BOOK REVIEW: THE ART OF R PROGRAMMING

MATTEO PELAGATTI


The language R has become the statistician’s gold standard for data analysis and programming and it is gaining popularity also among econometricians. The main reasons for this popularity are to be found in its being open-source and extremely rich of contributed packages (3686 in the Comprehensive R Archive Network or CRAN).

When econometricians try to switch to R from languages such as Gauss, Ox, Matlab or Stata, they are usually upset by some peculiarities of R that give a first chaotic impression of the software. These peculiarities turn out to be good grounds to use R maybe in combination with one of the aforementioned languages. For this reason, if one decides to switch to R, my suggestion is that (s)he should support this process with a good book such as Matloff’s *The Art of R Programming*.

Norman Matloff is certainly in a good position for writing a book about a statistics-oriented programming language, indeed he is a professor of computer science and a former professor of statistics at the University of California, Davis, and his expertise in debugging and parallel computing made him cover these two topics often overlooked in other books on R programming.

As for the readership, we quote directly from the book’s Introduction:

this book is for those who wish to develop *software* in R. The programming skills of our intended readers may range anywhere from those of a professional software developer to “I took a programming
course in college,” but their key goal is to write R code for specific purposes. (Statistical knowledge will generally not be needed.)

The 350 pages of the main matter of the book are organized into 16 chapters. The first chapter (Getting Started) introduces the basics of R, gives a first look at the main R data structures and overviews the various ways to get help from the R environment and the Internet. Chapters 2 though 6 cover the main data types and structures, namely, vectors, matrices, arrays, lists, data frames, factors and tables. Chapter 7 (Programming Structures), the longest in the book, focuses on programming. It treats control statements, function writing, environments and scopes of functions and variables, and reveals many of the idiosyncracitities of the R language. Chapter 8 (Doing Math and Simulation in R) concentrates on probability functions, random number generators, linear algebra and set operations. Chapter 9 (Object-Oriented Programming) covers the two alternative ways of writing object-oriented code in R, namely the so-called S3 and S4 classes. The coverage of the S4 classes is somewhat undersized, but at the moment of writing the book the use of this type of classes is still rare in practice and the S3 paradigm pervades both the core and contributed packages. The input/output streams are treated in Chapter 10 (Input/Output), where, besides the classical keyboard, monitor and file streams, one learns also how to access the Internet. Chapter 11 is on string manipulation and Chapter 12 covers graphics in a basic way. Graphics is one thing in which R really excels and entire books are devoted to this topic (e.g. Wickham, 2009), so the basic presentation of Chapter 12 does not do justice to R. However I believe that the author chose the right compromise in presenting this topic in a book on programming. Chapter 13 is on debugging, a topic too often overlooked by texts on R. Chapters 14 through 16 are probably the most interesting to regular R-users. Chapter 14 gives many tips no how to optimize the code for speed and memory usage. Chapter 15 shows how to write C/C++ functions to be called from R and how to use R from Pyhton. Chapter 16 extends Chapter 14 by showing how to implement parallel computing in R; there is even a short section on GPU
programming. Two very short appendices entitled *Installing R* and *Installing and Using Packages* conclude the book.

The writing style of the book is crisp and the choice of alternating explanations and examples makes the reading pleasant and never boring. Chapters are organized in many short sections and subsections and this simplifies the use of the book as a reference.

The 150 pages dedicated to the main R data structures may seem an overload to Gauss, Ox, Matlab or Stata programmers, but the initial chaotic impression that new R-users face is generally due to this rich availability of data structures not shared by most other econometric programming languages.

The only relevant flaw in the book is the lack of a chapter on how to write R packages. Indeed, the abundant availability of packages is one of the main reasons for many statisticians and econometricians to use R but, once they refine their programming skills, they usually want to contribute to the community with their own packages.

Some readers are probably wondering if there is any *Bible* of R programming, and how Matloff’s book compares to this *Bible*. Most R programmers would answer to the first question that the *Bible* is Chambers’ (2008) book. Since I agree with them, I will just try to answer the second question. First of all, John Chamber is the creator of the S programming language on which R is built and he is also a member of the R Development Core Team. Chambers’ book, which is 500 pages long, is written for making experienced R users turn into R programmers and contributors. Chamber wants the reader to learn how R works internally and gives philosophical justifications to some of the choices made in developing R. In this way, he sheds light on some features of R that may appear idiosyncratic. Yet this higher understanding comes at a cost: Chambers’ book is somewhat wordy (you choose if this is good or bad) and the material is not always well organized. So, if one is interested in learning to program with R in a quick and painless way, Matloff’s book is the best solution, but for a deeper understanding of R Chambers’ book is
unreplaceable. Moreover, Chambers covers also package writing while Matloff does not (but I hope that Matloff will fill this gap in the second edition of his book).

My final suggestion is that if you belong to one of the following groups you should buy Matloff’s book:

- users of other languages that want to learn R programming;
- R users that use only built-in procedures and packages and want to start programming with R;
- R programmers with basic skills who want to improve their style and write more efficient code;
- teachers and students of a course in R programming.

**References**


**Department of Statistics, Università degli Studi di Milano-Bicocca, Via Bicocca degli Arcimboldi 8, 20126 Milano, Italy**
Basically, R programming language is an implementation of the S programming Language. Also combines with lexical scoping semantics inspired by Scheme. Although, R was named partly after the first names of two R programming language authors. Moreover, the project conceives in 1992, with an initial version released in 1995 and a stable beta version in 2000. Install R Programming Language & R Studio. In this R tutorial, we are moving towards installations of R Programming and R Studio. As the language of the book is quite simple to understand and examples can be reproduced easily. b. The Art of R Programming by Norman Matloff. Basically, this book tells how to do software development. As from basic types and data structures to advanced topics.