Rd2roxygen: Convert Rd to **roxygen** documentation and utilities to enhance R documentation

Yihui Xie*

December 6, 2013

The package **Rd2roxygen** (Wickham and Xie, 2013) helps R package developers who used to write R documentation in the raw LaTeX-like commands but now want to switch their documentation to **roxygen2** (Wickham *et al.*, 2013), which is a convenient tool for developers, since we can write documentation as inline comments, e.g.

```
## the source code of the function `parse_and_save`
ex.file = system.file("examples", "parse_and_save.R", package = "Rd2roxygen")
cat(readLines(ex.file), sep = "\n")

##' Parse the input Rd file and save the roxygen documentation into a file.
##'
##' @param path the path of the Rd file
##' @param file the path to save the roxygen documentation
##' @param usage logical: whether to include the usage section in the output
##' @return a character vector if \code{file} is not specified, or write the vector
##' into a file
##' @export
##' @author Hadley Wickham; modified by Yihui Xie <\url{http://yihui.name}>
parse_and_save <- function(path, file, usage = FALSE) {
  parsed <- parse_file(path)
  output <- create_roxygen(parsed, usage = usage)
  if (missing(file)) output else
cat(paste(output, collapse = "\n"), file = file)
}
```

With **roxygen2** (typically using *roxygenize()*), we can create the real Rd file from the above source code like this:

```
rd.file = system.file("examples", "parse_and_save.Rd", package = "Rd2roxygen")
cat(readLines(rd.file), sep = "\n")

\name{parse_and_save}
\alias{parse_and_save}
\title{Parse the input Rd file and save the roxygen documentation into a file.}
\usage{parse_and_save(path, file, usage=FALSE)}
\description{Parse the input Rd file and save the roxygen documentation into a file.}
\value{a character vector if \code{file} is not specified, or write the vector
into a file}
```

*Department of Statistics, Iowa State University. Email: xie@yihui.name

```
\author{Hadley Wickham; modified by Yihui Xie <\url{http://yihui.name}>}
\arguments{\item{path}{the path of the Rd file}
\item{file}{the path to save the roxygen documentation}
\item{usage}{logical: whether to include the usage section in the output}}
```

The **Rd2roxygen** package goes exactly in the *opposite* way – it parses the Rd files and turns them back to roxygen comments. We can either do this job on single Rd files, or just convert the whole package. The latter might be more useful for developers who are considering the transition from Rd to roxygen.

1 Convert a whole package

The function *Rd2roxygen()* can take a path of a source package, parse all the Rd files under the `man` directory, and write the roxygen comments right above the source code of the functions under the `R` directory. See `?Rd2roxygen` for an example.

```
library(Rd2roxygen)
args(Rd2roxygen)

function (pkg, nomatch, usage = FALSE)
NULL

## e.g. Rd2roxygen('somewhere/to/source/pkg') there must be 'man' and 'R'
## directories under this path
```

2 Parse a single Rd file

We can parse a single Rd file and create the roxygen comments as well with *parse_file()* and *create_roxygen()*, e.g.:

```
## we can specify the roxygen comments prefix (#' by default)
options(roxygen.comment = "##' ")
(info = parse_file(rd.file))

$title
[1] "Parse the input Rd file and save the roxygen documentation into a file."

$usage
[1] "parse_and_save(path, file, usage=FALSE)"

$desc
[1] "Parse the input Rd file and save the roxygen documentation into a file."

$section
character(0)

$value
[1] "a character vector if \\code{file} is not specified, or write the vector\ninto a file"
```

```
$author
[1] "Hadley Wickham; modified by Yihui Xie <\\url{http://yihui.name}>"

$name
[1] "parse_and_save"

$keywords
list()

$params
[1] "path the path of the Rd file"
[2] "file the path to save the roxygen documentation"
[3] "usage logical: whether to include the usage section in the output"
```

```
cat(create_roxygen(info), sep = "\n")  # parse_and_save() combines these two steps
```

```
##' Parse the input Rd file and save the roxygen documentation into a file.
##'
##' Parse the input Rd file and save the roxygen documentation into a file.
##'
##'
##' @param path the path of the Rd file
##' @param file the path to save the roxygen documentation
##' @param usage logical: whether to include the usage section in the output
##' @return a character vector if \code{file} is not specified, or write the
##' vector into a file
##' @author Hadley Wickham; modified by Yihui Xie <\url{http://yihui.name}>
```

3 Roxygenize and build a package

This package also provides a tool *roxygen_and_build()* (or in short *rab()*) to help us build the package.

```
rab(pkg, build = TRUE, build.opts = "--no-manual", install = FALSE, check = FALSE,
    check.opts = "--as-cran --no-manual", remove.check = TRUE, reformat = TRUE, ...)
```

The main feature of *rab()* is the option to "reformat" the code in the usage and example sections. If we specify `reformat = TRUE` in *rab()*, the code will be reformated like this:

```
## original code
rab=function(pkg,build=TRUE,install=FALSE,
check=FALSE,check.opts='',remove.check=TRUE,reformat=TRUE,...){}
```

```
## the reformatted code; note the spaces and indent
rab = function(pkg, build = TRUE, install = FALSE, check = FALSE, check.opts = "",
    remove.check = TRUE, reformat = TRUE, ...) {
}
```

Note this functionality depends on the package **formatR** (Xie, 2013a), and sometimes it might be not be possible to reformat the code, e.g. the `\dontrun{}` command in Rd can contain arbitrary texts, which means there could be illegal R expressions and **formatR** will be unable to format the code. In this case, the original code will not be reformatted and a message will be printed on screen.

About this vignette

You might be curious about how this vignette was generated, because it looks different from other Sweave-based vignettes. The answer is **knitr** (Xie, 2013b). The real vignette is in LyX, which can be found at `https://github.com/yihui/Rd2roxygen/tree/master/vignettes`, and the corresponding Rnw source is here:

```
system.file("doc", "Rd2roxygen.Rnw", package = "Rd2roxygen")
```

Instructions on how to use **knitr** with LyX can be found at `http://yihui.name/knitr/demo/lyx/`.

References

Wickham H, Danenberg P, Eugster M (2013). *roxygen2: In-source documentation for R*. R package version 3.0.0.

Wickham H, Xie Y (2013). *Rd2roxygen: Convert Rd to roxygen documentation and utilities to improve documentation*. R package version 1.4, URL `http://yihui.name/Rd2roxygen`.

Xie Y (2013a). *formatR: Format R Code Automatically*. R package version 0.10.3, URL `http://yihui.name/formatR`.

Xie Y (2013b). *knitr: A general-purpose package for dynamic report generation in R*. R package version 1.5.14, URL `http://yihui.name/knitr/`.