Awesome R package development

awesome Track Awesome List

A curated list of awesome tools to assist R 📦 development.

Note before:

- Not all tools are available on CRAN; some might be available only via GitHub or GitLab.
- Only **tools** helpful for package development are included, and not other resources (like books, talks, presentations, etc.).

Contributing

If you wish to suggest any additional tools, please make a PR request or raise an issue here.

Please note that the `awesome-r-pkgtools` project is released with a Contributor Code of Conduct. By contributing to this project, you agree to abide by its terms.

⚠️⚠️ Note Before ⚠️⚠️

If you are not using **the latest** release of pandoc, please change only the `README.Rmd` document.

Swiss army knives 🛠️

Tools useful across all stages of package development (some of these are meta-packages and their component packages are also included in respective sections for the sake of completeness).

- `{devtools}`
- `{usethis}`
- `{packager}`

- `{pacs}`

Package skeletons 💀

- `{pkgkitten}`
- `{rcompendium}`
- `{golem}` (framework for building shiny applications)
- `{leprechaun}` (leaner framework for building shiny applications)
- `{fusen}` (`{rmarkdown}` -based)
- `{pkgverse}` (for package meta-verse)
- `{metamakr}` (for package meta-verse)

Naming things 🧸

- `{available}` (to check if a package name is available to use)
- `{collidr}` (to check for namespace collisions)

Working with package components

- `{rprojroot}` (accessing files w.r.t. package root directory)
- `{desc}` (manipulating `DESCRIPTION` files)
- `{withr}` (to manage package side effects by safely and temporarily modifying global states)
- `{pkgload}` (to simulate the process of installing and loading a package)
- `{pkgbuild}` (to find tools needed to build packages)

Documentation 📚

Manual

- `{roxygen2}`
- `{Rd2roxygen}` (in case you inherit a project where documentation was not written using `{roxygen2}`)
- `{sinew}` (generate `{roxygen2}` skeletons)
- `{roclang}` (helpers for diffusing content across function documentation)
- `{Rdpack}` (for inserting references, figures, and evaluated examples in Rd docs)
- `{roxygen2md}` (using Markdown syntax in package documentation)
- `{rd2list}` (converts Rd docs to a human-readable list)
- `roxygen2Comment` (Rstudio addin for adding and remove `{roxygen2}` comment)

Math in manual

- `{katex}`
- `{mathjaxr}`

Vignettes

- `{knitr}`
- `{rmarkdown}`
- `{prettydoc}` (creates lightweight yet pretty vignettes)
- `{learnr}` (interactive tutorials)

Website

- `{pkgdown}` (static website for package documentation)
- `{gitdown}` (software changes as a gitbook)
- `{altdoc}` (use docute or docsify to create a static website for package documentation)

Translation

- `{potools}` (for translating messages and checking the “health” of the messaging corpus)

Lifecycle

- `{lifecycle}` (to manage the life cycle of exported functions)

Badges and stickers

- `{badger}`
- `{hexSticker}`
- `hexwall` (to create a wall of hexstickers)

Presentation

- `{xaringan}`

Book

- `{bookdown}`

Documentation quality 🖋️

- `{docreview}` (to check quality of docs)
- `{spelling}` (to spell check)
- `{gramr}` (for grammar suggestions)

Package change tracking and versioning

- `{fledge}`

Unit testing 🧪

- `{testthat}`
- `{vdiffr}` (visual regression testing)
- `{mockthat}` (function mocking for unit testing to third-party packages)
- `{mockr}`
- `{roxytest}` (inline `{testthat}` tests with `{roxygen2}`)
- `{exampletestr}` (tests based on package examples)
- `{shinytest}` (testing Shiny apps)
- `{testdown}` (turn `{testthat}` results into a `{bookdown}` project)
- `{httptest}` (a test environment for HTTP requests in R)
- `{httptest2}` (the same for `{httr2}` package)
- `{autotest}`
- `{tinytest}`
- `{RUnit}`
- `{testit}`
- `{testthis}`
- `{xpectr}`
- `{unitizer}`
- `{r-hedgehog}` (property based testing)

Code analysis 🗃️⏱️

General

- `{codetools}`
- `{goodpractice}` (Swiss army knife for good practices)
- `{pkgcheck}` (checks if package follows good practices recommended for packages in the `rOpenSci` ecosystem)

Code coverage

- `{covr}` (computes code coverage)
- `{covrpage}` (provides summary `README` of code coverage and corresponding tests)

Lint detection

- `{lintr}` (static code analysis)
- `{roger}`
- `{cleanr}`

Code complexity

- `{cyclocomp}` (to index the complexity of a function)

Code similarity

- `{dupree}`
- `{rscc}`
- `{SimilaR}`

Benchmarking and profiling

- `{bench}`

- `{profvis}`

Lines of code

- `{cloc}`

Formatting 🧽

R code

- `{styler}` (especially relevant if you follow `{tidyverse}` style guide)
- `{formatR}`
- `AlignAssign` (RStudio addin)

Markdown documents

- `{stylermd}`

Dependencies ⚖️

- `{deepdep}` (to visualize and explore package dependencies)
- `{itdepends}` (to assess usage, measure weights, visualize proportions, and assist removal of dependencies)
- `{DependenciesGraphs}` (to visualize package dependencies)
- `{pkgnet}`(build a graph representation of a package and its dependencies)
- `{functiondepends}`
- `{pkgndep}` (checks the heaviness of the packages used)
- `{oysteR}` (secure package against insecure dependencies)
- `{attachment}`

CRAN checks, submission, and status 📬

- `{rcmdcheck}` (run `R CMD check` form R programmatically)
- `{rhub}` (to run `R CMD check` on CRAN architectures)
- `{checkhelper}` (submission help)
- `{foghorn}` (check results and submission portal status)
- `{urlchecker}` (URL checks)

Usage 🙈

- `{cranlogs}` (computing CRAN download counts)
- `{packageRank}` (visualizing CRAN download counts)

CI/CD 📟

CI/CD: continuous integration and either continuous delivery or continuous deployment

- `actions` (for GitHub Actions)
- `r-appveyor` (for AppVeyor)
- `{tic}` (for Circle CI and GitHub Actions)
- `{jenkins}` (for Jenkins CI)
- `{cronR}` (to schedule R scripts/processes with the cron scheduler)

Security/Privacy 👮

- `{digest}`
- `{hash}`

- `{gpg}`

Build systems

- `{fakemake}` (to mock Unix Make build system in case it is unavailable)

Validation frameworks

- `{valtools}` (in clinical research and drug development)

Debugging 🔭

- `{debugme}`
- `{debugr}`

Input validation 🔬

- `{checkmate}` (argument checks)
- `{assertthat}`
- `{assertive}`
- `{pkgcond}` (better error messages for package users)
- `{dreamerr}`

Package metadata 🖨️

- `{codemetar}`
- `{cffr}`
- `{pkgapi}`

- `{packagemetrics}` (for comparing among packages)
- `{devtoolbox}`
- `{pkgattrs}`
- `{foreman}`
- `{sessioninfo}`

Reverse dependency checks 📡⚰️

- `{revdepcheck}`
- `{xfun}` (specifically, `xfun::rev_check()`)

Docker container 🛍️

- `{containerit}`
- `{usethat}`

Gratitude 🙏💌

To thank the contributors or maintainers of packages you rely on.

- `{thankr}`
- `{allcontributors}`

Integration with other languages 🔗

C++

- `{Rcpp}`

- `{cpp11}`

- `{memtools}` (to solve memory leaks)

Python

- `{reticulate}`

Rust

- `{hellorust}`

.NET Framework

- `{rClr}`

JavaScript/HTML/CSS

- `{htmltools}`
- `{packer}`

Julia

- `{JuliaCall}`

Sundry 🗒️

- `{TODOr}` (RStudio addin to list things that you need to do or change)

- `{prefixer}` (prefix function with their namespace)

- `{gitignore}` (to fetch gitignore templates)

- `{touchstone}` (benchmarking pull requests)

- `{precommit}` (pre-commit hooks)