

Statistics 360: Advanced R for Data Science

Lecture 9

Brad McNeney

More on R packages

- ▶ Reference: R Packages online book
<https://r-pkgs.org/tests.html>
- ▶ vignettes (chapter 11)
- ▶ testing (chapter 12)
- ▶ data (chapter 14)

Vignettes

- ▶ A package vignette is long-form documentation that teaches a user how to use all of the features of your package
 - ▶ Some packages have multiple vignettes to illustrate different aspects of the package, but your mars package will have only one.
- ▶ These days, most vignettes are written in RMarkdown.
 - ▶ You can browse the RMarkdown source for my lecture notes on GitHub.

Getting started with vignettes

► Use `usethis::use_vignette("mars")`

```
> usethis::use_vignette("mars")
```

```
Setting active project to '/Users/mcneney/Teaching/Stat360_2022/SFUStat360/mar
```

```
Adding 'knitr' to Suggests field in DESCRIPTION
```

```
Setting VignetteBuilder field in DESCRIPTION to 'knitr'
```

```
Adding 'inst/doc' to '.gitignore'
```

```
Creating 'vignettes/'
```

```
Adding '*.html', '*.R' to 'vignettes/.gitignore'
```

```
Adding 'rmarkdown' to Suggests field in DESCRIPTION
```

```
Writing 'vignettes/mars.Rmd'
```

```
Modify 'vignettes/mars.Rmd'
```

Draft RMarkdown document

- ▶ As noted, `use_vignette()` generates a template RMarkdown document in the vignettes directory that you can edit.
- ▶ See the RStudio Help -> Markdown Quick Reference and <https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf> for help on RMarkdown.

Sections of your mars package vignette

- ▶ See the project rubric for details.
 - ▶ Introduction to the MARS algorithm
 - ▶ Calling `mars()`
 - ▶ Illustration of the methods

Testing

- ▶ Use `usethis::use_testthat()` to create a skeleton test suite for your package.
- ▶ `use_testthat()` prints messages to tell you what it's doing and then suggests calling `use_test()` to generate a test file.
 - ▶ We will not use `use_test()`. Instead, in the week 9 exercises you will manually create test files.

```
> usethis::use_testthat()  
Adding 'testthat' to Suggests field in DESCRIPTION  
Setting Config/testthat/edition field in DESCRIPTION to '3'  
Creating 'tests/testthat/'  
Writing 'tests/testthat.R'  
Call `use_test()` to initialize a basic test file and open it for editing.
```


Example test file

- ▶ Test files are saved in the `tests/testthat` folder and should:
 1. Load the package
 2. Load the `.RData` files that contain the inputs and expected output of the function(s) you are testing,
 3. Call `testthat::test_that()` to do the test.
 - ▶ `test_that()` takes a description of the test and code (usually a call to `expect_equal()`) to do the tests as arguments.
- ▶ `expect_equal()` takes two objects as input. The first (object) is the from running the function to be tested from the current version, and the second (expectation) is the “right answer” from a previous version of the function that was known to work.
 - ▶ In general you have to generate and save expectations to your `tests/testthat` folder, but in this class I have done that for you.

```
library(mars)
load("testsomething.RData") # contains an R object with the "right answer"
test_that("myfunc works properly",{
  # could have multiple lines/tests here in the expression between {}
  expect_equal(myfunc(myinput),testoutput)
})
```

Package datasets

- ▶ Datasets that you include with your package may be taken from external sources (spreadsheets, text files, websites) or may be simulated.
- ▶ The raw data source(s) and commands used to wrangle them into R objects should be stored in the `data-raw` directory of your package.
- ▶ Use `usethis::use_data_raw()` to get started and `usethis::use_data()` to save your dataset.
 - ▶ See the week 9 exercises for an example.

```
> usethis::use_data_raw("marstestdata")  
Creating 'data-raw/'  
Adding '^data-raw$' to '.Rbuildignore'  
Writing 'data-raw/marstestdata.R'  
Modify 'data-raw/marstestdata.R'  
Finish the data preparation script in 'data-raw/marstestdata.R'  
Use `usethis::use_data()` to add prepared data to package
```

Documenting datasets

- ▶ Datasets in `data/` are exported and so must be documented.
- ▶ Use Roxygen2 comments in a dummy `.R` script in your `R/` folder.
- ▶ See the `Exercises/ProjectTestfiles/data.R` script in the class repository for an example.
 - ▶ You will need to copy this example to your package's `R` folder.