R Package Development Tools

VP Nagraj

October 16 2017


devtools

objectives

use cases

what is an r package?

basic structure of an r package

documentation

unit testing

building

checking

release

best practices

credits

R packages (Hadley Wickham)

http://r-pkgs.had.co.nz/

Write your own R package (Jenny Bryan)

http://stat545.com/packages00_index.html

R package primer (Karl Broman)

http://kbroman.org/pkg_primer/

Writing an R package from scratch (Hilary Parker)

https://hilaryparker.com/2014/04/29/writing-an-r-package-from-scratch/

Writing R Extensions (R Project Team)

https://cran.r-project.org/doc/manuals/r-release/R-exts.html

at this very moment there are ...

11566 packages on CRAN

https://cran.r-project.org/

why?

Secure https://rmflight.github.io/posts/2014/07/analyses_as_packages.html

Deciphering life: One bit at a time

Robert M Flight's home on the web

Home About Blog Archive Blog Source Github Repos PubmedCommons Comments CV

Analyses as Packages

TL;DR

Instead of writing an analysis as a single or set of `R` scripts, use a `package` and include the analysis as a `vignette` of the package. Read below for the why, the how is in the next post.

Analyses and Reports

As data science or statistical researchers, we tend to do a lot of analyses, whether for our own research or as part of a collaboration, or even for supervisors depending on where we work. As I have continued working in `R`, I have progressed from having a simple `.R` script (or collection of related scripts) to using a package to structure as much of my research as possible, including analyses that generate reports.

Note that I have been meaning to write this post for a while, but the tipping point was seeing these tweets from @Hilary Parker and @David Nusinow

Hilary Parker
@hspter
Follow

I am all about the many short scripts rather than one long script when doing an analysis. I think I am alone here. #rstats

11:55 AM - Jul 15, 2014

28 2 13

what is a package?

library() vs require()

```r
library(foo)
require(foo)
```

library vs package

```r
.libPaths()
.Library()
installed.packages()
```

source vs binary

```r
install.packages("PATH/foo.tar.gz", repos = NULL, type = "source")
install.packages("foo")
```

https://www.rstudio.com/wp-content/uploads/2015/06/devtools-cheatsheet.pdf

```
Package
  DESCRIPTION
  R/
  tests/
  man/
  vignettes/
  data/
  NAMESPACE
```

- Setup
- Write code
- Test
- Document
- Teach
- Add data
- Organize

R/

```r
#' Say hello
#'
#' @param name specify what you would like the program to call you; d
#'
#' @export
#' @examples
#' hello()
#' hello("hal")

hello <- function(name = "user") {

  msg <- paste0("hello ", name, " ...")

  cat(msg)

}
```

NAMESPACE

```
# Generated by roxygen2: do not edit by hand

export(hello)
```

DESCRIPTION

```
Package: scratchr
Type: Package
Title: miscellaneous functions to demo package development
Version: 0.1.0
Author: VP Nagraj
Maintainer: vpnagraj <vpnagraj@virginia.edu>
Description: This package contains several functions that were writte
License: GPL-3
Encoding: UTF-8
LazyData: true
RoxygenNote: 6.0.1.9000
```

R/

```r
#' Say hello
#'
#' @param name specify what you would like the program to call you; c
#' @param animal this argument allows you to choose an animal to acc
#'
#' @export
#' @examples
#' hello()
#' hello("hal")

hello <- function(name = "user", animal = NULL) {

  msg <- paste0("hello ", name, " ...")

  if(is.null(animal)) {

    animal <- sample(names(cowsay::animals), 1)

  }

  cowsay::say(msg, by = animal)

}
```

DESCRIPTION

```
Package: scratchr
Type: Package
Title: miscellaneous functions to demo package development
Version: 0.1.0
Author: VP Nagraj
Maintainer: vpnagraj <vpnagraj@virginia.edu>
Description: This package contains several functions that were writte
License: GPL-3
Imports:
  cowsay
Encoding: UTF-8
LazyData: true
RoxygenNote: 6.0.1.9000
```

R/

```r
#' Birthday clock
#'
#' @param month English name for the month of your birth; see \code{
#' @param day integer day of the month in which you were born; shoul
#'
#' @export
#' @examples
#' bday_clock("January", 1)
bday_clock <- function(month, day) {

  bday_str <- paste(month,
                    day,
                    lubridate::year(Sys.Date()) + 1,
                    sep = "/")

  bday <- as.Date(bday_str, format = "%B/%d/%Y")
  d <- as.numeric(bday - Sys.Date(), units = "days")

  msg <- paste0(d, " days until your birthday there are ...")
  cowsay::say(msg, by = "yoda")

}
```

DESCRIPTION

```
Package: scratchr
Type: Package
Title: miscellaneous functions to demo package development
Version: 0.1.0
Author: VP Nagraj
Maintainer: vpnagraj <vpnagraj@virginia.edu>
Description: This package contains several functions that were writte
License: GPL-3
Imports:
  cowsay,
  lubridate
Encoding: UTF-8
LazyData: true
RoxygenNote: 6.0.1.9000
```

NAMESPACE

```
# Generated by roxygen2: do not edit by hand

export(bday_clock)
export(hello)
```

R/

```r
#' Next year helper function
#'
#' @return
#'
#'
next_year <- function() {

  lubridate::year(Sys.Date()) + 1

}
```

R/

```r
#' Birthday clock
#'
#' @param month English name for the month of your birth; see \code{
#' @param day integer day of the month in which you were born; shoul
#'
#' @export
#' @examples
#' bday_clock("January", 1)
bday_clock <- function(month, day) {

  bday_str <- paste(month,
                    day,
                    next_year(),
                    sep = "/")

  bday <- as.Date(bday_str, format = "%B/%d/%Y")
  d <- as.numeric(bday - Sys.Date(), units = "days")

  msg <- paste0(d, " days until your birthday there are ...")
  cowsay::say(msg, by = "yoda")

}
```

R/

```r
#' Say hello
#' @param name specify what you would like the program to call you; c
#' @param animal this argument allows you to choose an animal to acc
#' @export
#' @examples
#' hello("hal")
hello <- function(name = "user", animal = NULL) {

  msg <- paste0("hello ",
                name,
                " ...\n",
                "you better get busy\n",
                next_year(),
                " will be here soon :)")

  if(is.null(animal)) {
    animal <- sample(names(cowsay::animals), 1)
  }

  cowsay::say(msg, by = animal)

}
```

NAMESPACE

```
# Generated by roxygen2: do not edit by hand

export(bday_clock)
export(hello)
```

R/

```r
#' Birthday clock
#'
#' @param month English name for the month of your birth; see \code{
#' @param day integer day of the month in which you were born; shoul
#' @export
#' @examples
#' bday_clock("January", 1)
bday_clock <- function(month, day) {

  stopifnot(month %in% base::month.name)

  bday_str <- paste(month,
                    day,
                    next_year(),
                    sep = "/")

  bday <- as.Date(bday_str, format = "%B/%d/%Y")
  d <- as.numeric(bday - Sys.Date(), units = "days")

  msg <- paste0(d, " days until your birthday there are ...")
  cowsay::say(msg, by = "yoda")

}
```

testing

1. create testing template

```
devtools::use_testthat()
```

2. write tests

`{package}/tests/testthat/{test}.R`

3. run tests

```
devtools::test()
```

4. find gaps in test coverage

```
covr::package_coverage()
covr::shine(covr::package_coverage())
```

tests/testthat/

```
context("bday clock")

test_that("bday clock input is ok", {

  expect_error(bday_clock("foobrary", 1))

})
```

```
build %>% check
```

more checks

devtools

```
devtools::lint(".")
devtools::spell_check(".")
devtools::build_win(".")
```

goodpractice

```
source("https://install-github.me/MangoTheCat/goodpractice")
goodpractice::gp(".")
```

BiocCheck

```
source("https://bioconductor.org/biocLite.R")
biocLite("BiocCheck")
BiocCheck::BiocCheck(".")
```

including data

R data file with object(s) to be loaded = `/data`

instructions for creating / re-creating data = `/data-raw`

documentation for data = `R/data.R`

data-raw/

```r
# create bogus population projections
future <-
  ts(
    data = c(210,240,280,340,360,390),
    start = 1980,
    end = 2030,
    frequency = .1
  )

# combine future with built-in uspop dataset
futurepop <-
  ts(
    data  = c(datasets::uspop, future),
    start = 1790,
    end = 2030,
    frequency = .1
  )

save(futurepop, file = "data/futurepop.rda")
```

vignettes

```
devtools::use_vignette()

devtools::build_vignette()
```

continuous integration

```
devtools::use_travis()
```

release

```
devtools::release()
```

```
R CMD BUILD {PACKAGE}/ + upload
```