

Documentation & namespaces

Hadley Wickham

Assistant Professor / Dobelman Family Junior Chair
Department of Statistics / Rice University

June 2011



Tuesday, June 7, 2011

1. Package documentation
2. Function documentation
3. Namespaces
4. Text formatting

Package level

Package-level

README

NEWS

Package help topic

Vignettes + CITATION

Demos

<https://github.com/hadley/devtools/wiki/docs-package>

README

Should outline the basics of your package: what does it do, and why should people care?

You'll also use this when you email out package release announcements.

Particularly important if your package is on github.

`plyr, lubridate, stringr`

NEWS

Whenever you make a change to your package, make sure to make a note of it in the NEWS file.

R has a special format for NEWS, but it's poorly documented. Use `show_news()` to check it.

Use in release announcements.

ggplot2 0.9.0 **Version number**

MINOR CHANGES **Subheadings (optional)**

- * `'geom_text'` now supports `'fontfamily'`, `'fontface'`, and `'lineheight'` aesthetics for finer control over text display. (Thanks to Kohske Takahashi for the patch. Fixes #60)
- * `'collide'`, which powers `'position_dodge'` and `'position_stack'`, now does not error on single x values (Thanks to Brian Diggs for a fix. #157)
- * When printing a ggplot2 object, the rendered plot information is returned invisibly. You can capture this with (e.g.) `'x <- print(qplot(mpg, wt, data = mtcars))'` and in the future will be able to use it to get information about the plot computations, such as the range of all the scales, and the exact data that is plotted. **Individual items indented and start with ***
- * `'scale_shape'` finally returns an error when you try and use it with a continuous variable

DEVELOPMENT

- * ggplot2 has moved away from the two (!!) homegrown documentation systems that it previously relied on, and now uses roxygen extensively. The current downside is that this means that ggplot2 website can no longer be updated, but I hope work with the `'helpR'` package will resolve that shortly.

Package help topic

You should be able to get help on the package from within R using `?mypackage` or `package?mypackage`.

This is created in the same way as function level documentation so we'll discuss in the next section.

Vignettes

A vignette is a long-form document that introduces new users to your package.

If you write a nice one, it's worth publishing in JSS or the R-journal.

When published, use a CITATION file so users know what to cite.

I am bad at writing vignettes.

Your turn

Download the source code for the `coin` package.

How many vignettes are there?

How are they different from ordinary latex files?

Vignettes

Must be written in Sweave

<http://cran.r-project.org/doc/manuals/R-exts.html#Writing-package-vignettes>

<http://www.statistik.lmu.de/~leisch/Sweave/Sweave-Rnews-2003-2.pdf>

Demos

Long-form R scripts that show how all the pieces of your package fit together.

Function level

Roxygen

- Essential for function level documentation - huge time saver
- Converts source comments into the Rd files needed for R packages
- But current version on CRAN is slow and buggy.


```
\name{arrange}
\alias{arrange}
\title{Order a data frame by its columns.}
\usage{arrange(df, ...)}

\description{
  Order a data frame by its columns.
}

\details{
  This function completes the subsetting, transforming and
  ordering triad with a function that works in a similar
  way to \link{subset} and \link{transform}
  but for reordering a data frame by its columns. This
  saves a lot of typing!
}

\keyword{manip}
\arguments{
  \item{df}{data frame to reorder}
  \item{...}{expressions evaluated in the context of df and then fed
to \link{order}}}
}

\examples{mtcars[with(mtcars, order(cyl, disp)), ]
arrange(mtcars, cyl, disp)
arrange(mtcars, cyl, desc(disp))}
```

Tag	Purpose
@param	Describe inputs
@examples	Show how the function works
@author	Who wrote the function (if different from package)
@seealso	Pointers to related functions
@return	Describe outputs
@alias	Make it easier for users to find
@name	If would otherwise create invalid filename

Getting started

Install roxygen from <https://github.com/hadley/roxygen>

My fork fixes a few bugs and is much much faster. Will hopefully merge back into CRAN roxygen later this summer.

Your turn

Familiarise yourself with the roxygen documentation for this package.

Try making a few small changes to hof-doc then rerunning `document()`. How do the Rd files change?

Install the package and read the documentation. What works well? What could I have done better?

Documentation cycle

1. Update roxygen comments.
2. Re-run roxygen.
3. Install package.
4. Inspect documentation.

```
document("hof-doc")
check_doc("hof-doc")
install("hof-doc") # experimental
reload("hof-doc") # experimental
?hof
```

Your turn

Add a package help topic to hof-doc, following the description in <https://github.com/hadley/devtools/wiki/docs-package> and using other packages as needed for reference.

Re-run `document()`, install and check.

```
Ask <- function(f, x, ...) {  
  ind <- vapply(x, f, logical(1), ...)  
  na.false(ind)  
}
```

```
na.false <- function(x) {  
  !is.na(x) & x  
}
```



```
elements <- list(1:10, c(-1, 10), c(TRUE, FALSE),
  letters)
results <- lapply(elements, log)
results <- lapply(elements, function(x) try(log(x)))

is.error <- function(x) inherits(x, "try-error")

# Just to get the successful ones
successful <- Filter(Negate(is.error), results)

# Which ones failed?
failures <- vapply(results, is.error, logical(1))
failures <- Ask(is.error, results)
```

Your turn

In hof-doc, document the Ask function.

Re-run document(), re-install, re-load and check.

Namespaces

Motivation

- What happens if two packages both have a function with the same name?
- Namespaces provide a way to resolve this issue, and to reduce it.
- Splits functions into internal and external

Internal	External
Only for use within package	For use by others
Documentation optional	Must be documented
Easily changed	Changing will break other peoples code

Exporting

- `@export` adds a function to the namespace
- `@S3method` adds a `S3method` to the namespace (needed even if you don't export the generic)

Importing

- Add package to Imports in DESCRIPTION
- Add @imports package to your package documentation
- Maximises chances your package will not interfere with others.

load_all()

- Ignores namespaces
- Probably what you want for development
- But can hide namespace problems until you run R CMD check

This work is licensed under the Creative Commons Attribution-Noncommercial 3.0 United States License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/3.0/us/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.