R packages structure student research projects and enhance research reproducibility

Frederick J. Boehm[1,*] Bret M. Hanlon[2]

12 August, 2021

Abstract

We present a case for using R packages as the core structure for research projects. We review the ways in which R packages may be used for teaching research skills in data science and statistics. Building on existing software tools, we present reproducible R code that enables users to construct their own packages for use in data analysis and research. The R package structure has much to offer teachers and students of data science and statistics. Below, we review the standard structure of R packages. We then elaborate this structure to include a subdirectory for additional data analysis resources. Our audience is instructors of courses for upper level undergraduate students. We assume that these undergraduate students have modest familiarity with R computing, presumably developed in data science and statistics courses and independent research projects, so we don't discuss approaches to teaching basic R skills.

Contents

[1] University of Michigan
[2] University of Wisconsin-Madison

[*] Correspondence: Frederick J. Boehm <frederick.boehm@gmail.com>

Keywords: keyword 1; keyword 2; keyword 3

Highlights: These are the highlights.

1 Introduction

Many undergraduate programs in statistics and data science feature a capstone course with student projects. Such projects serve a variety of purposes; for example, they may aid students in preparing for transitions to the workplace or serve as a bridge to study for advanced degrees.

Given the ongoing crisis in research reproducibility, much recent discussion has sought to identify practices that promote research integrity and reproducibility. Throughout this article, we say that an article is

"reproducible" if, given the raw data and analysis instructions, a reasonably experienced user can generate exactly the reported results. What constitutes analysis instructions clearly plays a crucial role.

One practice that we've found useful, both in our own research and in our teaching of research skills, is the use of R packages as the skeleton structure of a research project in data science and statistics. Recent advances in developer-friendly R tools, such as the functions in the packages `devtools`, `usethis`, `testthat`, and `rrtools`, have lowered barriers to creating, testing, and maintaining R packages for data analysis projects.

Throughout this article, we share reproducible R code that can be used to create an R package for packages.

1.1 R package structure

The structure of an R package consists of a collection of specifically named directories and files. Every R package must have:

TABLE 1: List of required files and directories for an R package. (Not sure that I can name these off the top of my head, but it's easy to look them up). Files: DESCRIPTION, NAMESPACE, etc Directories: R, man, ...

- Need to explain contents and structure of each item in TABLE 1.

1.2 How to construct an R package

- usethis R package; devtools R package; rrtools R package on github
  - which functions in `usethis` to get started with a new package??
- Provide R exact code here
- Rstudio IDE shortcuts
- Github actions (added to package via calls to usethis functions)

We demonstrate one way to construct an R package from scratch. We do this from within an R session. We also discuss how users of Rstudio IDE software can quickly and efficiently start a new R package.

What subdirectories to add to a "mypackage/analysis" directory: Rscript: .R files that Rmd: Rmarkdown files and their outputs data: inputs of any format results: tables, figures, intermediate rds files?

1.3 Teaching others how to construct an R package

- Existing resources:
  - Check Greg Wilson's book Teaching Technology Together
  - Data Carpentry resources?? I'm unaware of R package materials from The Carpentries
  - Rstudio's online refs
- Karl Broman's tools4rr page: https://kbroman.org/Tools4RR/
-

1.4 Assessment of R packages for data analysis projects

- Include a detailed rubric in our paper
- Scaffolding a long-term project with intermediate deadlines

2 Discussion

3 Conclusion

4 Acknowledgements

5 References

5.0.1 Colophon

This report was generated on 2021-08-12 22:21:32 using the following computational environment and dependencies:

```
#> - Session info ---------------------------------------------------------------
#>  setting  value
#>  version  R version 4.0.4 (2021-02-15)
#>  os       Ubuntu 21.04
#>  system   x86_64, linux-gnu
#>  ui       X11
#>  language (EN)
#>  collate  en_US.UTF-8
#>  ctype    en_US.UTF-8
#>  tz       America/New_York
#>  date     2021-08-12
#>
#> - Packages -------------------------------------------------------------------
#>  package     * version     date       lib source
#>  bookdown      0.22        2021-04-22 [1] CRAN (R 4.0.4)
#>  cachem        1.0.5       2021-05-15 [1] CRAN (R 4.0.4)
#>  callr         3.7.0       2021-04-20 [1] CRAN (R 4.0.4)
#>  cli           2.5.0       2021-04-26 [1] CRAN (R 4.0.4)
#>  crayon        1.4.1       2021-02-08 [1] CRAN (R 4.0.3)
#>  desc          1.3.0       2021-03-05 [1] CRAN (R 4.0.4)
#>  devtools      2.4.1       2021-05-05 [1] CRAN (R 4.0.4)
#>  digest        0.6.27      2020-10-24 [1] CRAN (R 4.0.3)
#>  ellipsis      0.3.2       2021-04-29 [1] CRAN (R 4.0.4)
#>  evaluate      0.14        2019-05-28 [1] CRAN (R 4.0.1)
#>  fastmap       1.1.0       2021-01-25 [1] CRAN (R 4.0.3)
#>  fs            1.5.0       2020-07-31 [1] CRAN (R 4.0.3)
#>  glue          1.4.2       2020-08-27 [1] CRAN (R 4.0.3)
#>  htmltools     0.5.1.1     2021-01-22 [1] CRAN (R 4.0.3)
#>  knitr         1.33        2021-04-24 [1] CRAN (R 4.0.4)
#>  lifecycle     1.0.0       2021-02-15 [1] CRAN (R 4.0.3)
#>  magrittr      2.0.1       2020-11-17 [1] CRAN (R 4.0.3)
#>  memoise       2.0.0       2021-01-26 [1] CRAN (R 4.0.3)
#>  pkgbuild      1.2.0       2020-12-15 [1] CRAN (R 4.0.3)
#>  pkgload       1.2.1       2021-04-06 [1] CRAN (R 4.0.4)
#>  prettyunits   1.1.1       2020-01-24 [1] CRAN (R 4.0.1)
#>  processx      3.5.2       2021-04-30 [1] CRAN (R 4.0.4)
#>  ps            1.6.0       2021-02-28 [1] CRAN (R 4.0.4)
#>  purrr         0.3.4       2020-04-17 [1] CRAN (R 4.0.1)
#>  R6            2.5.0       2020-10-28 [1] CRAN (R 4.0.3)
#>  remotes       2.3.0       2021-04-01 [1] CRAN (R 4.0.4)
#>  rlang         0.4.11.9000 2021-05-11 [1] Github (r-lib/rlang@7cd1f5c)
#>  rmarkdown     2.9         2021-06-15 [1] CRAN (R 4.0.4)
#>  rprojroot     2.0.2       2020-11-15 [1] CRAN (R 4.0.3)
#>  sessioninfo   1.1.1       2018-11-05 [1] CRAN (R 4.0.1)
#>  stringi       1.6.2       2021-05-17 [1] CRAN (R 4.0.4)
#>  stringr       1.4.0       2019-02-10 [1] CRAN (R 4.0.1)
#>  testthat      3.0.2       2021-02-14 [1] CRAN (R 4.0.3)
#>  usethis       2.0.1.9000  2021-02-15 [1] Github (r-lib/usethis@aaf79d8)
#>  withr         2.4.2       2021-04-18 [1] CRAN (R 4.0.4)
#>  xfun          0.23        2021-05-15 [1] CRAN (R 4.0.4)
```

```
#>  yaml          2.2.1       2020-02-01 [1] CRAN (R 4.0.1)
#>
#> [1] /home/fred/R/x86_64-pc-linux-gnu-library/4.0
#> [2] /usr/local/lib/R/site-library
#> [3] /usr/lib/R/site-library
#> [4] /usr/lib/R/library
```

The current Git commit details are:

```
#> Local:    master /home/fred/work/research/reproducible
#> Remote:   master @ origin (https://github.com/fboehm/reproducible.git)
#> Head:     [3a39bc4] 2021-07-25: added preliminary abstract
```