Ten Simple Rules for Selecting an R Package

Caroline's ideas and notes for PLOS article

Summer 2020

Brainstorm

- CRAN
- documentation, help files, cheatsheets
- verify the authors' credibility and associations
- publications introducing package (e.g., *The R Journal*)
- percentile on CRAN
- compare number of downloads
- dates and versions; log of updates; latest version; bug fixes; "orphaned" package
- check source code to see if authors follow best practices for package development
- purpose; compare features across multiple packages that accomplish similar tasks
- dependency checks (strength of other packages it relies upon)
- unit tests (for each function to see it is working as expected) and full-coverage within unit tests

Two stages of package selection

1. **Exploration:** How do you find a package?

- Internet searches (. . . in R)
- Conferences
- Social media (#rstats)
- Textbooks
- CRAN task views

2. **Selection:** Once you found some options for packages to accomplish a task, how do you know which one is best to use?

- Features
- Repositories
- Documentation
- Authors
- Unit testing
- Peer review
- Established
- Explore code
- Indicators of package quality: reputation, association, resources and documentation, development, purpose and uses

Visualizations, tables, figures

- Table of repositories

- Top R packages by number of downloads (see Leaderboard in RDocumentation)
- Top R packages by percentile
- Types of documentation
- Roadmap

Audience

- Particular interest in R with applications in computational biology

Questions

- Why are the author superscripts the same in PLOS template?
- Is there an issue with indentation of the first paragraph of the introduction?
- "Open-source" or open source?
- Language in intro (you, we, one, or the user)?
- Should we add more audience-specific language (tasks, problems, decisions)?