

# The R Package Review Process

Perspectives from an rOpenSci Associate Editor

---

Brooke Anderson  
Colorado State University

✉: [broke.anderson@colostate.edu](mailto:broke.anderson@colostate.edu)

🐦: [@gbwanderson](https://twitter.com/gbwanderson)

🐙: [github.com/geanders](https://github.com/geanders)

## R Packages: What and Why

---

# Base R

The screenshot shows the CRAN homepage. On the left, there's a sidebar with links like CRAN, Mirrors, What's new?, Task Views, Search, About R, R Homepage, The R Journal, Software, R Sources, R Binaries, Packages, Other, Documentation, Manuals, FAQs, and Contributed. The main content area has a large R logo. It features a section titled "Download and Install R" with a red box around the download links for Linux, Mac OS X, and Windows. Below this, a note says R is part of many Linux distributions. Further down, it discusses source code and provides a list of bullet points about releases and sources.

←

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2020-10-10, Bunny-Wunnies Freak Out) [R-4.0.3.tar.gz](#), read [what's new](#) in the latest version.
- Sources of [R alpha and beta releases](#) (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are [available here](#). Please read about [new features and bug fixes](#) before filing corresponding feature requests or bug reports.
- Source code of older versions of R is [available here](#).
- Contributed extension [packages](#)

Comprehensive R Archive Network: <https://cran.r-project.org/>

## Base R



Source: Brio

# R Package Ecosystem



Source: Brio

# R Package Ecosystem



Mitchell O'Hara-Wild:

<https://www.mitchelloharawild.com/blog/user-2018-feature-wall/>

# R Package Ecosystem



Mitchell O'Hara-Wild:

<https://www.mitchelloharawild.com/blog/user-2018-feature-wall/>

# What's in an R Package?

Process:

$$\text{Temperature (F)} = \frac{9}{5} \text{Temperature (C)} + 32$$

Example R function:

```
celsius.to.fahrenheit <-
  function (T.celsius, round = 2)
{
  T.fahrenheit <- (9/5) * T.celsius + 32
  T.fahrenheit <- round(T.fahrenheit, digits = round)
  return(T.fahrenheit)
}
```

# What's in an R Package?

## Example help file:

`celsius.to.fahrenheit` {weathermetrics}

R Documentation

Convert from Celsius to Fahrenheit.

### Description

`celsius.to.fahrenheit` creates a numeric vector of temperatures in Fahrenheit from a numeric vector of temperatures in Celsius.

### Usage

```
celsius.to.fahrenheit(T.celsius, round = 2)
```

### Arguments

`T.celsius` Numeric vector of temperatures in Celsius.

`round` An integer indicating the number of decimal places to round the converted value.

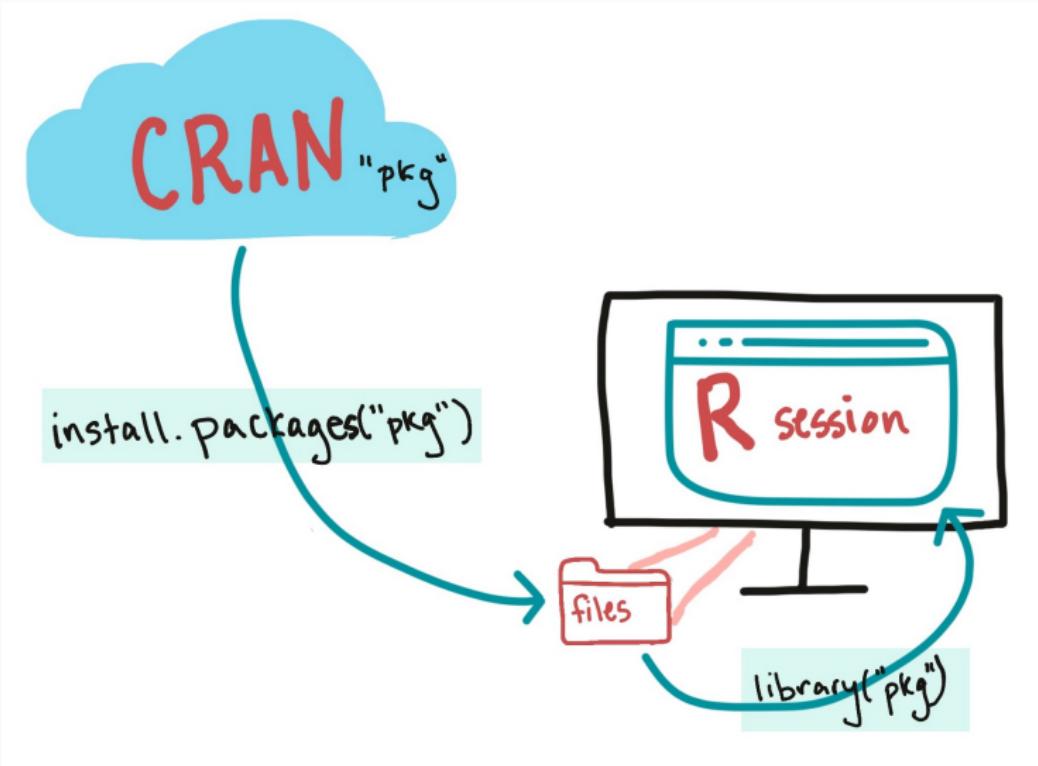
### Value

A numeric vector of temperature values in Fahrenheit.

# What's in an R Package?

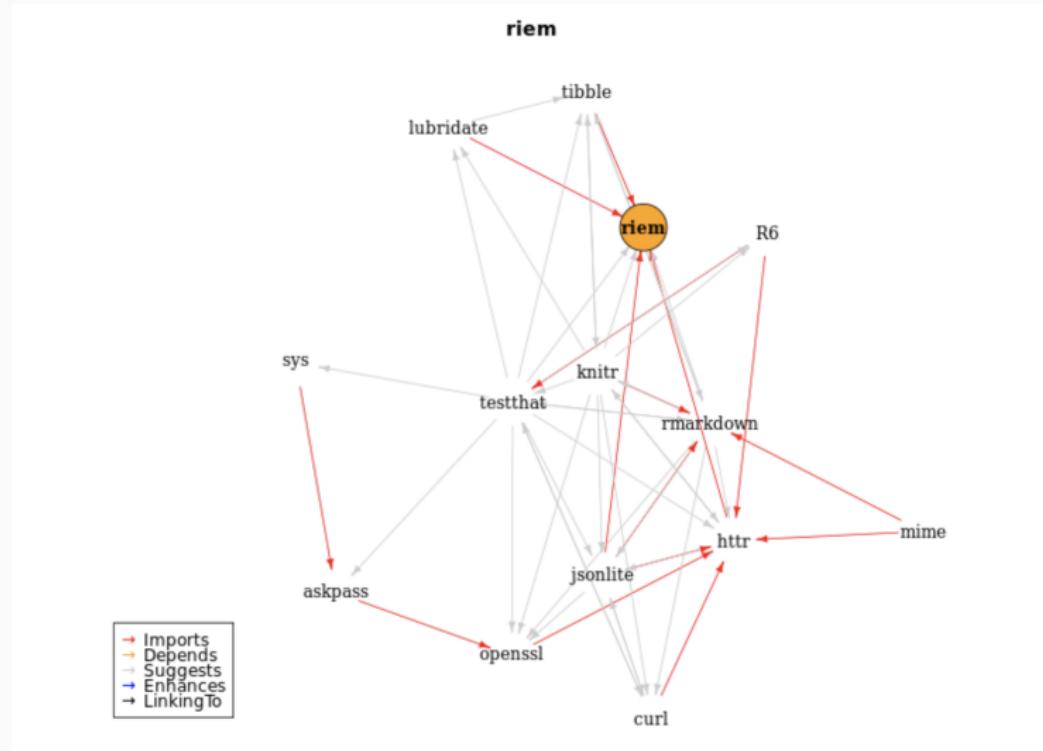
Folders	Documents	Developer
weathermetrics ►	cran-comments.md NEWS.md README.md	data.R heat_index.R moisture_conversions.R rainmeasure_conversion.R temperature_conversions.R weathermetrics.R wind_conversions.R
PDF Documents		
weathermetrics.pdf		
Other		
weathermetrics_1.2.0.tar.gz		
weathermetrics_1.2.2.tar.gz		

# How are R Packages Shared?



# How are R Packages Shared?

Dependencies for the rOpenSci package `riem`:



## **Peer Review: What and Why**

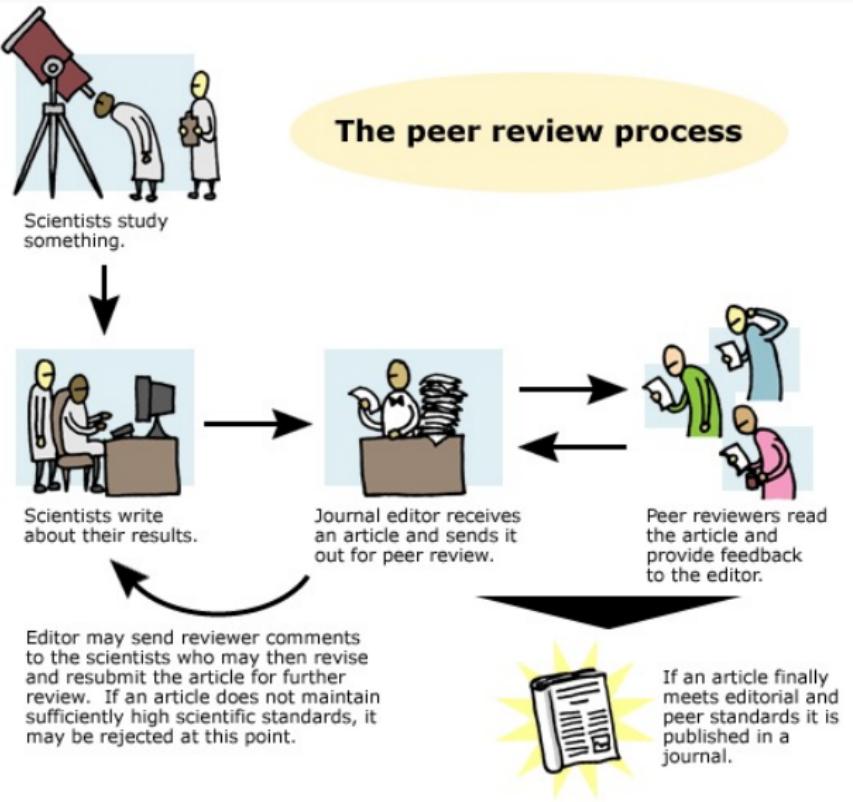
---

# Early Steps



Henri Testelin, <https://www.wga.hu/frames-e.html?/html/t/testelin/colbert.html>

# The Journal Peer Review Process



# The Challenges Continue Today

## The preprint problem: Unvetted science is fueling COVID-19 misinformation

Peer review moves to Twitter, muddling public health information.

JONATHAN M. GITLIN - 5/6/2020, 11:39 AM



Jonathan M. Gitlin: <https://arstechnica.com/science/2020/05/a-lot-of-covid-19-papers-havent-been-peer-reviewed-reader-beware/>

# Peer Review Related to R Code



## Navigation

[Current Issue](#)

[Accepted articles](#)

[Archive](#)

[R News](#)

[News and Notes](#)

[Submissions](#)

[Reviews and Proofreading](#)

[Editorial Board](#)

## Subscribe

[RSS Feed](#)

ISSN: 2073-4859

## The R Journal

*The R Journal* is the open access, refereed journal of the [R project](#) for statistical computing. It features short to medium length articles covering topics that should be of interest to users or developers of R. *The R Journal* intends to reach a wide audience and have a thorough review process. Papers are expected to be reasonably short, clearly written, not too technical, and of course focused on R. Authors of refereed articles should take care to:

- put their contribution in context, in particular discuss related R functions or packages;
- explain the motivation for their contribution;
- provide code examples that are reproducible.

Following revision of the content description of *The R Journal*, from January 2017 submitted articles may include:

### **Reviews and proposals:**

surveying and discussing challenges and opportunities of potential importance for the broader R community, including proposals and proof-of-concept implementations.

### **Comparisons and benchmarking:**

of implementations in base-R and contributed packages with each other, and where relevant with implementations in other software systems.

### **Applications:**

demonstrating how new or existing techniques can be applied in an area of current interest using R, providing a fresh view of such analyses in R that is of benefit beyond the specific application.

<https://journal.r-project.org/>

# Peer Review Related to R Code

*Journal of Statistical Software*

Instructions for Authors | Style Guide | Editorial Board | Contact | Register | Current Volume

Home > Vol 95 (2020)

Established in 1996, the Journal of Statistical Software publishes articles, book reviews, code snippets, and software reviews on the subject of statistical software and algorithms. The contents are freely available online. For both articles and code snippets, the source code is published along with the paper. Statistical software is the key link between statistical methods and their application in practice. Software that makes this link is the province of the journal, and may be realized as, for instance, tools for large scale computing, database technology, desktop computing, distributed systems, the World Wide Web, reproducible research, archiving and documentation, and embedded systems. We attempt to present research that demonstrates the joint evolution of computational and statistical methods and techniques. Implementations can use languages such as C, C++, S, Fortran, Java, PHP, Python and Ruby or environments such as Mathematica, MATLAB, R, S-PLUS, SAS, Stata, and XLISP-STAT.



## Recent Publications

### Articles

[Various Versatile Variances: An Object-Oriented Implementation of Clustered Covariances in R](#) [PDF](#)  
Achim Zeileis, Susanne Köll, Nathaniel Graham

[Estimating Animal Abundance with N-Mixture Models Using the R-INLA Package for R](#) [PDF](#)  
Timothy D. Meehan, Nicole L. Michel, Håvard Rue

[A Data Envelopment Analysis Toolbox for MATLAB](#) [PDF](#)  
Inmaculada C. Alvarez, Javier Barbero, José L. Zofío

[Zigzag Expanded Navigation Plots in R: The R Package zenplots](#) [PDF](#)  
Marius Hofert, Wayne Oldford

[Computationally Efficient Simulation of Queues: The R Package](#) [PDF](#)

**OPEN JOURNAL SYSTEMS**  
[Journal Help](#)

**USER**  
Username   
Password   
 Remember me

**FEEDS**  
• [RSS](#)  
• [Atom](#)

**NOTIFICATIONS**  
• [View](#)  
• [Subscribe](#)

<https://www.jstatsoft.org/index>

# rOpenSci Packages and Package Review

---

# Transforming science through open data, software & reproducibility

We help develop R packages for the sciences via community driven learning, review and maintenance of contributed software in the R ecosystem

<https://ropensci.org>

## Packages

Our packages are carefully vetted, staff- and community-contributed R software tools that in particular lower barriers to working with local and remote scientific data sources. Browse [use cases](#) and read our [blog](#) to learn how to use specific packages or contribute to their improvement.

<https://ropensci.org/packages/>

# rOpenSci Package Categories

## Computing Infrastructure

Workflow Tools for Your Code and Data

## Data Access

Get Data from the Web

## Data Extraction

Convert and Munge Data

## Data Publication

Document and Release Your Data

## Data Visualization

Visualize Data

## Databases

Work with Databases From R

## Geospatial

Access, Manipulate, Convert Geospatial Data

## HTTP tools

Interact with Web Resources

## Image & Audio Processing

Use Image & Audio Data

## Literature

Analyze Scientific Papers (and Text in General)

## Security

Secure Your Data and Workflow

## Taxonomy

Handle and Transform Taxonomic Information

<https://ropensci.org/packages/>

# rOpenSci Example Data Access Packages



**FedData**

CRAN

Peer-reviewed

Functions to Automate Downloading Geospatial Data Available from Several Federated Data Sources

Maintainer

R. Kyle Bocinsky



**rredlist**

CRAN

Staff maintained

IUCN Red List Client

Maintainer

Scott Chamberlain



**spocc**

CRAN

Staff maintained

Interface to Species Occurrence Data Sources

Maintainer

Scott Chamberlain



**auk**

CRAN

Peer-reviewed

eBird Data Extraction and Processing in R

Maintainer

Matthew Strimas-Mackey



**MODISstsp**

CRAN

Peer-reviewed

A Tool for Automating Download and Preprocessing of MODIS Land Products Data

Maintainer

Lorenzo Busetto



<https://ropensci.org/packages/data-access/>

# rOpenSci Example Data Extraction Packages

	<b>spelling</b>	CRAN	Staff maintained	Maintainer Jeroen Ooms	
	Tools for Spell Checking in R				
	<b>CoordinateCleaner</b>	CRAN	Peer-reviewed	Maintainer Alexander Zizka	
	Automated Cleaning of Occurrence Records from Biological Collections				
	<b>scrubr</b>	CRAN	Staff maintained	Maintainer Scott Chamberlain	
	Clean Biological Occurrence Records				
	<b>treedata.table</b>	CRAN	Peer-reviewed	Maintainer Cristian Roman-Palacios	
	Manipulation of Matched Phylogenies and Data using data.table				
	<b>lightr</b>	CRAN	Peer-reviewed	Maintainer Hugo Gruson	
	Read Spectrometric Data and Metadata				

<https://ropensci.org/packages/data-extraction/>

## How rOpenSci uses Code Review to Promote Reproducible Science

by nf-admin | Aug 11, 2017 | Blog, Projects

*This post was co-authored by the rOpenSci Editorial Board: Noam Ross, Scott Chamberlain, Karthik Ram, and Maëlle Salmon.*

“With the community approach came challenges. How could we ensure the quality of code written by scientists without formal training in software development practices? How could we drive adoption of best practices among our contributors? How could we create a community that would support each other in this work? We have had great success addressing these challenges via the **peer review**.”

<https://numfocus.org/blog/how-ropensci-uses-code-review-to-promote-reproducible-science>

## How rOpenSci uses Code Review to Promote Reproducible Science

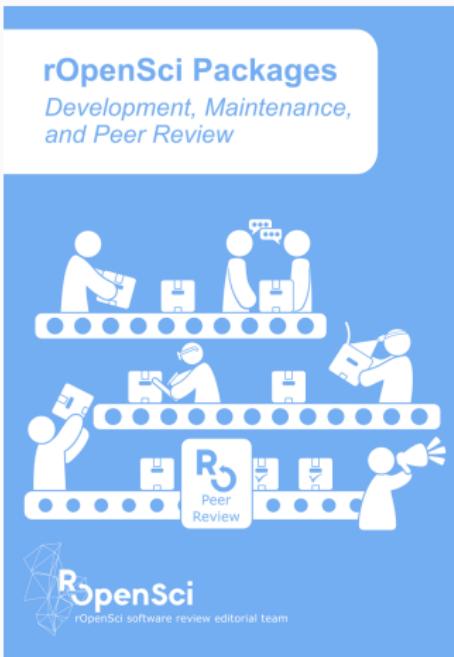
by nf-admin | Aug 11, 2017 | Blog, Projects

*This post was co-authored by the rOpenSci Editorial Board: Noam Ross, Scott Chamberlain, Karthik Ram, and Maëlle Salmon.*

"We draw elements from a process familiar to our target community—**academic peer review**—and a practice from the software development world—**production code review**—to create a system that fosters software quality, ongoing education, and community development."

<https://numfocus.org/blog/how-ropensci-uses-code-review-to-promote-reproducible-science>

# rOpenSci Package Review



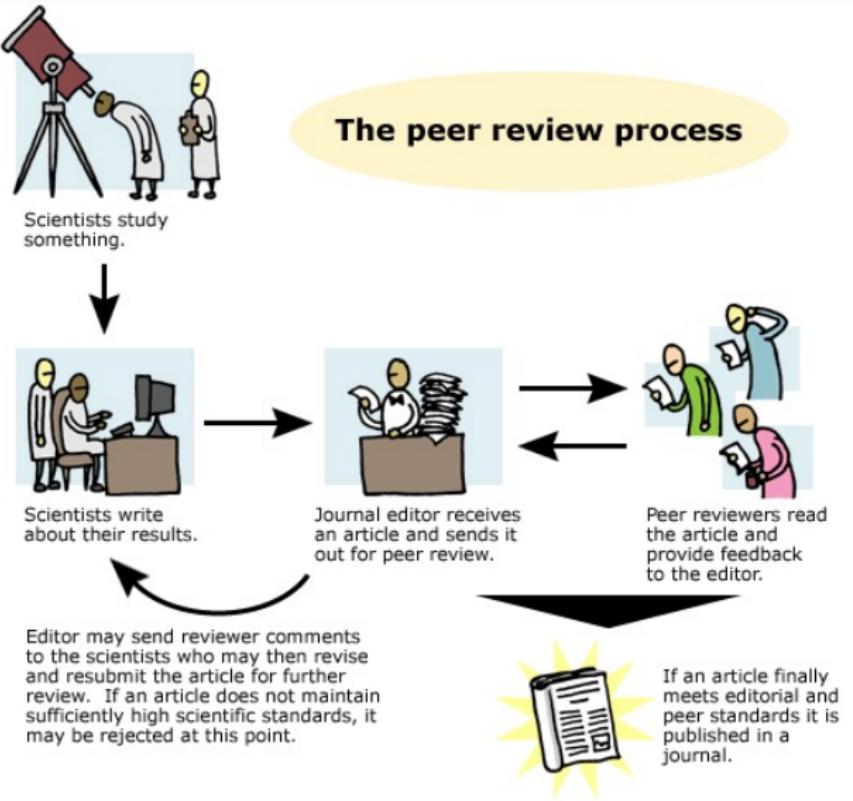
“Packages contributed by the community undergo a **transparent, constructive, non adversarial** and **open** review process”

<https://devguide.ropensci.org/>

# The rOpenSci Review Process

---

# The Journal Peer Review Process

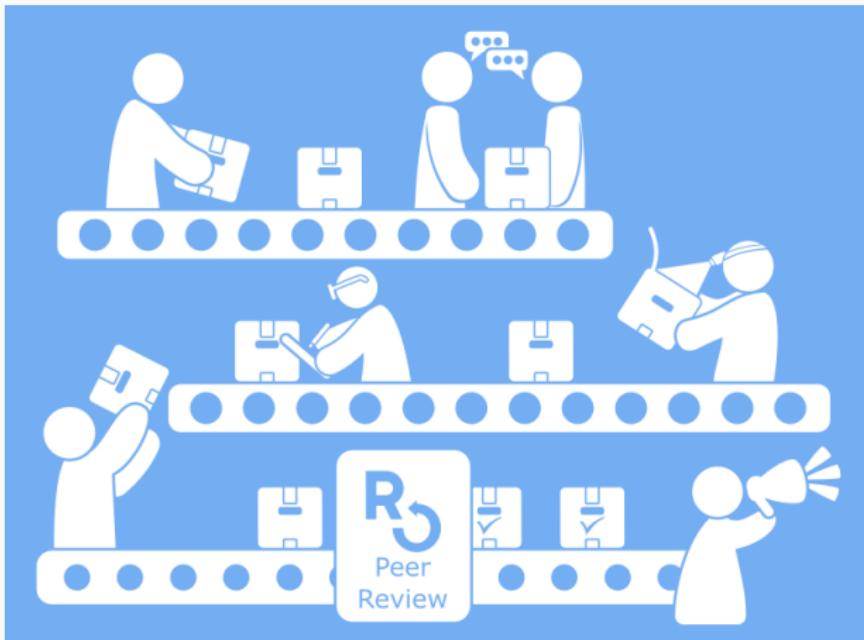


## Overview of Review Process

"In our process, **authors submit** complete R packages to rOpenSci. **Editors check** that packages fit into our project's scope, run a series of automated tests to ensure a baseline of code quality and completeness, and then **assign** two independent reviewers. **Reviewers comment** on usability, quality, and style of software code as well as documentation. **Authors make changes in response**, and once reviewers are satisfied with the updates, the package receives a badge of approval and joins our suite."

<https://numfocus.org/blog/how-ropensci-uses-code-review-to-promote-reproducible-science>

# Overview of Review Process



“Our reviewing threads are **public**. Authors, reviewers, and editors all know each other’s identities. The broader community can view or even participate in the conversation as it happens.”

# Review Framework

“Technically, we make the most of GitHub infrastructure: each package review process is **an issue** in the `ropensci/software-review` GitHub repository.”

<https://github.com/ropensci/software-review/issues>

The screenshot shows the GitHub interface for the `ropensci/software-review` repository. The search bar at the top contains the query `is:issue is:open`. Below the search bar, there are several filters: `Filters`, `Labels 57`, `Milestones 0`, and a `New issue` button. The main area displays 15 open issues:

- yotover: An Advanced Guide to Trade Policy Analysis** (#403) opened 14 hours ago by `pachamaltese`. Progress: 16 of 31.
- targets and tarchetypes** (#401) opened 20 days ago by `wlrandau`. Progress: 14 of 31.
- Submission: osmextract** (#395) opened on Aug 12 by `agilas5`. Progress: 13 of 31.
- CoSMoS - does our package fit?** (#390) opened on Jul 21 by `KevinShook`. Progress: 0 of 11.
- Submission: readODS** (#386) opened on Jun 29 by `chainsawriot`. Progress: 11 of 31.

# Review Framework

 **Submission: phonfieldwork** 6/approved package topic:reproducibility  
topic:scientific-software-wrappers topic:workflow-automation

#385 by agricolamz was closed 14 hours ago  13 of 31



35

## Submission: phonfieldwork #385



agricolamz opened this issue on Jun 20 · 35 comments



agricolamz commented on Jun 20 · edited by melvidoni ▾

Member



...

Submitting Author: George Moroz (@agricolamz)  
Repository: <https://github.com/agricolamz/phonfieldwork>  
Version submitted: 0.0.8  
Editor: @melvidoni  
Reviewer 1: @jonkeane  
Reviewer 2: @nikopartanen  
Archive: TBD  
Version accepted: 2020-10-20

<https://github.com/ropensci/software-review/issues/385>

# Step 1: Authors submit package

Open an issue and complete the submission template. It includes choices and short answers about topics like scope:

## Scope

- Please indicate which category or categories from our [package fit policies](#) this package falls under: (Please check an appropriate box below. If you are unsure, we suggest you make a pre-submission inquiry.):

- data retrieval
- data extraction
- data munging
- data deposition
- workflow automation
- version control
- citation management and bibliometrics
- scientific software wrappers
- field and lab reproducibility tools
- database software bindings
- geospatial data
- text analysis

- Explain how and why the package falls under these categories (briefly, 1-2 sentences):

phonfieldowrks is a tool that helps researchers create sound annotation viewer from multiple sounds. I believe that sound annotation viewer helps to make phonetic more reproducible, since it make sound data widely available. It helps reducing the time that annotators spent during annotations in Praat, Elan and EXMARaLDA, so it partially replace its functionality.

# Step 1: Authors submit package

It includes choices and short answers about topics like technical checks:

## Technical checks

Confirm each of the following by checking the box.

- I have read the [guide for authors](#) and [rOpenSci packaging guide](#).

This package:

- does not violate the Terms of Service of any service it interacts with.
- has a CRAN and OSI accepted license.
- contains a [README with instructions for installing the development version](#).
- includes [documentation with examples for all functions, created with roxygen2](#).
- contains a vignette with examples of its essential functions and uses.
- has a [test suite](#).
- has [continuous integration](#), including reporting of test coverage using services such as Travis CI, Coveralls and/or CodeCov.

Other topics include package metadata, publication options, and agreeing to rOpenSci's Code of Conduct.

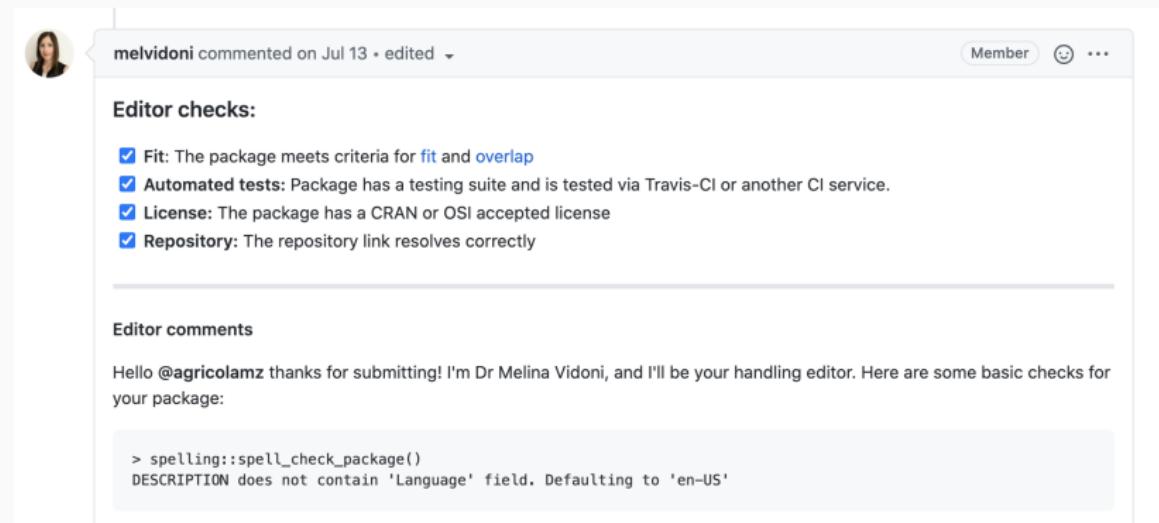
## Step 1: Authors submit package

At this point, the current Editor in Chief might ask for some clarifications, and the editorial board will consider if the package is in scope:

The screenshot shows a GitHub pull request comment. On the left is a circular profile picture of a person. To its right, the username **annakrystalli** is followed by the text "commented on Jun 23". To the right of the comment area are three small icons: a person (Member), a smiley face, and an ellipsis (...). The comment text reads:  
Hello @agricolamz and many thanks for your submission.  
We are discussing whether the package is in scope and need a bit more information.  
• If a package has overlapping functionality with other packages, we require it to demonstrate in the documentation [how it is best in class](#). Could you add a more detailed comparison to the packages you mention in the README so we can evaluate?  
• The package is to be used with data from human subjects and also mentions the publication of data. rOpenSci have specific policies on [Ethics, Data Privacy and Human Subjects Research](#) so we would expect care to be incorporated into package workflows to protect personal data and adhere to what subjects have consented to. Could you elaborate on any steps the package takes to ensure it conforms to these policies? A separate vignette might at some point be useful but a section in the README addressing these issues will be fine for now.

## Step 2: Editor Conducts Some Checks

Once the submission is clarified, an Associate Editor is assigned who runs some checks and assigns two reviewers.



A screenshot of a GitHub pull request comment. The comment is from user **melvidoni** on July 13, with an edit history. It includes a profile picture of a woman with dark hair. To the right are buttons for 'Member', a smiley face, and three dots for more options. The comment text starts with 'Editor checks:' followed by a bulleted list of four items, each with a checked checkbox. Below this is a section titled 'Editor comments' containing a message from the editor. At the bottom, there is a code block showing R package check output.

melvidoni commented on Jul 13 • edited

Member

Editor checks:

- Fit: The package meets criteria for [fit](#) and [overlap](#)
- Automated tests: Package has a testing suite and is tested via Travis-CI or another CI service.
- License: The package has a CRAN or OSI accepted license
- Repository: The repository link resolves correctly

---

Editor comments

Hello @agricolamz thanks for submitting! I'm Dr Melina Vidoni, and I'll be your handling editor. Here are some basic checks for your package:

```
> spelling::spell_check_package()
DESCRIPTION does not contain 'Language' field. Defaulting to 'en-US'
```

This process uses a [editor's template](#) and R packages for checking packages ([spelling](#), [goodpractice](#)).

## Step 3: Reviewers Examine and Comment

Reviewers examine the package and submit their review based on a [review template](#). This includes areas for the reviewer to comment on:

- Potential Conflict of Interest
- Documentation
- Functionality
- Review Comments

# Step 3: Reviewers Examine and Comment

## Package Review

- I have met George Moroz many times. I'm familiar with his work, and have used his another R package lingtypology often in my own work.
- As the reviewer I confirm that there are no [conflicts of interest](#) for me to review this work

### Documentation

The package includes all the following forms of documentation:

- A statement of need clearly stating problems the software is designed to solve and its target audience in README
- Installation instructions: for the development version of package and any non-standard dependencies in README
- Vignette(s) demonstrating major functionality that runs successfully locally
- Function Documentation: for all exported functions in R help
- Examples for all exported functions in R Help that run successfully locally
- Community guidelines including contribution guidelines in the README or CONTRIBUTING, and DESCRIPTION with `URL` , `BugReports` and `Maintainer` (which may be autogenerated via `Authors@R` ).

### Functionality

- Installation: Installation succeeds as documented.
- Functionality: Any functional claims of the software been confirmed.
- Performance: Any performance claims of the software been confirmed.
- Automated tests: Unit tests cover essential functions of the package and a reasonable range of inputs and conditions. All tests pass on the local machine.
- Packaging guidelines: The package conforms to the rOpenSci packaging guidelines

# Step 3: Reviewers Examine and Comment

## Review Comments

Overall this was really interesting to review and I think {phonfieldwork} shows a lot of promise (and already goes along way!) to help make some of the more repetitive tasks involved with phonetic and phonological work much easier and more reliable. I threw a few elan files that I had from my dissertation research at it and was impressed it handled them easily. I even managed to read in a folder of elan files that ultimately had around 166k annotations in ~30 seconds with `eaf_to_df()` (and that was even over network storage, so some of that time is shuffling the files from a server!).

I have a few larger/more sweeping comments and ideas for changes that are more architectural or more broad and then at the bottom I went through the package function by function and have specific comments and suggestions about the code. Small note: I did some of this reviewing in two batches somewhat far apart and there were a few pushes to the github repo between the two, so if the line numbers don't match up or make sense, let me know and I'll look at them again and make sure I have the most up to date ones.

### Large(r) scale comments

Think about making classes+methods for some of the analysis types that get read in

There are a number of places where you detect if an object is a TextGrid (typically by checking that the second element has "TextGrid" in it, examples: `extract_intervals()`, `tier_to_df()`, `create_subannotation()`, `annotate_textgrid()`, `df_to_tier()`, `textgrid_to_df()`, `set_textgrid_names()`). I think it might be beneficial to make a class for TextGrids (S3 is probably sufficient, but there are other class systems in R as well) and then you can use `inherits(object, "TextGrid")` instead of `grep("TextGrid", textgrid[2])`. Additionally seeing that check in a few places, you might also want to make a helper function that either returns the TextGrid object if it is a TextGrid or reads in the TextGrid if it's a path.

## Step 3: Reviewers Examine and Comment

**Functionality:** Any functional claims of the software been confirmed.

```
celsius.to.fahrenheit <-
  function (T.celsius, round = 2)
  {
    T.fahrenheit <- (9/5) * T.celsius + 32
    T.fahrenheit <- round(T.fahrenheit, digits = round)
    return(T.fahrenheit)
  }
```

- Do the functions work?
- Are the algorithms appropriate?
- Are there alternatives that might be better?

## Step 3: Reviewers Examine and Comment

### Documentation: Is documentation clear, thorough, and helpful?

`celsius.to.fahrenheit` {weathermetrics}

R Documentation

Convert from Celsius to Fahrenheit.

#### Description

`celsius.to.fahrenheit` creates a numeric vector of temperatures in Fahrenheit from a numeric vector of temperatures in Celsius.

#### Usage

```
celsius.to.fahrenheit(T.celsius, round = 2)
```

#### Arguments

`T.celsius` Numeric vector of temperatures in Celsius.

`round` An integer indicating the number of decimal places to round the converted value.

#### Value

A numeric vector of temperature values in Fahrenheit.

# Review and Revise as a Conversation

## Readme:

| First find ott ids for a set of names:

What is an ott ?

Example from review of rotl:

<https://github.com/ropensci/software-review/issues/17>

# Review and Revise as a Conversation

## README

What is an ott id?

Thanks for pointing it out. Tried to clarify: see [ropensci/rotl#43](https://github.com/ropensci/rotl#43)

Example from review of `rotl`:

<https://github.com/ropensci/software-review/issues/17>

# Review and Revise as a Conversation

```
✓ 5 README.Rmd [ ]  
.. @ -58,7 +58,10 @@ There are two vignettes:  
58 58  
59 59     ### Get a little bit of the big Open Tree tree  
60 60  
61 - First find ott ids for a set of names:  
61 + Taxonomic names are represented in the Open Tree by numeric identifiers, the  
62 + `ott_ids` (Open Tree Taxonomy identifiers). To extract a portion of a tree from  
63 + the Open Tree, you first need to find `ott_ids` for a set of names using the  
64 + `tnrs_match_names` function:  
62 65  
63 66     ```{r resolve}  
64 67     library(rotl)  
...  
↓
```

Example from review of rotl:

<https://github.com/ropensci/software-review/issues/17>

## Once package is accepted

- “The process is an ongoing conversation until acceptance of the package, with two external reviews as important milestones”



- “Once aboard, your package will continue to receive support from rOpenSci members. You'll retain ownership and control of your package, but we can help with ongoing maintenance issues such as those associated with updates to R and dependencies and CRAN policies.”
- “rOpenSci will promote your package through our webpage, blog, and social media. Packages in our suite also get a documentation website that is automatically built and deployed after each push.”

## How to Join In

---

## How to Join In

- Use the Packages: <https://ropensci.org/packages/>
- Read the Packages Book: <https://devguide.ropensci.org/>
- Read Package Reviews: <https://github.com/ropensci/software-review>
- Volunteer to Review: <https://ropensci.org/onboarding/>
- Submit Your Own Package for Review:  
<https://devguide.ropensci.org/>

# Volunteer to Review

Sam Albers · Toph Allen · Alison Appling · Zebulun Arendsee · Taylor Arnold · Dean Attali · Mara Averick · Suzan Baert · James Balamuta · Vikram Baliga · David Bapst · Joëlle Barido-Sottani · Cale Basaraba · John Baumgartner · Marcus Beck · Gabe Becker · Jason Becker · Dom Bennett · Kenneth Benoit · Aaron Berdanier · Fred Boehm · Carl Boettiger · Ben Bond-Lamberty · Alison Boyer · François Briatte · Eric Brown · Julien Brun · Jenny Bryan · Lorenzo Busetto · Scott Chamberlain · Jennifer Chang · Jorge Cimentada · Chase Clark · Jon Clayden · Will Cornwell · Enrico Crema · Ildiko Czeller · Christophe Dervieux · Amanda Dobbyn · Jasmine Dumas · Remko Duursma · Mark Edmondson · Paul Egeler · Evan Eskew · Manuel Fernandez · Kim Fitter · Rich FitzJohn · Robert Flight · Zachary Foster · Auriel Fournier · Carl Ganz · Duncan Garmonsway · Sharla Gelfand · Monica Gerber · Duncan Gillespie · David Gohel · Laura Graham · Charles Gray · Corinna Gries · Hugo Gruson · Julia Gustavsen · W Kyle Hamilton · Ivan Hanigan · Jeff Hanson · Rayna Harris · Ted Hart · Nujcharee Haswell · Verena Haunschmid · Andrew Heiss · Max Held · Rafael Pilliard Hellwig · Bea Hernandez · Jim Hester · Peter Hickey · Roel M. Hogervorst · Jeff Hollister · Kelly Hondula · Sean Hughes · Brandon Hurr · Najko Jahn · Tamora D James · Max Joseph · Krunoslav Juraic · Soumya Kalra · Zhian N. Kamvar · Michael Kane · Andee Kaplan · Hazel Kavili · Os Keyes · Michael Koontz · Bianca Kramer · Will Landau · Erin LeDell · Thomas Leeper · Sam Levin · Stephanie Locke · Robin Lovelace · Julia Stewart Lowndes · Tim Lucas · Andrew MacDonald · Jesse Maegan · Tristan Mahr · Ben Marwick · Miles McBain · Lucy D'Agostino McGowan · Amelia McNamara · Elaine McVey · Bryce Mecum · Francois Michonneau · Helen Rodgers Miller · Jessica Minnier · Priscilla Minotti · Nichole Monhait · Paula Moraga · Ross Mounce · Lincoln Mullen · Matt Mulvahill · Maria Victoria Munafó · Dillon Niederhut · Rory Nolan · Kari Norman · Jakub Nowosad · Daniel Nüst · Paul Oldham · Samantha Oliver · Jeroen Ooms · Philipp Ottolinger · Mark Padgham · Marina Papadopoulou · Edzer Pebesma · Thomas Lin Pedersen · Lindsay Platt · Nicholas Potter · Etienne Racine · Nistara Randhawa · David Ranzolin · Neal Richardson · Emily Riederer · tyler rinker · Emily Robinson · Julia Romanowska · Xavier Rotllan-Puig · Bob Rudis · Edgar Ruiz · Kent Russel · Francisco Rodriguez Sanchez · Alicia Schep · Clemens Schmid · Marco Sciaiani · Heidi Seibold · Julia Silge · Margaret Siple · Margaret Siple · Peter Slaughter · Mike Smith · Frank Soboczenski · Tuija Sonkkila · Gaurav Sood · Adam Sparks · Joseph Stachelek · Christine Stawitz · Irene Steves · Michael Sumner · Chung-Kai Sun · Sarah Supp · Jason Taylor · Filipe Teixeira · Andy Teucher · Jennifer Thompson · Joe Thorley · Tiffany Timbers · Tim Trice · Ted Underwood · Kevin Ushey · Josef Uyeda · Frans van Dunné · Mauricio Vargas · Remi Vergnon · Claudia Vitolo · Jake Wagner · Ben Ward · Elin Waring · Rachel Warnock · Leah Wasser · Lukas Weber · Karissa Whiting · Stefan Widgren · Anna Willoughby · Luke Winslow · David Winter · Kara Woo · Bruna Wundervald · Lauren Yamane · Taras Zakharko · Hao Zhu · Chava Zibman · Naupaka Zimmerman · Jake Zwart · Lluis · NA · NA · NA