Contents

Preface xiii

1 Introduction 1
- 1.1 Why R? 1
- 1.2 Who should read this book 3
- 1.3 What you will get out of this book 4
- 1.4 What you will not learn 4
- 1.5 Meta-techniques 5
- 1.6 Recommended reading 5
- 1.7 Getting help 6
- 1.8 Acknowledgments 7
- 1.9 Conventions 10
- 1.10 Colophon 12

I Foundations 13

Introduction 15

2 Names and values 17
- 2.1 Introduction 17
- 2.2 Binding basics 19
- 2.3 Copy-on-modify 22
- 2.4 Object size 29
- 2.5 Modify-in-place 31
- 2.6 Unbinding and the garbage collector 36
- 2.7 Quiz answers 38

3 Vectors 39
- 3.1 Introduction 39
- 3.2 Atomic vectors 41
- 3.3 Attributes 45
- 3.4 S3 atomic vectors 50
- 3.5 Lists 55
- 3.6 Data frames and tibbles 58

3.7 NULL 69
3.8 Quiz answers 70

4 Subsetting **73**
4.1 Introduction 73
4.2 Selecting multiple elements 74
4.3 Selecting a single element 81
4.4 Subsetting and assignment 85
4.5 Applications 87
4.6 Quiz answers 94

5 Control flow **97**
5.1 Introduction 97
5.2 Choices 98
5.3 Loops 102
5.4 Quiz answers 106

6 Functions **107**
6.1 Introduction 107
6.2 Function fundamentals 109
6.3 Function composition 113
6.4 Lexical scoping 115
6.5 Lazy evaluation 120
6.6 ... (dot-dot-dot) 125
6.7 Exiting a function 128
6.8 Function forms 134
6.9 Quiz answers 141

7 Environments **143**
7.1 Introduction 143
7.2 Environment basics 144
7.3 Recursing over environments 153
7.4 Special environments 156
7.5 Call stacks 165
7.6 As data structures 169
7.7 Quiz answers 170

8 Conditions **171**
8.1 Introduction 171
8.2 Signalling conditions 173
8.3 Ignoring conditions 178
8.4 Handling conditions 180
8.5 Custom conditions 188
8.6 Applications 193
8.7 Quiz answers 201

II Functional programming **203**

Introduction **205**

9 Functionals **209**
9.1 Introduction 209
9.2 My first functional: `map()` 211
9.3 Purrr style 221
9.4 Map variants 223
9.5 Reduce family 233
9.6 Predicate functionals 239
9.7 Base functionals 242

10 Function factories **247**
10.1 Introduction 247
10.2 Factory fundamentals 249
10.3 Graphical factories 256
10.4 Statistical factories 261
10.5 Function factories + functionals 268

11 Function operators **271**
11.1 Introduction 271
11.2 Existing function operators 272
11.3 Case study: Creating your own function operators 278

III Object-oriented programming **283**

Introduction **285**

12 Base types **291**
12.1 Introduction 291
12.2 Base versus OO objects 292
12.3 Base types 293

13 S3 **297**
13.1 Introduction 297
13.2 Basics 298
13.3 Classes 302
13.4 Generics and methods 309
13.5 Object styles 312
13.6 Inheritance 314
13.7 Dispatch details 320

14 R6 **325**
14.1 Introduction 325
14.2 Classes and methods 326

14.3 Controlling access 332
14.4 Reference semantics 336
14.5 Why R6? 340

15 S4 341
15.1 Introduction 341
15.2 Basics 343
15.3 Classes 344
15.4 Generics and methods 349
15.5 Method dispatch 352
15.6 S4 and S3 359

16 Trade-offs 363
16.1 Introduction 363
16.2 S4 versus S3 364
16.3 R6 versus S3 365

IV Metaprogramming 371

Introduction 373

17 Big picture 375
17.1 Introduction 375
17.2 Code is data 376
17.3 Code is a tree 377
17.4 Code can generate code 378
17.5 Evaluation runs code 380
17.6 Customising evaluation with functions 381
17.7 Customising evaluation with data 382
17.8 Quosures 383

18 Expressions 385
18.1 Introduction 385
18.2 Abstract syntax trees 386
18.3 Expressions 391
18.4 Parsing and grammar 399
18.5 Walking AST with recursive functions 404
18.6 Specialised data structures 411

19 Quasiquotation 415
19.1 Introduction 415
19.2 Motivation 416
19.3 Quoting 419
19.4 Unquoting 424
19.5 Non-quoting 433
19.6 ... (dot-dot-dot) 436

19.7 Case studies 442
19.8 History 448

20 Evaluation 451
20.1 Introduction 451
20.2 Evaluation basics 452
20.3 Quosures 457
20.4 Data masks 462
20.5 Using tidy evaluation 468
20.6 Base evaluation 472

21 Translating R code 481
21.1 Introduction 481
21.2 HTML 482
21.3 LaTeX 492

V Techniques 501

Introduction 503

22 Debugging 505
22.1 Introduction 505
22.2 Overall approach 506
22.3 Locating errors 507
22.4 Interactive debugger 509
22.5 Non-interactive debugging 513
22.6 Non-error failures 516

23 Measuring performance 519
23.1 Introduction 519
23.2 Profiling 520
23.3 Microbenchmarking 526

24 Improving performance 531
24.1 Introduction 531
24.2 Code organisation 532
24.3 Checking for existing solutions 534
24.4 Doing as little as possible 535
24.5 Vectorise 539
24.6 Avoiding copies 541
24.7 Case study: t-test 542
24.8 Other techniques 544

25 Rewriting R code in C++ 547
25.1 Introduction 547
25.2 Getting started with C++ 548

25.3 Other classes 557
25.4 Missing values 559
25.5 Standard Template Library 563
25.6 Case studies 570
25.7 Using Rcpp in a package 574
25.8 Learning more 575
25.9 Acknowledgments 576

Bibliography **577**

Index **581**