The R Documentation Task Force: The Next Generation R Documentation System

Andrew Redd, PhD
University of Utah, School of Medicine

Introduction

The Problem

The documentation system built into core R, hereafter referred to as the Rd system, has changed little over it's life while many other languages and tools have surpassed it's ability to document. The `roxygen2`[Wickham et al., 2015] package provides alternative methods of defining documentation which are translated into Rd. The package enjoys phenomenal popularity with much new code documented in this way. It is time for a rethinking of documentation, how it is created, stored, handled and distributed. There are two aspects of R documentation I intend to address which will make R an exemplary system for documentation.

The first aspect is storage. The mechanism of storing documentation in separate Rd files hinders the development process and ties documentation to the packaging system, and this need not be so. Life does not always follow the ideal; code and data are not always distributed via nice packages. Decoupling the documentation from the packaging system will allow for more dynamic and flexible documentation strategies, while also simplifying the process of transitioning to packages distributed through CRAN or other outlets.

The second aspect is flexibility of defining documentation. R is a language of flexibility and preference. There are many paths to the same outcome in R. While this has often been a source of confusion to new users of R, however it is also one of R's greatest strengths. With packages flexibility has allowed for many contributions, some have fallen in favor while others have proven superior. Adding flexibility in documentation methods will allow for newer, and ideally improved, methods to be developed.

Current State of R Documentation

The Rd system is the default for documenting R code. This system based on the TEXmarkup language is well defined for distinguishing parts, but not very manipulable programmatically, especially when edited for special formatting. The Rd system has been extended with with packages such as Roxygen2[Wickham et al., 2015] and inlinedocs[Hocking et al., 2013], which provides some relief.

The `roxygen2` package, takes inspiration from the Doxygen[van Heesch, 2008]. This allows for placing documentation in special comments along with special tags to denote different types of documentation. For example `@title` denotes that the line gives the documentation title. One area where `roxygen2` falls short is documentation in place, such as documenting function arguments next to the declaration of the argument. The `inlinedocs` package helps with this however it is not tied into the far more often used `roxygen2` system causing fragmentation. Both these systems are augments to the Rd system, which is fundamentally limited in it's aspirations.

One system that we can take advice from is the `knitr` package [Xie, 2016, 2015, 2014]. This package has changed the literate programming paradigm for R, changing the standard for documents from Sweave using tex to Rmarkdown using the more friendly language of Markdown or CommonMark[MacFarlane, 2016].

Who should care

All R users that both create and consume functions, packages, classes, and data should care as all these objects should have documentation. Within organizations this can be particularly poignant, as code is rarely packaged but nearly always required to be documented. My proposal opens options for these organizations. Since essentially all R users have a stake in documentation it will be important that any solution to the problem consider the many varied voices in the R community.

Proposed Solution

Overview

I propose creating an R Documentation Task Force with the purpose of completing the aims outlined below.

Aim 1

> We will abstract, document and implement in R a representation for documentation of R objects; functions, classes, datasets, etcetera.

The core of the proposal is to define a set of documentation objects that can be stored internal to R, and manipulated programmatically. Currently the Rd parser produces an internal format that is tied closely to the structure of the .Rd source. In this project we will abstract, document, and extend the structure of the internal format so that it becomes the primary target of programming effort.

Part of the abstraction effort will be defining relationships between documentation types and restrictions on documentation. Restrictions will be graded according to severity such that violations can be set to produce messages, warnings or errors. The restriction level can be adjusted according to need and circumstance. For example, what may be a warning in an interactive session, would be elevated to an error when building documentation intended to go into a package.

Aim 2

> The Task Force will make recommendations for methods, in addition to the Rd format, to be supported for converting into internal documentation objects. Recommendations will also be made for supported formats to convert to directly.

The Task force will decide what languages will be supported for conversion to and from the internal storage format. We will consider methods such as Roxygen and Roxygen extended to include inline comments, for supported methods. We will evaluate languages such as Markdown/CommonMark to evaluate if there is sufficient rigor to form an input method or if portions should be integrated into other methods, such as if markdown present in comments should be parsed and converted.

Allowing documentation from multiple sources such as in comments to be converted to the new internal content-focused storage mechanism, raises these systems, such as Roxygen, to the same status as Rd documentation, since the intermediate step of converting to Rd will no longer be needed. Allowing users multiple methods of defining documentation, provides flexibility while also maintaining backwards compatibility with existing systems. Flexibility in turn allows programmers to document in the style they are most comfortable with, which makes documentation easier and more likely to be completed. Adding documentation capabilities present in other languages will put R at the forefront of documentation of code as it is now a leader in literate programming with knitr.

Aim 3

> We will impliment the abstraction of documentation according to the results of Aim 1. According to the recommendations resulting from aim 2, we will implement methods to convert from supported sources of documentation to newly define documentation objects, and from the internal representation to the supported output formats.

The system we propose to implement will allow for multiple inputs and outputs with the internal representation at the center. This will bring all the recommendations of the Documentation Task Force to the programmer for use. The system will provide feedback on the documentation prior to building packages that will assist in faster development.

Operation of the Task Force

To ensure that all voices with a stake in documentation are heard the announcement of the project will include an invitation to those interested to participate. According to the interest shown final members may need to be selected. Any member of the R Core Team, wishing to participate will be given a seat. R Core Team members that have already indicated their intention to participate are listed later in the proposal. Additionally, any individual representing an R Consortium member will be given a seat. Should interest outside of the R Core Team and R Consortium members exceed reasonable limits for a task force, interested individuals will be selected based on experience in R development and code documentation in consultation with the R Core Team members participating on the Task Force, while trying to maximize the coverage of experience in different areas.

In addition to myself Duncan Murdoch of the R Core Team has agreed to participate on the task force and included a letter of support. Michael Lawrence of the R Core Team and the R Journal, as well as others from the R community at large, have expressed interest in participating.

The task force will work on the principle of consensus, seeking a solution that all can agree address their concerns[Fisher et al., 2011]. If there are issues with incompatible solutions where a consensus cannot be reached the task force will seek outside comments through blog posts, online polls, and editorials in the R Journal.

Potential Benefits

The are many potential benefits to having an internal documentation system. The idea of documentation that can be changes via code fits very well in the meta-programming ideology of R. While not all benefits can be foreseen, some can. These examples are not under the scope of this project, but given as motivation why a more programmatic approach to documentation is desirable.

For example, at the 2016 UseR! conference there were two systems presented[ass, 2016, che, 2016] for run-time parameter checks. Additionally the `assertthat` package also does run-time parameter checks. It is common that significant code is dedicated to checking of arguments for validity. Using documentation available in a programmatic sense that includes information such as allowed classes, length restrictions, and range or value restrictions one could create a system that checks all function parameters against all documented restrictions in one call. Since the checks are designed off structured documentation there would be little worry that the restrictions would not be documented.

Another example and benefit to tool developers such as Rstudio is that documentation could be queried for information to provide auto-completion and IntelliSense style help.

I am proposing decoupling documentation from the packaging system. For organizations that use R internally but do not distribute code as packages this will allow for documentation of all R work shared withing the organization. Even single files passed around can include documentation that can be extracted and utilized

for the benefit of the recipient. Additionally, large development shops can utilize design tools that create documentation which could be imported into R.

Prior Work

At UseR! 2016 international R users conference I presented work I had done in the direction of extending the R documentation engine. The work is contained in the `inlinedoc` branch of the `lint` package available on github[Redd]. This implementation has facilities for in context documentation of function, a class system for representing documentation, the ability to extract documentation from source files, methods for printing documentation in either Rd or markdown formats, and classes for attaching documentation to functions that assist in printing documentation to the console. This work has not been released to CRAN because it is incomplete. The work provides an important proof of concept, however decisions were made unilaterally. The Task Force will first seek to create an abstraction of documentation needed for R objects, then create an implementation of that abstraction.

Considerations for the Task Force

The following is a non-exhaustive list of issues that the documentation task force would provide input for.

Class system

There are several class systems in place in R: S3, S4, Reference Classes , and R6 [R Core Team, 2016, Chang, 2016]. Each provides advantages and disadvantages. The test force will need to decide on the important features for storing documentation and ultimately decide on the Class system that will be used to internally represent documentation inside R.

Storage mechanisms

In lint I attach documentation directly to the function or object to which the documentation pertains. When a package is built and loaded the attributes are stripped from functions, which would negate the benefit of attaching documentation directly. An alternative would be similar to the class system where there is a central repository of documentation that is called on as necessary. This has the disadvantage of problems from renaming and could be seen as going against the principle of proximity. For either solution for storing documentation there are many considerations that should be accounted for: performance, memory usage, and loading times, just to name a few.

Appropriate methods of documentation

Consider the problem with in context documentation. Many programs are written as follows:

```
hello_world <-
function( greeting  = 'hello',
          recipient = 'world'
        ){...}
```

however how is one to interpret the documentation when added:

```
hello_world <-
function( greeting  = 'hello', #< The greeting
          recipient = 'world' #< Who to
        ){...}
```

Should this be allowed, as it is in Doxygen, with the documentation for the greeting be after the comma or should this be an error with recipient having two documentation lines as the R parser interprets the `#< The greeting` to belong to the parent expression that contains recipient. The current implementation in lint circumvents the problem by placing the comma at the beginning of the line.

```
hello_world <-
function( greeting  = 'hello' #< The greeting
        , recipient = 'world' #< Who to
        ){...}
```

No user should be expected to be forced into putting commas at the beginning of a line, yet behavior consistent with the R parser and language definition is also desired.

Dissemination

All work produced by the documentation task force will be available online and to the public. The R Documentation Task Force will create a github account for the development of code, and storage of meeting notes. All tools and packages created will be licensed under the GPL Version 2, the same license base R uses. Documentation, such as meeting notes and design diagrams, will be licensed under a creative commons attribution and share alike license.

The wider R community will be made aware of the documentation task force via blog post to the R-Consortium blog. I will also announce the project on the R-announce, R-package-devel, and R-devel mailing lists. An article summarizing the recommendations of the documentation task force and developed tools will be submitted to the R Journal. The R Journal does not publish on a time table that would be amenable to announcement of the project.

Timeline

- *Mid-August 2016* notification of approval.
- *September 1, 2016* Kickoff for the R Documentation Task Force with final members.
- *September 16, 2016* Deadline for submitting posts to the R-consortium blog, the R-announce, R-package-devel, and R-devel mailing lists, announcing the project.
- *September 1 through November 27th 2016* The task force conducts bi-weekly meetings via Lync to address issues in documentation.
- *November 27th, 2016* Deadline for preliminary recommendations of documentation extensions. Recommendations and conflicts written up and submitted to the R journal to be published in the December 2016 issue.
- *December 2016* Posts made to the R Consortium blog, and R mailing lists to coincide with the R Journal article to call for public participation.
- *January 27, 2017* Deadline for general comments on recommendations. Work begins to finalize new documentation system.
- *February 2017* Task force meets to finalize decisions after public input.
- *February-May 2017* Task force meets monthly as necessary to monitor progress on code development.
- *May 2017* Article is submitted outlining final recommendations and the subsequent tools developed to the R Journal for review targeting the June 2017 issue.

- *July 4-7, 2017* Developments will be presented at the International R users conference in Brussels, Belgium.

How the R Consortium can help

By sponsoring this project the R Consortium will give much needed attention to the project. Additionally the consortium has access to members of the consortium sponsors, and contacts with the R community at large to ensure that the interests are represented. Recommendations of additional members to include on the task force is appreciated. I am asking for funding to cover salary for the programming work to implement the system as well as travel to the 2017 R users conference to presenting the recommendations and the system.

Budget

I am requesting $10,000 of funding for the following reasons:

- $3750 for travel to R Users Conference 2017 in Brussels, Belgium to present the recommendations and system as outlined above, and
- $6250 in salary support for programming the internal representation of documentation and converts.

Work hours dedicated to the administration of the R Documentation Task Force will be supported by the University of Utah Population Health Research Foundation for Discovery (letter of support included).

References

Run-time testing using assertive. Presented at the 2016 International R users Conference, Stanford University, CA, 2016.

Checkmate: Fast and versitile argument checks. Presented at the 2016 International R users Conference, Stanford University, CA, 2016.

Winston Chang. *R6: Classes with Reference Semantics*, 2016. URL https://CRAN.R-project.org/package=R6. R package version 2.1.2.

Roger Fisher, William L Ury, and Bruce Patton. *Getting to yes: Negotiating agreement without giving in*. Penguin, 2011.

Toby Dylan Hocking, Thomas Wutzler, Keith Ponting, and Philippe Grosjean. Sustainable, extensible documentation generation using inlinedocs. *Journal of Statistical Software*, 54(6):1–20, 2013. URL http://www.jstatsoft.org/v54/i06/.

John MacFarlane. *CommonMark Spec*, 2016. URL http://spec.commonmark.org/0.25.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016. URL https://www.R-project.org/.

Andrew Redd. halpo/lint at inlinedoc. URL https://github.com/halpo/lint/tree/inlinedoc.

Dimitri van Heesch. Doxygen: Source code documentation generator tool. *URL: http://www. doxygen. org*, 2008.

Hadley Wickham, Peter Danenberg, and Manuel Eugster. *roxygen2: In-Source Documentation for R*, 2015. URL https://CRAN.R-project.org/package=roxygen2. R package version 5.0.1.

Yihui Xie. knitr: A comprehensive tool for reproducible research in R. In Victoria Stodden, Friedrich Leisch, and Roger D. Peng, editors, *Implementing Reproducible Computational Research*. Chapman and Hall/CRC, 2014. URL http://www.crcpress.com/product/isbn/9781466561595. ISBN 978-1466561595.

Yihui Xie. *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition, 2015. URL http://yihui.name/knitr/. ISBN 978-1498716963.

Yihui Xie. *knitr: A General-Purpose Package for Dynamic Report Generation in R*, 2016. URL http://yihui.name/knitr/. R package version 1.13.

Dept. of Stat. and Act. Sci.
University of Western Ontario

July 9, 2016

The Infrastructure Steering Committee
The R Consortium

Dear committee members:

I am writing this letter in support of Andrew Redd's proposal for the R Documentation Task Force.

I have worked on R's documentation for many years. When I wrote our current parser in 2008, one idea was that it would allow automatic translation of .Rd files into a better format, easier for authors and for programmatic transformations. I believe the approach we've taken with vignettes (allowing multiple input formats) is better, and I believe this project will lead to similar flexibility for help files.

Sincerely,

Duncan Murdoch

UNIVERSITY OF UTAH
SCHOOL OF MEDICINE

July 8, 2016

Infrastructure Steering Committee
R Consortium

Dear Members of the Infrastructure Steering Committee,

I am delighted to express the full support of the Population Health Research Foundation for Discovery (PHR) of the Utah Center for Clinical and Translational Science for Dr. Andrew M. Redd's proposal "The R Documentation Task Force: The Next Generation R Documentation System." This project, seeks to make improvements to the documentation systems available to R programmers and users.

The PHR consists of five cores, including the Study Design and Biostatistics Center (SDBC) and the Health Measurement and Survey Methods (HMSM) Core, and includes more than 40 members many of whom are regular R users. The SDBC supports the Utah R Users group which Dr. Redd has lead for the last 6 years. Dr. Redd also leads the SDBC Internal Standards Committee tasked with defining standard operating procedure requirements such as documenting functions. His proposed work promises to provide important tools that relate to the standards of high quality work to which PHR members adhere and aligns with the primary mission of the PHR, which is to support clinical and translational research at the University of Utah.

The PHR expresses our support for the aims in Dr. Redd's proposal. Should the proposal be funded the PHR is willing to support Dr. Redd's time spent in Administration of the R Documentation Task Force and consider it as service to the PHR and SDBC.

Sincerely,

Tom Greene

Tom Greene, Ph.D.
Chief, Division of Biostatistics, Department of Population Health Science
Professor, Department of Internal Medicine
Director, Population Health Research Foundation in the University of Utah Center for Clinical and Translational Science
University of Utah

Department of Population Health Science
Study Design and Biostatistics Center
Center for Clinical and Translational Science
295 Chipeta Way
Salt Lake City, UT 84132
Phone (801) 585-6667
Fax (801) 581-3623