Journal of Statistical Software

May 2018, Volume 85, Book Review 1. *doi: 10.18637/jss.v085.b01*

Reviewer: Daniel Grose
Lancaster University

Testing R Code

Richard Cotton
Chapman & Hall/CRC, Boca Raton, 2017.
ISBN 978-1-4987-6365-3. 178 pp. GBP 27.19.
http://www.crcpress.com/9781498763653

The basic premise of this book is that "Testing is the only way to be sure that your code, and your results, are correct", and many programmers that I know would agree with this[1]. However, in practice, and for many different reasons, much research software is still developed with only a passing gesture to this principle. The author believes that "Testing code is conceptually easy; the hard part is developing habits to incorporate testing into your workflow", and this may explain, at least in part, the shortfall of testing. Of course, software testing is a broad field that encompasses many issues from numerical stability and platform dependencies to graphical user interfaces, security and real-time processing. This means it is difficult to provide a small number of good, interoperable generic tools that cover all aspects of software development, whilst facilitating the development of these habits. However, it is possible when a specific programming language or environment is considered in isolation. In my opinion this book, in conjunction with the the tools that it employs, goes a long way towards realising this for the R programming environment. This book is suitable for anyone with a basic knowledge of the R programming language and is relevant for all who program using R, whether it be for creating small scripts, complete packages or anything in between.

The book sets out by distinguishing between two types of testing – development time and run time. In the author's words – "The point of development time testing is to make sure that you [the programmer] have not done something stupid. By contrast, the point of run-time testing is to make sure that the user has not done something stupid." These two types are examined separately in Chapters 2 and 3, which detail the two key packages on which the book is based - **assertive** (Cotton 2016) for run time testing (maintained by the author) and **testthat** (Wickham 2011) for development time testing. They also include numerous examples and exercises, with solutions, which underpin the authors imperative "Attempt all exercises!". Having attempted most of them myself it is clear that they are well thought out and instructive. Both chapters are very complete, and at the end of reading them I was pleasantly surprised as to how quickly I could implement the methods they introduced into

[1]Nearly all of them would mostly agree, and at least one would quote Edsger Dijkstra – "Program testing can be used to show the presence of bugs, but never show their absence!".

my day to day programming. In fact, I believe these two chapters alone constitute a valuable resource for learning and implementing the testing of R code.

Chapter 4 introduces heuristics and techniques for programming which result in more testable code. It would be easy in undertaking this task to do so in a dogmatic fashion and with little supporting evidence. That is not the case for this book, which makes two common sense suggestions – do not repeat yourself, and keep it simple stupid – which are then backed up by several specific ideas as to how to achieve this, a motivating case study and the ever ubiquitous exercises.

Package development has by now established itself as a standard way to share and distribute reusable methodology, data and programming tools for R. It is therefore important that development time testing can be incorporated into the package development process, and how this is achieved is explained in Chapter 5. The chapter starts with an introduction to R package development which is necessary so that integration of **testthat** into the package structure can be demonstrated. However, this should not be seen as a complete standalone reference for package development which would not be appropriate in a book focussed on testing. There already exist good sources of information to which the reader can turn to on this matter, for example Wickham (2015). The chapter does however provide some useful background on version control (using `git`) and on several issues regarding submission of R packages to CRAN.

Chapters 6 and 7 form the advanced content of the book. Chapter 6 concerns development time testing in the context of data bases, compiled C++ code with **Rcpp**, graphics and code that has side effects (for example writing to files, opening web connections and so on). As I optimise a lot of my code using compilation I found the section on integrating **testthat** with **Rcpp** (Eddelbuettel and François 2011; Eddelbuettel 2013) particularly useful. Until recently my C++, R code and package testing were conducted separately. Having read this book this is no longer the case, and I have already saved considerable time and effort by adopting the approaches introduced in it.

The **assertive** and **testthat** packages are impressively comprehensive, but as the author points out, there is a practically unlimited number of possible assertions and tests that could be applied. For this reason both packages are designed to be extensible by the programmer so that further testing tools do not have to be adopted and learnt. How these extensions are created and used is explained in Chapter 7.

The book is well organised. Important notes and key concepts are boxed and easy to find. Each chapter has a brief summary, exercises have answers and there is very useful function index. There are a small number of minor typographical errors but I found none that caused any ambiguity in the main material. Unfortunately, the numbering in the contents is not properly matched to the location of the material in the book which can be a little frustrating, particularly since the book can be used as a reference manual for **assertive** and **testthat**.

It was easy to engage with this book not only because of its value as a self contained instruction and reference manual for testing R code, but also due to its light-hearted and yet matter of fact style. This is a credit to the author who has managed to produce a valuable and enjoyable book on a topic that is often considered quite dry and unapproachable.

References

Cotton R (2016). ***assertive**: Readable Check Functions to Ensure Code Integrity*. R package version 0.3-5, URL https://CRAN.R-project.org/package=assertive.

Eddelbuettel D (2013). *Seamless R and C++ Integration with **Rcpp***. Springer-Verlag, New York.

Eddelbuettel D, François R (2011). "**Rcpp**: Seamless R and C++ Integration." *Journal of Statistical Software*, **40**(8), 1–18. doi:10.18637/jss.v040.i08.

Wickham H (2011). "**testthat**: Get Started with Testing." *The R Journal*, **3**, 5–10. URL https://journal.R-project.org/archive/2011-1/RJournal_2011-1_Wickham.pdf.

Wickham H (2015). *R Packages*. O'Reilly.

Reviewer:

Daniel Grose
Lancaster University
Department of Mathematics and Statistics
Fylde College
Lancaster University
Lancaster LA1 4YF, United Kingdom
E-mail: dan.grose@lancaster.ac.uk
URL: http://www.lancaster.ac.uk/maths/about-us/people/daniel-grose

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
May 2018, Volume 85, Book Review 1 *Published:* 2018-06-10
doi:10.18637/jss.v085.b01