

R Packages

Turn your R code into packages that others can easily download and use. This practical book shows you how to bundle reusable R functions, sample data, and documentation together by applying author Hadley Wickham's package development philosophy. In the process, you'll work with devtools, roxygen, and testthat, a set of R packages that automates common development tasks. Devtools encapsulates best practices that Hadley has learned from years of working with this programming language.

Ideal for developers, data scientists, and programmers with various backgrounds, this book starts with the basics and shows you how to improve your package writing over time. You'll learn to focus on what you want your package to do, rather than think about package structure.

- Learn about the most useful components of an R package, including vignettes and unit tests
- Take advantage of devtools to automate anything you can
- Get tips on good style, such as organizing functions into files
- Streamline your development process with devtools
- Discover the best way to submit your package to the Comprehensive R Archive Network (CRAN)
- Learn from a well-respected member of the R community who created 30 R packages, including ggplot2, dplyr, and tidyr

Hadley Wickham is Chief Scientist at RStudio. He's a well-respected member of the R community who has written and contributed to over 30 R packages. Hadley won the John Chambers Award for Statistical Computing for his work developing tools for data reshaping and visualization.

"This book is a practical, hands-on guide for building high-quality software in R. Any R programmer looking to 'reach the next level' would do well to give this a read."

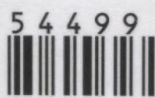
—Wes McKinney
creator of pandas

DATA/DATA SCIENCE

US \$44.99

CAN \$59.99

ISBN: 978-1-491-91059-7



9



Twitter: @oreillymedia
facebook.com/oreilly

Preface.....	ix
--------------	----

Part I. Getting Started

1. Introduction.....	1
Philosophy.....	2
Getting Started.....	3
Conventions.....	4
Colophon.....	4
2. Package Structure.....	5
Naming Your Package.....	5
Requirements for a Name.....	5
Strategies for Creating a Name.....	5
Creating a Package.....	6
RStudio Projects.....	8
What Is an RStudio Project File?.....	9
What Is a Package?.....	11
Source Packages.....	11
Bundled Packages.....	12
Binary Packages.....	13
Installed Packages.....	14
In-Memory Packages.....	15
What Is a Library?.....	16

Part II. Package Components

3. R Code..... 21

 R Code Workflow 21

 Organizing Your Functions 21

 Code Style 22

 Object Names 23

 Spacing 24

 Curly Braces 25

 Line Length 25

 Indentation 25

 Assignment 26

 Commenting Guidelines 26

 Top-Level Code 27

 Loading Code 27

 The R Landscape 28

 When You Do Need Side Effects 29

 S4 Classes, Generics, and Methods 31

 CRAN Notes 31

4. Package Metadata..... 33

 Dependencies: What Does Your Package Need? 34

 Versioning 36

 Other Dependencies 36

 Title and Description: What Does Your Package Do? 37

 Author: Who Are You? 38

 On CRAN 40

 License: Who Can Use Your Package? 40

 On CRAN 41

 Version 41

 Other Components 42

5. Object Documentation..... 43

 The Documentation Workflow 44

 Alternative Documentation Workflow 46

 Roxygen Comments 47

 Documenting Functions 49

 Documenting Datasets 51

 Documenting Packages 51

 Documenting Classes, Generics, and Methods 51

 S3 51

S4	52
RC	53
Special Characters	54
Do Repeat Yourself	54
Inheriting Parameters from Other Functions	55
Documenting Multiple Functions in the Same File	55
Text Formatting Reference Sheet	56
Character Formatting	57
Links	57
Lists	57
Mathematics	58
Tables	58
6. Vignettes: Long-Form Documentation.....	59
Vignette Workflow	60
Metadata	61
Markdown	62
Sections	63
Lists	63
Inline Formatting	64
Tables	64
Code	64
Knitr	65
Options	66
Development Cycle	67
Advice for Writing Vignettes	68
Organization	68
CRAN Notes	69
Where to Go Next	69
7. Testing.....	71
Test Workflow	72
Test Structure	73
Expectations	74
Writing Tests	76
What to Test	77
Skipping a Test	77
Building Your Own Testing Tools	78
Test Files	80
CRAN Notes	80

8. Namespace.....	81
Motivation	81
Search Path	82
The NAMESPACE	84
Workflow	86
Exports	86
S3	87
S4	88
RC	88
Data	88
Imports	88
R Functions	89
S3	89
S4	90
Compiled Functions	90
9. External Data.....	91
Exported Data	91
Documenting Datasets	93
Internal Data	93
Raw Data	94
Other Data	94
CRAN Notes	94
10. Compiled Code.....	97
C++	97
Workflow	98
Documentation	99
Exporting C++ Code	100
Importing C++ Code	100
Best Practices	100
C	101
Getting Started with .Call()	102
Getting Started with .C()	103
Workflow	104
Exporting C Code	104
Importing C Code	106
Best Practices	106
Debugging Compiled Code	107
Makefiles	109
Other Languages	110
Licensing	110

Development Workflow	110
CRAN Issues	111
11. Installed Files.....	113
Package Citation	114
Other Languages	115
12. Other Components.....	117
Demos	117
<hr/>	
Part III. Best Practices	
13. Git and GitHub.....	121
RStudio, Git, and GitHub	122
Initial Setup	123
Creating a Local Git Repository	124
Seeing What's Changed	126
Recording Changes	128
Best Practices for Commits	130
Ignoring Files	132
Undoing Mistakes	132
Synchronizing with GitHub	134
Benefits of Using GitHub	135
Working with Others	137
Issues	138
Branches	139
Making a Pull Request	140
Submitting a Pull Request to Another Repo	142
Reviewing and Accepting Pull Requests	144
Learning More	145
14. Automated Checking.....	147
Workflow	147
Checks	148
Check Metadata	148
Package Structure	149
Description	151
Namespace	152
R Code	153
Data	155
Documentation	156

Demos	158
Compiled Code	158
Tests	158
Vignettes	159
Checking After Every Commit with Travis	160
Basic Config	161
Other Uses	161
15. Releasing a Package.....	163
Version Number	163
Backward Compatibility	164
The Submission Process	166
Test Environments	168
Check Results	169
Reverse Dependencies	169
CRAN Policies	170
Important Files	171
README.md	171
README.Rmd	172
NEWS.md	173
Release	173
On Failure	174
Binary Builds	175
Prepare for Next Version	176
Publicizing Your Package	176
Congratulations!	176
Index.....	177