting `words'­---i.e. providing new `grammars' for data manipulation.

Chapter 7:

Grammar of graphics

Three main data plotting systems are available to R users: base R, package ‘lattice’ and package ‘ggplot2’, the last one being the most recent and currently most popular system available in R for plotting data. In this chapter you will learn the concepts of the grammar of graphics, on which package ‘ggplot2’ is based. You will also learn how to build several types of the data plots with package ‘ggplot2’. Because of the popularity and flexibility of ‘ggplot2’, many contributed packages extending its functionality have been developed. The focus here is mainly on the grammar itself as defined in package ‘ggplot2’ and some extensions to this grammar provided by contributed packages including ‘ggpmisc’, developed by the author.

Chapter 8:

Data import and export

Base R and the recommended packages include several functions for importing and exporting data. Contributed packages provide both replacements for some of these functions and support for several additional file formats. In the present chapter you will learn how to exchange data between R and the outside world, covering in detail the most common ``foreign'' data formats. Developing software to read and write files using foreign formats can easily result in long, messy, and ugly R scripts. This chapter will guide you on how to find functions that are well tested or validated and teach you how to use them to read data encoded using foreign formats.