Journal of Statistical Software

September 2017, Volume 80, Book Review 3. doi: 10.18637/jss.v080.b03

Reviewer: Michel Lang
TU Dortmund University

Efficient R Programming

Colin Gillespie, Robin Lovelace
O'Reilly Media, Sebastopol, 2017.
ISBN 978-1-491-95078-4. 222 pp. USD 39.99 (P).
https://csgillespie.github.io/efficientR/

Overview

R has been developed with a strong focus on interactive data analysis using a read-eval-print loop (REPL), and many operations have been optimized for convenient interactive use rather than for computational performance. As data grows bigger and more complex, this consequently leads more and more often to situations where runtime or memory consumption are road blocks in the data analysis. Furthermore, interacting with R via the REPL by sending code in a piecemeal fashion frequently leads to situations where programmers loose track of the state of the data analysis, repeatedly write the same code to perform the same operation, and all in all interact inefficiently with the R language.

The authors of *Efficient R Programming* try to tackle these problems by giving an encompassing overview of the most important levers to increase both algorithmic efficiency and programmers productivity (the amount of work a person can do per unit time). The covered topics can roughly be divided into three categories:

Workflow: Set up of a working environment, project management, coding styles, version control systems, finding help efficiently.

Programming: Vectorization, apply family of functions, pre-allocation, type juggling, caching, indexing, parallelization, porting code to C++.

Data wrangling: Input and output of text and binary data, reshaping and manipulation of tabular data, databases.

The authors state to aim at a target audience of three groups: (1) experienced programmers with little experience with R, (2) experienced R users with little programming experience, and (3) R beginners with little experience in programming.

The 200-page book is structured into ten chapters, whereas each chapter after the introduction starts with the *Top Five Tips* for the respective topic, summarizing the conclusions of the subordinate sections. The chapters are in no obvious order but are self-contained, thus the book presents itself more like a reference book than a classical text book. Exercises allow the readers to consolidate their knowledge for some topics, but not all. Supplementary source code and data is provided as an R package hosted on GitHub.

It is worth mentioning that the book comes with many external references which are nicely hyperlinked in the electronic version, but hard to follow in the print version.

Chapter discussions

In the introductory chapter the authors start off by distinguishing between algorithmic efficiency and programming productivity. The authors make the excellent argument that every algorithmic optimization has to be taken with a grain of salt, as the speedup may never exceed the time required to implement it. The key concepts of benchmarking with **microbenchmark** and profiling with **profvis** are also covered in the introduction.

Chapter 2 is primarily intended for R beginners and describes how to set up your system to work efficiently with R. The authors explain how to install R, keep its library up to date and customize R's initialization files. The authors briefly discuss alternatives for the shipped BLAS as well as alternative R interpreters. Furthermore, **RStudio** is presented as IDE, including the most important keyboard shortcuts and hints to many crucial settings.

Chapter 3 focuses on improving efficiency via programming by illustrating the paradigms of efficient R programming: vectorization, pre-allocation, and avoiding loops in favor of apply functions. Lesser known techniques for closures, condition handling and factors are addressing the more advanced reader. Furthermore, memoization (caching the return value of a function based on its parameters) with the package **memoise** is shown.

Chapter 4 covers efficient project planning. Besides generally helpful advice, this chapter also includes the choice of R packages to use in the project and criteria to judge their quality. The last section of this chapter recommends to publish the results in a report with **knitr** or create a standalone R package. The part on R packages is more or less a stub, referring the reader to Wickham (2015).

Chapter 5 discusses packages for efficient input/output (IO) for plain text formats (**rio**, **readr**, **data.table**), R's binary format (RData/RDS) and cross-language, cross-platform binary formats (**feather**, **RProtoBuf**). For comparison, the packages are benchmarked for different file sizes in order to reveal their scaling behavior. Additionally, loading data from some internet resources and accessing data stored in packages is briefly outlined.

Chapter 6 is devoted to data wrangling, here called data carpentry. The authors focus on the so-called *tidy data* principle and its accompanying *tidyverse* packages, mainly **tibble** as a replacement for data frames, **tidyr** to reshape the data, and **dplyr** for operations on data frames and databases. The part on **data.table** is comparatively short, although the authors present a benchmark showing superior performance of **data.table** over **dplyr**. The authors also scratch on the surface of data access via **DBI** and string manipulation with regular expressions.

Chapter 7 focuses again on the programming aspect of efficiency. In contrast to Chapter 3 it deals with techniques to optimize existing code rather than writing efficient code from scratch.

The package **profvis** is applied to profile a simulation of the Monopoly board game and reveals some interesting bottlenecks. The chapter continues with many small sections about rather specific optimizations, e.g., do's and don'ts for R's base functions (`ifelse()` vs. `if()-else`, `&` vs. `&&`, ...), explicit use of integers, **matrixStats** for efficient matrix operations, or sparse matrices. The presented optimizations are applied to the Monopoly simulation and yield an impressive 20-fold speedup over the first version. Parallel computing unfortunately only covers parallelization on the local machine using **parallel**'s multicore mode. Here, I clearly miss references to socket clusters for windows, to abstractions for parallel backends like **foreach** or **parallelMap**, and to the CRAN task view *High-Performance and Parallel Computing with R* (Eddelbuettel 2017) which lists many packages that are implicitly parallelized. Thereafter, the authors introduce on roughly 10 pages how to translate R loops to C++ using **Rcpp**. This is a topic which likely overwhelms most readers unless they already have sound experience with C/C++.

Chapter 8 deals with hardware and explains the basics of random access memory, and why it is obviously good to have plenty. Furthermore, the authors provide some background knowledge about 32-bit vs. 64-bit operating systems and compare HDDs to SDDs. The package **benchmarkme** is introduced to generate CPU benchmark scores, followed by two short paragraphs about cloud computing.

Chapter 9 describes the prerequisites of collaborative programming: agreeing on a coding style and naming conventions. Git is briefly outlined as recommended version control system. This includes the key concepts of branching, forking, pulls and clones.

The final Chapter 10 addresses beginners again and deals with efficient learning: using R's internal help system, discovering vignettes, producing minimal reproducible examples and asking the right questions on external online resources like StackOverflow.

Conclusions

Gillespie and Lovelace give an excellent overview of possibilities to improve efficiency and productivity. They mostly find a good balance between explaining fundamental ideas, illustrating key concepts with exemplary code and providing technical background information. This is based on the internal structure of the sections: more technical notes as well as warnings about common pitfalls are typeset in blocks to not distract the reader from the thread. The reader is rightly referred to external resources for complex topics out of scope for this book, e.g., creating an R package or regular expressions for string operations. The book is therefore clearly not intended as a standalone resource to teach yourself efficient R programming, but rather gives a good starting point to dive deeper into the respective topics. As such, the book is appealing to a broad audience.

This being said, the wide spectrum of the book is probably also its biggest limitation. Depending on the skill level, some sections in the book will not be relevant to the reader. Experienced programmers usually know what operating system they are running, have heard about the concept of touch typing and already have their working environment set up.

Beginners on the other hand are better off skipping some topics, e.g., the 10 page introduction to C++. However, catching a glimpse of approaches and techniques which are still too advanced might prove useful for future projects to protect the reader from reinventing the wheel. Nevertheless, the reader could be guided better through these advanced topics in order to avoid confusion and frustration. For example, it could be stated that it is a prerequisite to familiarize with Unix, SSH and virtualization before even attempting to start cloud computing with R.

All in all, probably nobody will find each chapter helpful, but nearly everyone will find some bits and pieces which substantial improve their efficiency while working with R. Therefore I recommend the book to novice and ambitious R users and to users switching from other programming languages to R.

References

Eddelbuettel D (2017). *CRAN Task View: High-Performance and Parallel Computing with R.* Version 2017-08-29, URL https://CRAN.R-project.org/view=HighPerformanceComputing.

Wickham H (2015). *R Packages: Organize, Test, Document, and Share your Code.* O'Reilly Media, Sebastopol.

Reviewer:

Michel Lang
TU Dortmund University
Department of Statistics
44227 Dortmund, Germany
E-mail: lang@statistik.tu-dortmund.de
URL: https://www.statistik.tu-dortmund.de/lang00.html

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
September 2017, Volume 80, Book Review 3 *Published:* 2017-09-15
doi:10.18637/jss.v080.b03