Building R Package in RStudio Using roxygen2

Shuai Wang

Saturday, May 30, 2015

Creating a Package

We are going to write a R package called `topten` to build a prediction model for a high dimensional data set using just the top ten predictor (http://goo.gl/Ro8R3J). In RStudio, click "File" -> "new project" -> "New Directory" -> "R package", then follow instructions. After it, folders and files are created for you, and you need fill in the contents.

Filling in Contents

First let us fill in the DESCRIPTION file. Everything is straight forward, and you may choose any open source license, like "GPL-3" as your license. Then write your R codes (we will skip it).

For the documentation and NAMESPACE in our package, we're going to use `roxygen2` package. It allows you to put all the documentation in the code file itself. What the `roxygen2` package does is that it strips out the documentation that you put in the code file and it makes the `man` pages, and it formats it in the appropriate way. So you don't have to worry about writing out separate documentation files. There are two nice things about this. One is that it keeps you focused on one file. You don't have to constantly switch back and forth. And the other thing is that since the documentation is actually physically close to the code, there's a better chance that the documentation will stay up to date because you'll be able to see if there are any discrepancies between the documentation and the code itself.

Now write your documentation in R code file with leading `#'`, and RStudio will insert `#'` when you do carriage return (all of the following are written after `#'`). Use `@param` before describing parameters, `@return` before describing return values, `@examples` before examples, `@author` before author, `@details` before details, `@seealso` before the reference to other R functions (functions are specified in the content using `\code{}`, e.g. `\code{lm}` to specify `lm()` function). You can write to the next line for a paragraph if you have many to say. Remember to write `@export` (leave the content blank to export the function following) and `@importFrom` (e.g. `@importFrom stats lm` when you need lm() in your package). Write each documentation before the corresponding function in the R code file.

Building and Checking Package

Click "Build & Reload" in the "Build" tab (it only appears when you created a package), then R session will restart and your package will be loaded (RStudio does not close in this process). Click "More" -> "Configure Build Tools", check "Generate documentation with Roxygen" and click "Configure", and check "Rd files", "NAMESPACE file" under "Use roxygen to generate", and check "R CMD check", "Source and binary package builds", "Build & Reload" under "Automatically roxygenize when running". Delete the documentation file generated automatically when you created the package. Click "Build & Reload" in the "Build" tab again, and the documentation files are generated for you by Roxygen.

You can now check your package. Type in the console `library(help=topten)`, and the DESCRIPTION file of your `topten` package will be displayed for you. Type in your function name, and the console will print out your code. Use `?` followed by your function name, and the corresponding help file will be displayed. If you want to check to see if your package passes R CMD check (it has been automatically done by Roxygen), click "Check" in the "Build" tab, and it will build the package and run R CMD check.