

University of Nebraska - Lincoln

DigitalCommons@University of Nebraska - Lincoln

---

The R Journal

Statistics, Department of

---

12-2021

## The R Developer Community Does Have a Strong Software Engineering Culture

Maëlle Salmon

Karthik Ram

Follow this and additional works at: <https://digitalcommons.unl.edu/r-journal>



Part of the [Numerical Analysis and Scientific Computing Commons](#), and the [Programming Languages and Compilers Commons](#)

---

This Article is brought to you for free and open access by the Statistics, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in The R Journal by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

# The R Developer Community Does Have a Strong Software Engineering Culture

by *Maëlle Salmon and Karthik Ram*

**Abstract** There is a strong software engineering culture in the R developer community. We recommend creating, updating and vetting packages as well as keeping up with community standards. We invite contributions to the rOpenSci project, where participants can gain experience that will shape their work and that of their peers.

## Introduction

The R programming language was originally created for statisticians, by statisticians, but evolved over time to attract a “massive pool of talent that was previously untapped” (Hadley Wickham in Thieme (2018)). Despite the fact that most R users are academic researchers and business data analysts without a background in software engineering, we are witnessing a rapid rise in software engineering within the community. In this comment we spotlight recent progress in tooling, dissemination and support, including specific efforts led by the rOpenSci project. We hope that readers will take advantage of and participate in the tools and practices we describe.

## The modern R package developer toolbox: user-friendlier, more comprehensive

The basic infrastructure for creating, building, installing, and checking packages has been in place since the early days of the R language. During this time (1998-2011), the barriers to entry were very high and access to support and Q&A for beginners were extremely limited. With the introduction of the **devtools** (Wickham et al., 2021b) package in 2011, the process of creating and updating packages became substantially easier. Documentation also became simpler to maintain. The **roxygen2** (Wickham et al., 2021a) package allowed developers to keep documentation in sync with changes in code, similar to the doxygen approach that was embraced in more mature languages. Combined with the rise in popularity of StackOverflow and the growth of rstats blogs, the number of packages on the Comprehensive R Archive Network (CRAN) skyrocketed from 400 new packages in 2010 to 1000 new packages by 2014. As of this writing, there are nearly 19k packages on CRAN.

For novices without substantial software engineer experience, the early testing frameworks were also difficult to use. With the release of **testthat** (Wickham, 2011), testing also became smoother. There are now several actively maintained testing frameworks such as **tinytest** (van der Loo, 2020); as well as testthat-compatible specialized tooling for testing database interactions (**dittodb** (Keane and Vargas, 2020)), web resources (**vcr** (Chamberlain, 2021)), **httptest** (Richardson, 2021), and **webfakes** (Csárdi, 2021) which enables the use of an embedded C/C++ web server for testing HTTP clients like **httr2** (Wickham, 2021)).

The testthat package has recently been improved with snapshot tests that make it possible to test plot outputs. The rOpenSci project has released **autotest** (Padgham, 2021), a package that supports automatic mutation testing.

Beyond checking for compliance with R CMD CHECK, several other packages such as **good-practice** (Csárdi and Frick, 2018), **riskmetric** (R Validation Hub et al., 2021), rOpenSci’s **pkgcheck** (Padgham and Salmon, 2021) check packages against a large list of actionable, community recommended best practices for software development. Collectively these tools allow domain researchers to release software packages that meet high standards for software engineering.

The development and testing ecosystem of R is rich and has sometimes borrowed successful implementations from other languages (e.g. the vcr R package is a port, i.e. translation to R, of the vcr Ruby gem; testthat snapshot tests were inspired by JS Jest<sup>1</sup>).

## Emergence of a welcoming community

As underlined in Thieme (2018), community is the strong suit of the R language. Many organizations and venues offer dedicated support for package developers. Examples include Q&A on the r-package-devel mailing list<sup>2</sup>, and the package development category of the RStudio community forum<sup>3</sup>, and

<sup>1</sup><https://www.tidyverse.org/blog/2020/10/testthat-3-0-0/#snapshot-testing>

<sup>2</sup><https://stat.ethz.ch/mailman/listinfo/r-package-devel>

<sup>3</sup><https://community.rstudio.com/c/package-development/11>

the rstats section of StackOverflow<sup>4</sup>. Traditionally, R package developers have been mostly male and white. Although the status quo remains similar, efforts from groups such as R-Ladies<sup>5</sup> meetups, Minorities in R (Scott and Smalls-Perkins, 2020), and the package development modules offered by Forwards for underrepresented groups<sup>6</sup> have made considerable inroads towards improving diversity. These efforts have worked hard to put the spotlight on developers beyond the “usual suspects”.

### rOpenSci community and software review

The rOpenSci organization (Boettiger et al., 2015) is an attractive venue for developers & supporters of scientific R software. One of our most successful and continuing initiatives is our Software Peer Review system (Ram et al., 2019), a combination of academic peer-review and code review from industry. About 150 packages have been reviewed by volunteers to date, creating better packages as well as a growing knowledgebase in our development guide (rOpenSci et al., 2021) while also building a living community of practice.

Our model has been the fundamental inspiration for projects such as the Journal of Open Source Software (Smith et al., 2018), and PyOpenSci [Wasser and Holdgraf (2019)](Trizna et al., 2021). We are continuously improving our system and reducing cognitive overload on editors and reviewers by automating repetitive tasks. Most recently we have expanded our offerings to peer review of packages that implement statistical methods (Statistical Software Peer Review) (Padgham et al., 2021).

Beside software review, rOpenSci community is a safe, welcoming and informative place for package developers, with Q&A happening on our public forum and semi-open Slack workspace. (Butland and LaZerte, 2020)

### Creation and dissemination of resources for R programmers

The aforementioned tools, venues and organizations benefit from and support crucial dissemination efforts.

Publishing technical know-how is crucial for progress of the R community. R news has been circulating on Twitter<sup>7</sup>, R Weekly<sup>8</sup> and R-Bloggers<sup>9</sup>. Some sources have been more specifically aimed at R package developers of various experience and interests. While “Writing R Extensions”<sup>10</sup> is the official & exhaustive reference on writing R packages, it is a reference rather than a learning resource: many R package developers, if not learning by example, get introduced to R package development via introductory blog posts or tutorials, and the R packages book by Hadley Wickham and Jenny Bryan [Wickham (2015)](Wickham and Bryan) that accompany the devtools suite of packages is freely available online and strives to improving the R package development experience. The rOpenSci guide “rOpenSci Packages: Development, Maintenance, and Peer Review” (rOpenSci et al., 2021) contains our community-contributed guidance on how to develop packages and review them. It features *opinionated requirements* such as the use of `roxygen2` (Wickham et al., 2021a) for package documentation; *criteria helping make an informed decision* on gray area topics such as limiting dependencies; *advice on widely accepted and emerging tools*. As it is a living document also used as reference for editorial decisions, we maintain a changelog<sup>11</sup>, and summarize each release in a blog post<sup>12</sup>. rOpenSci also hosts a book on a specialized topic, HTTP testing in R<sup>13</sup>, that presents both principles for testing packages that interact with web resources, as well as relevant packages. Beside these examples of long-form documentation, knowledge around R software engineering is shared through blogs and talks. In the R blogging world, the rOpenSci blog posts<sup>14</sup>, technical notes<sup>15</sup> and a section of our monthly newsletter<sup>16</sup> feature some topics relevant to package developers, as do some of the posts on the Tidyverse blog<sup>17</sup>. The blog of the R-hub project<sup>18</sup> contains information on package development topics, in particular about common problems such as sharing data via R packages or understanding CRAN checks. Expert programmers

<sup>4</sup><https://stackoverflow.com/questions/tagged/r?tab=Newest>

<sup>5</sup><http://rladies.org/>

<sup>6</sup><https://buzzrbeeline.blog/2021/02/09/r-forwards-package-development-modules-for-women-and-other-underrepresented-groups/>

<sup>7</sup><https://www.t4rstats.com/>

<sup>8</sup><https://rweekly.org/>

<sup>9</sup><https://www.r-bloggers.com/>

<sup>10</sup><https://cran.r-project.org/doc/manuals/R-exts.html>

<sup>11</sup><https://devguide.ropensci.org/booknews.html>

<sup>12</sup><https://ropensci.org/tags/dev-guide/>

<sup>13</sup><https://books.ropensci.org/http-testing/>

<sup>14</sup><https://ropensci.org/blog/>

<sup>15</sup><https://ropensci.org/technotes/>

<sup>16</sup><https://ropensci.org/news/>

<sup>17</sup><https://www.tidyverse.org/categories/programming/>

<sup>18</sup><https://blog.r-hub.io/post/>

have been sharing their R specific wisdom as well as software engineering lessons learned from other languages (e.g. Jenny Bryan’s useR! Keynote address “code feels, code smells”<sup>19</sup>).

## Conclusion

In summary, we observe that there is already a strong software engineering culture in the R developer community. By surfacing the rich suite of resources to new developers we can but only hope the future will bring success to all aforementioned initiatives. We recommend creating, updating and vetting packages with the tools we mentioned as well as keeping up with community standards with the venues we mentioned in the previous section. We invite contributions to the rOpenSci project, where participants can gain experience that will shape their work and that of their peers. Thanks to these efforts, we hope the R community will continue to be a thriving place of application for software engineering, by diverse practitioners from many different paths.

## Bibliography

- C. Boettiger, S. Chamberlain, E. Hart, and K. Ram. Building software, building community: Lessons from the ropensci project. *Journal of Open Research Software*, 3(1):e8, 2015. doi: 10.5334/jors.bu. [p19]
- S. Butland and S. LaZerte. *rOpenSci Community Contributing Guide*. Zenodo, 2020. doi: 10.5281/ZENODO.4000532. URL <https://contributing.ropensci.org/>. [p19]
- S. Chamberlain. *vcr: Record ‘HTTP’ Calls to Disk*, 2021. URL <https://CRAN.R-project.org/package=vcr>. R package version 1.0.2. [p18]
- G. Csárdi. *webfakes: Fake Web Apps for HTTP Testing*, 2021. <https://webfakes.r-lib.org/>, <https://github.com/r-lib/webfakes>. [p18]
- G. Csárdi and H. Frick. *goodpractice: Advice on R Package Building*, 2018. URL <https://CRAN.R-project.org/package=goodpractice>. R package version 1.0.2. [p18]
- J. Keane and M. Vargas. *dittodb: A Test Environment for Database Requests*, 2020. URL <https://CRAN.R-project.org/package=dittodb>. R package version 0.1.3. [p18]
- M. Padgham. *autotest: Automatic Package Testing*, 2021. <https://docs.ropensci.org/autotest/>, <https://github.com/ropensci-review-tools/autotest>. [p18]
- M. Padgham and M. Salmon. *pkgcheck: rOpenSci Package Checks*, 2021. <https://docs.ropensci.org/pkgcheck/>, <https://github.com/ropensci-review-tools/pkgcheck>. [p18]
- M. Padgham, M. Salmon, N. Ross, J. Nowosad, R. FitzJohn, yilong zhang, C. Sax, F. Rodriguez-Sanchez, F. Briatte, and L. Collado-Torres. ropensci/statistical-software-review-book: Official first standards versions, Oct. 2021. URL <https://doi.org/10.5281/zenodo.5556756>. [p19]
- R Validation Hub, D. Kelkhoff, M. Gotti, E. Miller, K. K, Y. Zhang, E. Milliman, and J. Manitz. *riskmetric: Risk Metrics to Evaluating R Packages*, 2021. <https://pharmar.github.io/riskmetric/>, <https://github.com/pharmaR/riskmetric>. [p18]
- K. Ram, C. Boettiger, S. Chamberlain, N. Ross, M. Salmon, and S. Butland. A community of practice around peer review for long-term research software sustainability. *Computing in Science Engineering*, 21(2):59–65, 2019. doi: 10.1109/MCSE.2018.2882753. [p19]
- N. Richardson. *httptest: A Test Environment for HTTP Requests*, 2021. <https://enpiar.com/r/httptest/>, <https://github.com/nealrichardson/httptest>. [p18]
- rOpenSci, B. Anderson, S. Chamberlain, L. DeCicco, J. Gustavsen, A. Krystalli, M. Lepore, L. Mullen, K. Ram, N. Ross, M. Salmon, and M. Vidoni. rOpenSci Packages: Development, Maintenance, and Peer Review, Feb. 2021. URL <https://doi.org/10.5281/zenodo.4554776>. [p19]
- D. Scott and D. Smalls-Perkins. Introducing mir: A community for underrepresented minority users of r, Feb 2020. URL <https://medium.com/@doritolay/introducing-mir-a-community-for-underrepresented-users-of-r-7560def7d861>. [p19]

<sup>19</sup><https://github.com/jennybc/code-smells-and-feels>

- A. M. Smith, K. E. Niemeyer, D. S. Katz, L. A. Barba, G. Githinji, M. Gymrek, K. D. Huff, C. R. Madan, A. C. Mayes, K. M. Moerman, P. Prins, K. Ram, A. Rokem, T. K. Teal, R. V. Guimera, and J. T. Vanderplas. Journal of open source software (JOSS): design and first-year review. *PeerJ Computer Science*, 4:e147, Feb. 2018. doi: 10.7717/peerj-cs.147. URL <https://doi.org/10.7717/peerj-cs.147>. [p19]
- N. Thieme. R generation. *Significance*, 15(4):14–19, 2018. doi: 10.1111/j.1740-9713.2018.01169.x. URL <https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1740-9713.2018.01169.x>. [p18]
- M. Trizna, L. A Wasser, and D. Nicholson. pyopensci: Open and reproducible research, powered by python. *Biodiversity Information Science and Standards*, 5:e75688, 2021. doi: 10.3897/biss.5.75688. URL <https://doi.org/10.3897/biss.5.75688>. [p19]
- M. van der Loo. A method for deriving information from running r code. *The R Journal*, page Accepted for publication, 2020. URL <https://arxiv.org/abs/2002.07472>. [p18]
- L. A. Wasser and C. Holdgraf. pyOpenSci Promoting Open Source Python Software To Support Open Reproducible Science. In *AGU Fall Meeting Abstracts*, volume 2019, pages NS21A–13, Dec. 2019. [p19]
- H. Wickham. testthat: Get started with testing. *The R Journal*, 3:5–10, 2011. URL [https://journal.r-project.org/archive/2011-1/RJournal\\_2011-1\\_Wickham.pdf](https://journal.r-project.org/archive/2011-1/RJournal_2011-1_Wickham.pdf). [p18]
- H. Wickham. *R Packages*. O’Reilly Media, 2015. [p19]
- H. Wickham. *httr2: Perform HTTP Requests and Process the Responses*, 2021. URL <https://CRAN.R-project.org/package=httr2>. R package version 0.1.1. [p18]
- H. Wickham and J. Bryan. R packages. Second Edition [Online]. URL <https://r-pkgs.org/>. [p19]
- H. Wickham, P. Danenberg, G. Csárdi, and M. Eugster. *roxygen2: In-Line Documentation for R*, 2021a. URL <https://CRAN.R-project.org/package=roxygen2>. R package version 7.1.2. [p18, 19]
- H. Wickham, J. Hester, and W. Chang. *devtools: Tools to Make Developing R Packages Easier*, 2021b. URL <https://CRAN.R-project.org/package=devtools>. R package version 2.4.2. [p18]

*Maëlle Salmon*  
*The rOpenSci Project*

<https://masalmon.eu>  
ORCID: 0000-0002-2815-0399  
[msmaellesalmon@gmail.com](mailto:msmaellesalmon@gmail.com)

*Karthik Ram*  
*Berkeley Institute for Data Science and The rOpenSci Project*

<https://ram.berkeley.edu/>  
ORCID: 0000-0002-0233-1757  
[karthik.ram@berkeley.edu](mailto:karthik.ram@berkeley.edu)