

Introduction to building R packages

Irina Gaynanova

10/8/2019

Useful R package references

- ▶ **R Packages** by H.Wickham
- ▶ **Writing R Extensions** - an official CRAN guide, not very friendly
- ▶ **Hilary Parker's blog** -very short
- ▶ **Fong Chun Chan's Blog** - very short

Many other resources.

R package in Rstudio

Rstudio provides a convenient environment for R package development

Prerequisites:

1. GNU software development tools including a C/C++ compiler
2. LaTeX for building R manuals and vignettes.

You may build a package without either 1 or 2, however you may experience troubles later on depending on what you want to do.

Click [here](#) for advice on how to install these for your system.

R package in Rstudio

You can use graphical interface in Rstudio to initiate a new package.

File — > New Project — > New Directory — > R package

There are additional options such as **R package with Rcpp**, these are useful if you think you will use some C++ code in your package (will discuss later), but can be added later on.

We instead will use R command line utilizing **usethis** R package.

R package and usethis

usethis is an R package that automates a lot of repetitive tasks associated with R package development

Please install the package on your computer following

```
install.packages("usethis")
```

You can also install the development version following

```
devtools::install_github("r-lib/usethis")
```

Creating R package skeleton with usethis

- ▶ Decide the path where you want to store the package, for example

```
path <- getwd() # current working directory  
# or explicit path  
path <- "/Users/Irina/Documents/Teaching/TAMU/Computing"
```

- ▶ Decide on the package name, i.e. **MyPackageTest1**
- ▶ Use **file.path()** and **create_package()** functions from usethis

```
fullpath <- file.path(path, "MyPackageTest1")  
create_package(fullpath)
```

Perform these steps on your machine. What do you see in the output?

create_package() in usethis

The function created an empty R package skeleton, and an Rstudio project associated with it.

- ▶ **R/** - this is the folder that will store all the R functions for the package
- ▶ **man/** - this is the folder that will store all the documentation (BEST to generate automatically, more on this later)
- ▶ **DESCRIPTION** - basic information about the package
- ▶ **NAMESPACE** - tells which function are imported and exported by the package (BEST to generate automatically, more on this later)
- ▶ You have automatically a new Rstudio window opened that corresponds to your Rpackage project

Version control setup

You still need to set up version control with your package, i.e. `git init` from the terminal window.

It's a good time to make initial commit with everything that has been just created.

DESCRIPTION

- ▶ **DESCRIPTION** - basic information about the package

Open the DESCRIPTION file that was created. What does it have?

Open the [CRAN description](#) of **glmnet** R package.