

SECOND EDITION

Practical Data Science with

R



Nina Zumel
John Mount

FOREWORD

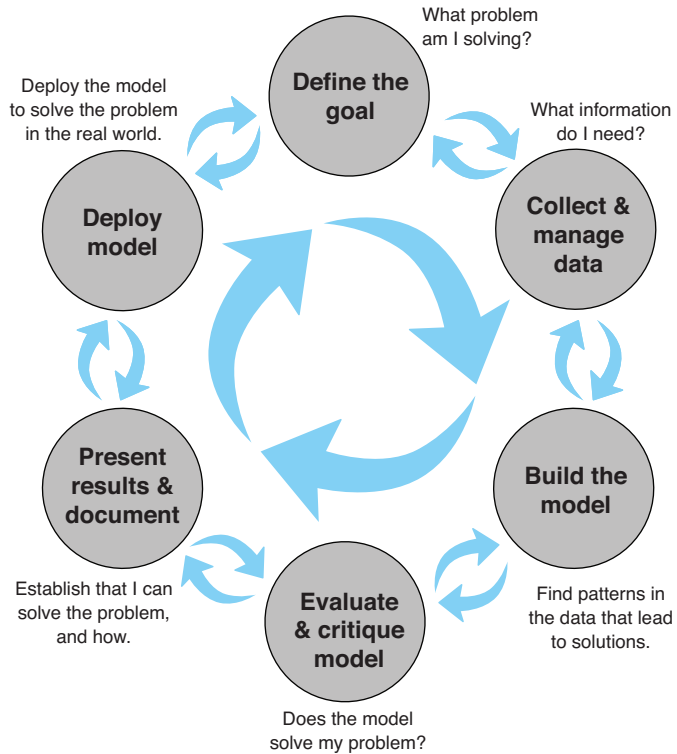
Jeremy Howard
Rachel Thomas



MANNING

Practical Data Science with R

The lifecycle of a data science project: loops within loops



Praise for the First Edition

Clear and succinct, this book provides the first hands-on map of the fertile ground between business acumen, statistics, and machine learning.

—Dwight Barry,
Group Health Cooperative

This is the book that I wish was available when I was first learning Data Science. The author presents a thorough and well-organized approach to the mechanics and mastery of Data Science, which is a conglomeration of statistics, data analysis, and computer science.

—Justin Fister, AI researcher,
PaperRater.com

The most comprehensive content I have seen on Data Science with R.

—Romit Singhai, SGI

Covers the process end to end, from data exploration to modeling to delivering the results.

—Nezih Yigitbasi,
Intel

Full of useful gems for both aspiring and experienced data scientists.

—Fred Rahmanian,
Siemens Healthcare

Hands-on data analysis with real-world examples. Highly recommended.

—Dr. Kostas Passadis,
IPTO

In working through the book, one gets the impression of being guided by knowledgeable and experienced professionals who are holding nothing back.

—Amazon reader

*Practical Data Science
with R*

SECOND EDITION

NINA ZUMEL
AND JOHN MOUNT

FOREWORD BY JEREMY HOWARD
AND RACHEL THOMAS



MANNING
SHELTER ISLAND

For online information and ordering of this and other Manning books, please visit www.manning.com. The publisher offers discounts on this book when ordered in quantity. For more information, please contact


Special Sales Department
Manning Publications Co.
20 Baldwin Road
PO Box 761
Shelter Island, NY 11964
Email: orders@manning.com

© 2020 by Manning Publications Co. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by means electronic, mechanical, photocopying, or otherwise, without prior written permission of the publisher.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in the book, and Manning Publications was aware of a trademark claim, the designations have been printed in initial caps or all caps.

© Recognizing the importance of preserving what has been written, it is Manning's policy to have the books we publish printed on acid-free paper, and we exert our best efforts to that end. Recognizing also our responsibility to conserve the resources of our planet, Manning books are printed on paper that is at least 15 percent recycled and processed without the use of elemental chlorine.

 Manning Publications Co.
20 Baldwin Road
PO Box 761
Shelter Island, NY 11964

Development editor: Dustin Archibald
Technical development editor: Doug Warren
Review editor: Aleksandar Dragosavljević
Project manager: Lori Weidert
Copy editor: Ben Berg
Proofreader: Katie Tennant
Technical proofreader: Taylor Dolezal
Typesetter: Dottie Marsico
Cover designer: Marija Tudor

ISBN 9781617295874
Printed in the United States of America

To our parents

*Olive and Paul Zumel
Peggy and David Mount*

brief contents

PART 1	INTRODUCTION TO DATA SCIENCE	1
1	■ The data science process	3
2	■ Starting with R and data	18
3	■ Exploring data	51
4	■ Managing data	88
5	■ Data engineering and data shaping	113
PART 2	MODELING METHODS	161
6	■ Choosing and evaluating models	163
7	■ Linear and logistic regression	215
8	■ Advanced data preparation	274
9	■ Unsupervised methods	311
10	■ Exploring advanced methods	353
PART 3	WORKING IN THE REAL WORLD	401
11	■ Documentation and deployment	403
12	■ Producing effective presentations	437

contents

foreword xv
preface xvi
acknowledgments xvii
about this book xviii
about the authors xxv
about the foreword authors xxvi
about the cover illustration xxvii

PART 1 INTRODUCTION TO DATA SCIENCE1

1 *The data science process* 3

1.1 The roles in a data science project 4

Project roles 4

1.2 Stages of a data science project 6

Defining the goal 7 ▪ *Data collection and management* 8

Modeling 10 ▪ *Model evaluation and critique* 12

Presentation and documentation 14 ▪ *Model deployment
and maintenance* 15

1.3 Setting expectations 16

Determining lower bounds on model performance 16

2 *Starting with R and data* 18

2.1 Starting with R 19

Installing R, tools, and examples 20 ▪ *R programming* 20

- 2.2 Working with data from files 29
 - Working with well-structured data from files or URLs* 29
 - Using R with less-structured data* 34
- 2.3 Working with relational databases 37
 - A production-size example* 38

3 **Exploring data** 51

- 3.1 Using summary statistics to spot problems 53
 - Typical problems revealed by data summaries* 54
- 3.2 Spotting problems using graphics and visualization 58
 - Visually checking distributions for a single variable* 60
 - Visually checking relationships between two variables* 70

4 **Managing data** 88

- 4.1 Cleaning data 88
 - Domain-specific data cleaning* 89 ▪ *Treating missing values* 91 ▪ *The vtreat package for automatically treating missing variables* 95
- 4.2 Data transformations 98
 - Normalization* 99 ▪ *Centering and scaling* 101
 - Log transformations for skewed and wide distributions* 104
- 4.3 Sampling for modeling and validation 107
 - Test and training splits* 108 ▪ *Creating a sample group column* 109 ▪ *Record grouping* 110 ▪ *Data provenance* 111

5 **Data engineering and data shaping** 113

- 5.1 Data selection 116
 - Subsetting rows and columns* 116 ▪ *Removing records with incomplete data* 121 ▪ *Ordering rows* 124
- 5.2 Basic data transforms 128
 - Adding new columns* 128 ▪ *Other simple operations* 133
- 5.3 Aggregating transforms 134
 - Combining many rows into summary rows* 134
- 5.4 Multitable data transforms 137
 - Combining two or more ordered data frames quickly* 137
 - Principal methods to combine data from multiple tables* 143
- 5.5 Reshaping transforms 149
 - Moving data from wide to tall form* 149 ▪ *Moving data from tall to wide form* 153 ▪ *Data coordinates* 158

PART 2 MODELING METHODS161

6 *Choosing and evaluating models* 163

- 6.1 Mapping problems to machine learning tasks 164
 - Classification problems* 165 ▪ *Scoring problems* 166
 - Grouping: working without known targets* 167
 - Problem-to-method mapping* 169
- 6.2 Evaluating models 170
 - Overfitting* 170 ▪ *Measures of model performance* 174
 - Evaluating classification models* 175 ▪ *Evaluating scoring models* 185 ▪ *Evaluating probability models* 187
- 6.3 Local interpretable model-agnostic explanations (LIME) for explaining model predictions 195
 - LIME: Automated sanity checking* 197 ▪ *Walking through LIME: A small example* 197 ▪ *LIME for text classification* 204
 - Training the text classifier* 208 ▪ *Explaining the classifier's predictions* 209

7 *Linear and logistic regression* 215

- 7.1 Using linear regression 216
 - Understanding linear regression* 217 ▪ *Building a linear regression model* 221 ▪ *Making predictions* 222
 - Finding relations and extracting advice* 228 ▪ *Reading the model summary and characterizing coefficient quality* 230
 - Linear regression takeaways* 237
- 7.2 Using logistic regression 237
 - Understanding logistic regression* 237 ▪ *Building a logistic regression model* 242 ▪ *Making predictions* 243
 - Finding relations and extracting advice from logistic models* 248 ▪ *Reading the model summary and characterizing coefficients* 249 ▪ *Logistic regression takeaways* 256
- 7.3 Regularization 257
 - An example of quasi-separation* 257 ▪ *The types of regularized regression* 262 ▪ *Regularized regression with glmnet* 263

8 *Advanced data preparation* 274

- 8.1 The purpose of the vtreat package 275
- 8.2 KDD and KDD Cup 2009 277
 - Getting started with KDD Cup 2009 data* 278 ▪ *The bull-in-the-china-shop approach* 280

- 8.3 Basic data preparation for classification 282
 - The variable score frame* 284 ▪ *Properly using the treatment plan* 288
- 8.4 Advanced data preparation for classification 290
 - Using mkCrossFrameCExperiment()* 290 ▪ *Building a model* 292
- 8.5 Preparing data for regression modeling 297
- 8.6 Mastering the vtreat package 299
 - The vtreat phases* 299 ▪ *Missing values* 301
 - Indicator variables* 303 ▪ *Impact coding* 304
 - The treatment plan* 305 ▪ *The cross-frame* 306

9 Unsupervised methods 311

- 9.1 Cluster analysis 312
 - Distances* 313 ▪ *Preparing the data* 316 ▪ *Hierarchical clustering with hclust* 319 ▪ *The k-means algorithm* 332
 - Assigning new points to clusters* 338 ▪ *Clustering takeaways* 340
- 9.2 Association rules 340
 - Overview of association rules* 340 ▪ *The example problem* 342
 - Mining association rules with the arules package* 343
 - Association rule takeaways* 351

10 Exploring advanced methods 353

- 10.1 Tree-based methods 355
 - A basic decision tree* 356 ▪ *Using bagging to improve prediction* 359 ▪ *Using random forests to further improve prediction* 361 ▪ *Gradient-boosted trees* 368 ▪ *Tree-based model takeaways* 376
- 10.2 Using generalized additive models (GAMs) to learn non-monotone relationships 376
 - Understanding GAMs* 376 ▪ *A one-dimensional regression example* 378 ▪ *Extracting the non-linear relationships* 382
 - Using GAM on actual data* 384 ▪ *Using GAM for logistic regression* 387 ▪ *GAM takeaways* 388
- 10.3 Solving “inseparable” problems using support vector machines 389
 - Using an SVM to solve a problem* 390 ▪ *Understanding support vector machines* 395 ▪ *Understanding kernel functions* 397
 - Support vector machine and kernel methods takeaways* 399

PART 3 WORKING IN THE REAL WORLD401

11 *Documentation and deployment* 403

- 11.1 Predicting buzz 405
- 11.2 Using R markdown to produce milestone documentation 406
 - What is R markdown?* 407 ▪ *knitr technical details* 409
 - Using knitr to document the Buzz data and produce the model* 411
- 11.3 Using comments and version control for running documentation 414
 - Writing effective comments* 414 ▪ *Using version control to record history* 416 ▪ *Using version control to explore your project* 422 ▪ *Using version control to share work* 424
- 11.4 Deploying models 428
 - Deploying demonstrations using Shiny* 430 ▪ *Deploying models as HTTP services* 431 ▪ *Deploying models by export* 433 ▪ *What to take away* 435

12 *Producing effective presentations* 437

- 12.1 Presenting your results to the project sponsor 439
 - Summarizing the project's goals* 440 ▪ *Stating the project's results* 442 ▪ *Filling in the details* 444 ▪ *Making recommendations and discussing future work* 446
 - Project sponsor presentation takeaways* 446
- 12.2 Presenting your model to end users 447
 - Summarizing the project goals* 447 ▪ *Showing how the model fits user workflow* 448 ▪ *Showing how to use the model* 450 ▪ *End user presentation takeaways* 452
- 12.3 Presenting your work to other data scientists 452
 - Introducing the problem* 452 ▪ *Discussing related work* 453
 - Discussing your approach* 454 ▪ *Discussing results and future work* 455 ▪ *Peer presentation takeaways* 457

appendix A Starting with R and other tools 459

appendix B Important statistical concepts 484

appendix C Bibliography 519

index 523

foreword

Practical Data Science with R, Second Edition, is a hands-on guide to data science, with a focus on techniques for working with structured or tabular data, using the R language and statistical packages. The book emphasizes machine learning, but is unique in the number of chapters it devotes to topics such as the role of the data scientist in projects, managing results, and even designing presentations. In addition to working out how to code up models, the book shares how to collaborate with diverse teams, how to translate business goals into metrics, and how to organize work and reports. If you want to learn how to use R to work as a data scientist, get this book.

We have known Nina Zumel and John Mount for a number of years. We have invited them to teach with us at Singularity University. They are two of the best data scientists we know. We regularly recommend their original research on cross-validation and impact coding (also called target encoding). In fact, chapter 8 of *Practical Data Science with R* teaches the theory of impact coding and uses it through the author's own R package: `vtreat`.

Practical Data Science with R takes the time to describe what data science is, and how a data scientist solves problems and explains their work. It includes careful descriptions of classic supervised learning methods, such as linear and logistic regression. We liked the survey style of the book and extensively worked examples using contest-winning methodologies and packages such as random forests and `xgboost`. The book is full of useful, shared experience and practical advice. We notice they even include our own trick of using random forest variable importance for initial variable screening.

Overall, this is a great book, and we highly recommend it.

—JEREMY HOWARD
AND RACHEL THOMAS

preface

This is the book we wish we'd had as we were teaching ourselves that collection of subjects and skills that has come to be referred to as *data science*. It's the book that we'd like to hand out to our clients and peers. Its purpose is to explain the relevant parts of statistics, computer science, and machine learning that are crucial to data science.

Data science draws on tools from the empirical sciences, statistics, reporting, analytics, visualization, business intelligence, expert systems, machine learning, databases, data warehousing, data mining, and big data. It's because we have so many tools that we need a discipline that covers them all. What distinguishes data science itself from the tools and techniques is the central goal of deploying effective decision-making models to a production environment.

Our goal is to present data science from a pragmatic, practice-oriented viewpoint. We work toward this end by concentrating on fully worked exercises on real data—altogether, this book works through over 10 significant datasets. We feel that this approach allows us to illustrate what we really want to teach and to demonstrate all the preparatory steps necessary in any real-world project.

Throughout our text, we discuss useful statistical and machine learning concepts, include concrete code examples, and explore partnering with and presenting to non-specialists. If perhaps you don't find one of these topics novel, we hope to shine a light on one or two other topics that you may not have thought about recently.

acknowledgments

We wish to thank our colleagues and others who read and commented on our early chapter drafts. Special appreciation goes to our reviewers: Charles C. Earl, Christopher Kardell, David Meza, Domingo Salazar, Doug Sparling, James Black, John MacKintosh, Owen Morris, Pascal Barbedo, Robert Samohyl, and Taylor Dolezal. Their comments, questions, and corrections have greatly improved this book. We especially would like to thank our development editor, Dustin Archibald, and Cynthia Kane, who worked on the first edition, for their ideas and support. The same thanks go to Nichole Beard, Benjamin Berg, Rachael Herbert, Katie Tennant, Lori Weidert, Cheryl Weisman, and all the other editors who worked hard to make this a great book.

In addition, we thank our colleague David Steier, Professor Doug Tygar from UC Berkeley's School of Information Science, Professor Robert K. Kuzoff from the Departments of Biological Sciences and Computer Science at the University of Wisconsin-Whitewater, as well as all the other faculty and instructors who have used this book as a teaching text. We thank Jim Porzak, Joseph Rickert, and Alan Miller for inviting us to speak at the R users groups, often on topics that we cover in this book. We especially thank Jim Porzak for having written the foreword to the first edition, and for being an enthusiastic advocate of our book. On days when we were tired and discouraged and wondered why we had set ourselves to this task, his interest helped remind us that there's a need for what we're offering and the way we're offering it. Without this encouragement, completing this book would have been much harder. Also, we'd like to thank Jeremy Howard and Rachel Thomas for writing the new foreword, inviting us to speak, and providing their strong support.

about this book

This book is about data science: a field that uses results from statistics, machine learning, and computer science to create predictive models. Because of the broad nature of data science, it's important to discuss it a bit and to outline the approach we take in this book.

What is data science?

The statistician William S. Cleveland defined data science as an interdisciplinary field larger than statistics itself. We define data science as managing the process that can transform hypotheses and data into actionable predictions. Typical predictive analytic goals include predicting who will win an election, what products will sell well together, which loans will default, and which advertisements will be clicked on. The data scientist is responsible for acquiring and managing the data, choosing the modeling technique, writing the code, and verifying the results.

Because data science draws on so many disciplines, it's often a "second calling." Many of the best data scientists we meet started as programmers, statisticians, business intelligence analysts, or scientists. By adding a few more techniques to their repertoire, they became excellent data scientists. That observation drives this book: we introduce the practical skills needed by the data scientist by concretely working through all of the common project steps on real data. Some steps you'll know better than we do, some you'll pick up quickly, and some you may need to research further.

Much of the theoretical basis of data science comes from statistics. But data science as we know it is strongly influenced by technology and software engineering methodologies, and has largely evolved in heavily computer science- and information technology-driven groups. We can call out some of the engineering flavor of data science by listing some famous examples:

- Amazon's product recommendation systems
- Google's advertisement valuation systems
- LinkedIn's contact recommendation system
- Twitter's trending topics
- Walmart's consumer demand projection systems

These systems share a lot of features:

- All of these systems are *built off large datasets*. That's not to say they're all in the realm of big data. But none of them could've been successful if they'd only used small datasets. To manage the data, these systems require concepts from computer science: database theory, parallel programming theory, streaming data techniques, and data warehousing.
- Most of these systems are *online or live*. Rather than producing a single report or analysis, the data science team deploys a decision procedure or scoring procedure to either directly make decisions or directly show results to a large number of end users. The production deployment is the last chance to get things right, as the data scientist can't always be around to explain defects.
- All of these systems are *allowed to make mistakes* at some non-negotiable rate.
- None of these systems are *concerned with cause*. They're successful when they find useful correlations and are not held to correctly sorting cause from effect.

This book teaches the principles and tools needed to build systems like these. We teach the common tasks, steps, and tools used to successfully deliver such projects. Our emphasis is on the whole process—project management, working with others, and presenting results to nonspecialists.

Roadmap

This book covers the following:

- Managing the data science process itself. The data scientist must have the ability to measure and track their own project.
- Applying many of the most powerful statistical and machine learning techniques used in data science projects. Think of this book as a series of explicitly worked exercises in using the R programming language to perform actual data science work.
- Preparing presentations for the various stakeholders: management, users, deployment team, and so on. You must be able to explain your work in concrete terms to mixed audiences with words in their common usage, *not in whatever technical definition is insisted on in a given field*. You can't get away with just throwing data science project results over the fence.

We've arranged the book topics in an order that we feel increases understanding. The material is organized as follows.

Part 1 describes the basic goals and techniques of the data science process, emphasizing collaboration and data. Chapter 1 discusses how to work as a data scientist. Chapter 2 works through loading data into R and shows how to start working with R.

Chapter 3 teaches what to first look for in data and the important steps in characterizing and understanding data. Data must be prepared for analysis, and data issues will need to be corrected. Chapter 4 demonstrates how to correct the issues identified in chapter 3.

Chapter 5 covers one more data preparation step: basic data wrangling. Data is not always available to the data scientist in a form or “shape” best suited for analysis. R provides many tools for manipulating and reshaping data into the appropriate structure; they are covered in this chapter.

Part 2 moves from characterizing and preparing data to building effective predictive models. Chapter 6 supplies a mapping of business needs to technical evaluation and modeling techniques. It covers the standard metrics and procedures used to evaluate model performance, and one specialized technique, LIME, for explaining specific predictions made by a model.

Chapter 7 covers basic linear models: linear regression, logistic regression, and regularized linear models. Linear models are the workhorses of many analytical tasks, and are especially helpful for identifying key variables and gaining insight into the structure of a problem. A solid understanding of them is immensely valuable for a data scientist.

Chapter 8 temporarily moves away from the modeling task to cover more advanced data treatment: how to prepare messy real-world data for the modeling step. Because understanding how these data treatment methods work requires some understanding of linear models and of model evaluation metrics, it seemed best to defer this topic until part 2.

Chapter 9 covers unsupervised methods: modeling methods that do not use labeled training data. Chapter 10 covers more advanced modeling methods that increase prediction performance and fix specific modeling issues. The topics covered include tree-based ensembles, generalized additive models, and support vector machines.

Part 3 moves away from modeling and back to process. We show how to deliver results. Chapter 11 demonstrates how to manage, document, and deploy your models. You’ll learn how to create effective presentations for different audiences in chapter 12.

The appendixes include additional technical details about R, statistics, and more tools that are available. Appendix A shows how to install R, get started working, and work with other tools (such as SQL). Appendix B is a refresher on a few key statistical ideas.

The material is organized in terms of goals and tasks, bringing in tools as they’re needed. The topics in each chapter are discussed in the context of a representative project with an associated dataset. You’ll work through a number of substantial projects

over the course of this book. All the datasets referred to in this book are at the book's GitHub repository, <https://github.com/WinVector/PDSwR2>. You can download the entire repository as a single zip file (one of GitHub's services), clone the repository to your machine, or copy individual files as needed.

Audience

To work the examples in this book, you'll need some familiarity with R and statistics. We recommend you have some good introductory texts already on hand. You don't need to be expert in R before starting the book, but you will need to be familiar with it.

To start with R, we recommend *Beyond Spreadsheets with R* by Jonathan Carroll (Manning, 20108) or *R in Action* by Robert Kabacoff (now available in a second edition: <http://www.manning.com/kabacoff2/>), along with the text's associated website, *Quick-R* (<http://www.statmethods.net>). For statistics, we recommend *Statistics*, Fourth Edition, by David Freedman, Robert Pisani, and Roger Purves (W. W. Norton & Company, 2007).

In general, here's what we expect from our ideal reader:

- *An interest in working examples.* By working through the examples, you'll learn at least one way to perform all steps of a project. You must be willing to attempt simple scripting and programming to get the full value of this book. For each example we work, you should try variations and expect both some failures (where your variations don't work) and some successes (where your variations outperform our example analyses).
- *Some familiarity with the R statistical system and the will to write short scripts and programs in R.* In addition to Kabacoff, we list a few good books in appendix C. We'll work specific problems in R; you'll need to run the examples and read additional documentation to understand variations of the commands we didn't demonstrate.
- *Some comfort with basic statistical concepts such as probabilities, means, standard deviations, and significance.* We'll introduce these concepts as needed, but you may need to read additional references as we work through examples. We'll define some terms and refer to some topic references and blogs where appropriate. But we expect you will have to perform some of your own internet searches on certain topics.
- *A computer (macOS, Linux, or Windows) to install R and other tools on, as well as internet access to download tools and datasets.* We strongly suggest working through the examples, examining R `help()` on various methods, and following up with some of the additional references.

What is not in this book?

- *This book is not an R manual.* We use R to concretely demonstrate the important steps of data science projects. We teach enough R for you to work through the examples, but a reader unfamiliar with R will want to refer to appendix A as well as to the many excellent R books and tutorials already available.
- *This book is not a set of case studies.* We emphasize methodology and technique. Example data and code is given only to make sure we're giving concrete, usable advice.
- *This book is not a big data book.* We feel most significant data science occurs at a database or file manageable scale (often larger than memory, but still small enough to be easy to manage). Valuable data that maps measured conditions to dependent outcomes tends to be expensive to produce, and that tends to bound its size. For some report generation, data mining, and natural language processing, you'll have to move into the area of big data.
- *This is not a theoretical book.* We don't emphasize the absolute rigorous theory of any one technique. The goal of data science is to be flexible, have a number of good techniques available, and be willing to research a technique more deeply if it appears to apply to the problem at hand. We prefer R code notation over beautifully typeset equations even in our text, as the R code can be directly used.
- *This is not a machine learning tinkerer's book.* We emphasize methods that are already implemented in R. For each method, we work through the theory of operation and show where the method excels. We usually don't discuss how to implement them (even when implementation is easy), as excellent R implementations are already available.

Code conventions and downloads

This book is example driven. We supply prepared example data at the GitHub repository (<https://github.com/WinVector/PDSwR2>), with R code and links back to original sources. You can explore this repository online or clone it onto your own machine. We also supply the code to produce all results and almost all graphs found in the book as a zip file (<https://github.com/WinVector/PDSwR2/raw/master/CodeExamples.zip>), since copying code from the zip file can be easier than copying and pasting from the book. Instructions on how to download, install, and get started with all the suggested tools and example data can be found in appendix A, in section A.1.

We encourage you to try the example R code as you read the text; even when we're discussing fairly abstract aspects of data science, we'll illustrate examples with concrete data and code. Every chapter includes links to the specific dataset(s) that it references.

In this book, code is set with a fixed-width font like `this` to distinguish it from regular text. Concrete variables and values are formatted similarly, whereas abstract math will be in *italic font like this*. R code is written without any command-line prompts such as `>` (which is often seen when displaying R code, but not to be typed in as new R code). Inline results are prefixed by R's comment character `#`. In many cases, the

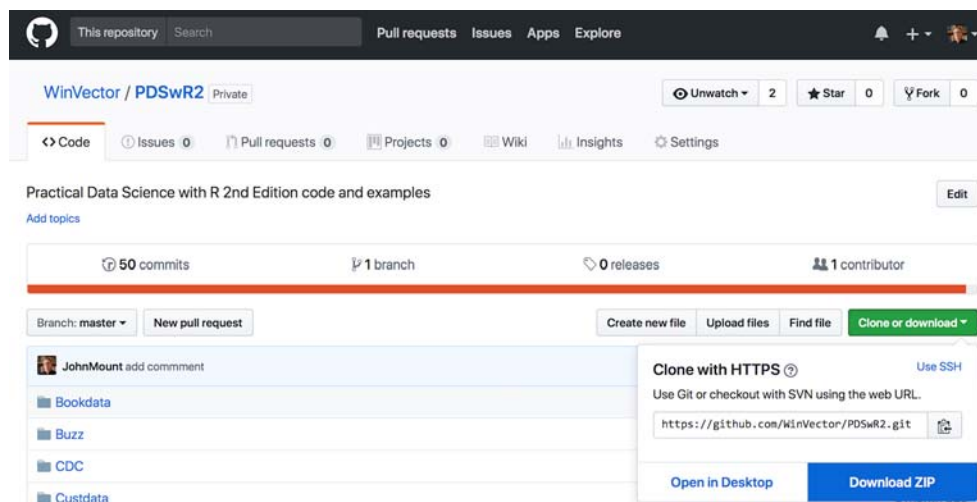
original source code has been reformatted; we’ve added line breaks and reworked indentation to accommodate the available page space in the book. In rare cases, even this was not enough, and listings include line-continuation markers (`\>`). Additionally, comments in the source code have often been removed from the listings when the code is described in the text. Code annotations accompany many of the listings, highlighting important concepts.

Working with this book

Practical Data Science with R is best read while working at least some of the examples. To do this we suggest you install R, RStudio, and the packages commonly used in the book. We share instructions on how to do this in section A.1 of appendix A. We also suggest you download all the examples, which include code and data, from our GitHub repository at <https://github.com/WinVector/PDSwR2>.

DOWNLOADING THE BOOK’S SUPPORTING MATERIALS/REPOSITORY

The contents of the repository can be downloaded as a zip file by using the “download as zip” GitHub feature, as shown in the following figure, from the GitHub URL <https://github.com/WinVector/PDSwR2>.



GitHub download example

Clicking on the “Download ZIP” link should download the compressed contents of the package (or you can try a direct link to the ZIP material: <https://github.com/WinVector/PDSwR2/archive/master.zip>). Or, if you are familiar with working with the Git source control system from the command line, you can do this with the following command from a Bash shell (not from R):

```
git clone https://github.com/WinVector/PDSwR2.git
```

In all examples, we assume you have either cloned the repository or downloaded and unzipped the contents. This will produce a directory named PDSwR2. Paths we discuss will start with this directory. For example, if we mention working with PDSwR2/UCI-Car, we mean to work with the contents of the UCICar subdirectory of wherever you unpacked PDSwR2. You can change R's working directory through the `setwd()` command (please type `help(setwd)` in the R console for some details). Or, if you are using RStudio, the file-browsing pane can also set the working directory from an option on the pane's gear/more menu. All of the code examples from this book are included in the directory PDSwR2/CodeExamples, so you should not need to type them in (though to run them you will have to be working in the appropriate data directory—not in the directory you find the code in).

The examples in this book are supplied in lieu of explicit exercises. We suggest working through the examples *and* trying variations. For example, in section 2.3.1, where we show how to relate expected income to schooling and gender, it makes sense to try relating income to employment status or even age. Data science requires curiosity about programming, functions, data, variables, and relations, and the earlier you find surprises in your data, the easier they are to work through.

Book forum

Purchase of *Practical Data Science with R* includes free access to a private web forum run by Manning Publications where you can make comments about the book, ask technical questions, and receive help from the author and from other users. To access the forum, go to <https://forums.manning.com/forums/practical-data-science-with-r-second-edition>. You can also learn more about Manning's forums and the rules of conduct at <https://forums.manning.com/forums/about>.

Manning's commitment to our readers is to provide a venue where a meaningful dialogue between individual readers and between readers and the authors can take place. It is not a commitment to any specific amount of participation on the part of the authors, whose contribution to the forum remains voluntary (and unpaid). We suggest you try asking them some challenging questions lest their interest stray! The forum and the archives of previous discussions will be accessible from the publisher's website as long as the book is in print.

about the authors



Nina Zumel has worked as a scientist at SRI International, an independent, nonprofit research institute. She has worked as chief scientist of a price optimization company and founded a contract research company. Nina is now a principal consultant at Win-Vector LLC. She can be reached at nzumel@win-vector.com.



John Mount has worked as a computational scientist in biotechnology and as a stock trading algorithm designer, and has managed a research team for Shopping.com. He is now a principal consultant at Win-Vector LLC. John can be reached at jmount@win-vector.com.

about the foreword authors

JEREMY HOWARD is an entrepreneur, business strategist, developer, and educator. Jeremy is a founding researcher at fast.ai, a research institute dedicated to making deep learning more accessible. He is also a faculty member at the University of San Francisco, and is chief scientist at doc.ai and platform.ai.

Previously, Jeremy was the founding CEO of Enlitic, which was the first company to apply deep learning to medicine, and was selected as one of the world's top 50 smartest companies by MIT Tech Review two years running. He was the president and chief scientist of the data science platform Kaggle, where he was the top-ranked participant in international machine learning competitions two years running.

RACHEL THOMAS is director of the USF Center for Applied Data Ethics and cofounder of fast.ai, which has been featured in *The Economist*, *MIT Tech Review*, and *Forbes*. She was selected by *Forbes* as one of 20 Incredible Women in AI, earned her math PhD at Duke, and was an early engineer at Uber. Rachel is a popular writer and keynote speaker. In her TEDx talk, she shares what scares her about AI and why we need people from all backgrounds involved with AI.

about the cover illustration

The figure on the cover of *Practical Data Science with R* is captioned “Habit of a Lady of China in 1703.” The illustration is taken from Thomas Jefferys’ *A Collection of the Dresses of Different Nations, Ancient and Modern* (four volumes), London, published between 1757 and 1772. The title page states that these are hand-colored copperplate engravings, heightened with gum arabic. Thomas Jefferys (1719–1771) was called “Geographer to King George III.” He was an English cartographer who was the leading map supplier of his day. He engraved and printed maps for government and other official bodies and produced a wide range of commercial maps and atlases, especially of North America. His work as a mapmaker sparked an interest in local dress customs of the lands he surveyed and mapped; they are brilliantly displayed in this four-volume collection.

Fascination with faraway lands and travel for pleasure were relatively new phenomena in the eighteenth century, and collections such as this one were popular, introducing both the tourist as well as the armchair traveler to the inhabitants of other countries. The diversity of the drawings in Jefferys’ volumes speaks vividly of the uniqueness and individuality of the world’s nations centuries ago. Dress codes have changed, and the diversity by region and country, so rich at that time, has faded away. It is now often hard to tell the inhabitant of one continent from another. Perhaps, viewing it optimistically, we have traded a cultural and visual diversity for a more varied personal life—or a more varied and interesting intellectual and technical life.

At a time when it is hard to tell one computer book from another, Manning celebrates the inventiveness and initiative of the computer business with book covers based on the rich diversity of national costumes three centuries ago, brought back to life by Jefferys’ pictures.

Part 1

Introduction to data science

In part 1, we concentrate on the most essential tasks in data science: working with your partners, defining your problem, and examining your data.

Chapter 1 covers the lifecycle of a typical data science project. We look at the different roles and responsibilities of project team members, the different stages of a typical project, and how to define goals and set project expectations. This chapter serves as an overview of the material that we cover in the rest of the book, and is organized in the same order as the topics that we present.

Chapter 2 dives into the details of loading data into R from various external formats and transforming the data into a format suitable for analysis. It also discusses the most important R data structure for a data scientist: the data frame. More details about the R programming language are covered in appendix A.

Chapters 3 and 4 cover the data exploration and treatment that you should do before proceeding to the modeling stage. In chapter 3, we discuss some of the typical problems and issues that you'll encounter with your data and how to use summary statistics and visualization to detect those issues. In chapter 4, we discuss data treatments that will help you deal with the problems and issues in your data. We also recommend some habits and procedures that will help you better manage the data throughout the different stages of the project.

Chapter 5 covers how to wrangle or manipulate data into a ready-for-analysis shape.

On completing part 1, you'll understand how to define a data science project, and you'll know how to load data into R and prepare it for modeling and analysis.

The data science process



This chapter covers

- Defining data science
- Defining data science project roles
- Understanding the stages of a data science project
- Setting expectations for a new data science project

Data science is a cross-disciplinary practice that draws on methods from data engineering, descriptive statistics, data mining, machine learning, and predictive analytics. Much like operations research, data science focuses on implementing data-driven decisions and managing their consequences. For this book, we will concentrate on data science as applied to business and scientific problems, using these techniques.

The data scientist is responsible for guiding a data science project from start to finish. Success in a data science project comes not from access to any one exotic tool, but from having quantifiable goals, good methodology, cross-discipline interactions, and a repeatable workflow.

This chapter walks you through what a typical data science project looks like: the kinds of problems you encounter, the types of goals you should have, the tasks that you're likely to handle, and what sort of results are expected.

We'll use a concrete, real-world example to motivate the discussion in this chapter.¹

Example *Suppose you're working for a German bank. The bank feels that it's losing too much money to bad loans and wants to reduce its losses. To do so, they want a tool to help loan officers more accurately detect risky loans.*

This is where your data science team comes in.

1.1 The roles in a data science project

Data science is not performed in a vacuum. It's a collaborative effort that draws on a number of roles, skills, and tools. Before we talk about the process itself, let's look at the roles that must be filled in a successful project. Project management has been a central concern of software engineering for a long time, so we can look there for guidance. In defining the roles here, we've borrowed some ideas from Fredrick Brooks' "surgical team" perspective on software development, as described in *The Mythical Man-Month: Essays on Software Engineering* (Addison-Wesley, 1995). We also borrowed ideas from the agile software development paradigm.

1.1.1 Project roles

Let's look at a few recurring roles in a data science project in table 1.1.

Table 1.1 Data science project roles and responsibilities

Role	Responsibilities
Project sponsor	Represents the business interests; champions the project
Client	Represents end users' interests; domain expert
Data scientist	Sets and executes analytic strategy; communicates with sponsor and client
Data architect	Manages data and data storage; sometimes manages data collection
Operations	Manages infrastructure; deploys final project results

Sometimes these roles may overlap. Some roles—in particular, client, data architect, and operations—are often filled by people who aren't on the data science project team, but are key collaborators.

PROJECT SPONSOR

The most important role in a data science project is the project sponsor. The sponsor is the person who wants the data science result; generally, they represent the business interests.

¹ For this chapter, we'll use a credit dataset donated by Dr. Hans Hofmann, professor of integrative biology, to the UCI Machine Learning Repository in 1994. We've simplified some of the column names for clarity. The original dataset can be found at [http://archive.ics.uci.edu/ml/datasets/Statlog+\(German+Credit+Data\)](http://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data)). We'll show how to load this data and prepare it for analysis in chapter 2. Note that the German currency at the time of data collection was the deutsche mark (DM).

In the loan application example, the sponsor might be the bank's head of Consumer Lending. The sponsor is responsible for deciding whether the project is a success or failure. The data scientist may fill the sponsor role for their own project if they feel they know and can represent the business needs, but that's not the optimal arrangement. The ideal sponsor meets the following condition: if they're satisfied with the project outcome, then the project is by definition a success. *Getting sponsor sign-off becomes the central organizing goal of a data science project.*

KEEP THE SPONSOR INFORMED AND INVOLVED It's critical to keep the sponsor informed and involved. Show them plans, progress, and intermediate successes or failures in terms they can understand. A good way to guarantee project failure is to keep the sponsor in the dark.

To ensure sponsor sign-off, you must get clear goals from them through directed interviews. You attempt to capture the sponsor's expressed goals as quantitative statements. An example goal might be "Identify 90% of accounts that will go into default at least two months before the first missed payment with a false positive rate of no more than 25%." This is a precise goal that allows you to check in parallel if meeting the goal is actually going to make business sense and whether you have data and tools of sufficient quality to achieve the goal.

CLIENT

While the sponsor is the role that represents the business interests, the client is the role that represents the model's end users' interests. Sometimes, the sponsor and client roles may be filled by the same person. Again, the data scientist may fill the client role if they can weight business trade-offs, but this isn't ideal.

The client is more hands-on than the sponsor; they're the interface between the technical details of building a good model and the day-to-day work process into which the model will be deployed. They aren't necessarily mathematically or statistically sophisticated, but are familiar with the relevant business processes and serve as the domain expert on the team. In the loan application example, the client may be a loan officer or someone who represents the interests of loan officers.

As with the sponsor, you should keep the client informed and involved. Ideally, you'd like to have regular meetings with them to keep your efforts aligned with the needs of the end users. Generally, the client belongs to a different group in the organization and has other responsibilities beyond your project. Keep meetings focused, present results and progress in terms they can understand, and take their critiques to heart. If the end users can't or won't use your model, then the project isn't a success, in the long run.

DATA SCIENTIST

The next role in a data science project is the data scientist, who's responsible for taking all necessary steps to make the project succeed, including setting the project strategy and keeping the client informed. They design the project steps, pick the data sources, and pick the tools to be used. Since they pick the techniques that will be

tried, they have to be well informed about statistics and machine learning. They're also responsible for project planning and tracking, though they may do this with a project management partner.

At a more technical level, the data scientist also looks at the data, performs statistical tests and procedures, applies machine learning models, and evaluates results—the science portion of data science.

Domain empathy

It is often too much to ask for the data scientist to become a domain expert. However, in all cases the data scientist must develop strong *domain empathy* to help define and solve the right problems.

DATA ARCHITECT

The data architect is responsible for all the data and its storage. Often this role is filled by someone outside of the data science group, such as a database administrator or architect. Data architects often manage data warehouses for many different projects, and they may only be available for quick consultation.

OPERATIONS

The operations role is critical both in acquiring data and delivering the final results. The person filling this role usually has operational responsibilities outside of the data science group. For example, if you're deploying a data science result that affects how products are sorted on an online shopping site, then the person responsible for running the site will have a lot to say about how such a thing can be deployed. This person will likely have constraints on response time, programming language, or data size that you need to respect in deployment. The person in the operations role may already be supporting your sponsor or your client, so they're often easy to find (though their time may be already very much in demand).

1.2 Stages of a data science project

The ideal data science environment is one that encourages feedback and iteration between the data scientist and all other stakeholders. This is reflected in the lifecycle of a data science project. Even though this book, like other discussions of the data science process, breaks up the cycle into distinct stages, in reality the boundaries between the stages are fluid, and the activities of one stage will often overlap those of other stages.² Often, you'll loop back and forth between two or more stages before moving forward in the overall process. This is shown in figure 1.1.

² One common model of the machine learning process is the cross-industry standard process for data mining (CRISP-DM) (https://en.wikipedia.org/wiki/Cross-industry_standard_process_for_data_mining). The model we'll discuss here is similar, but emphasizes that back-and-forth is possible at any stage of the process.

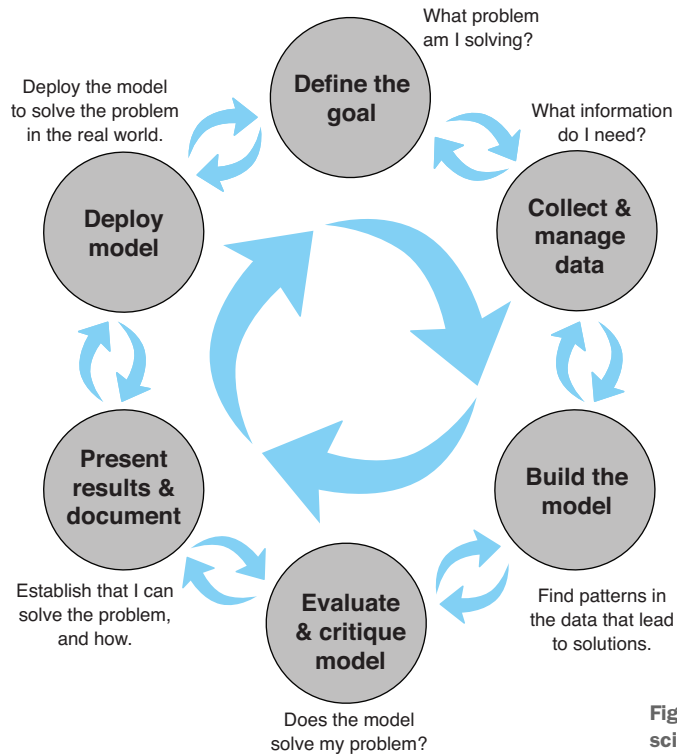


Figure 1.1 The lifecycle of a data science project: loops within loops

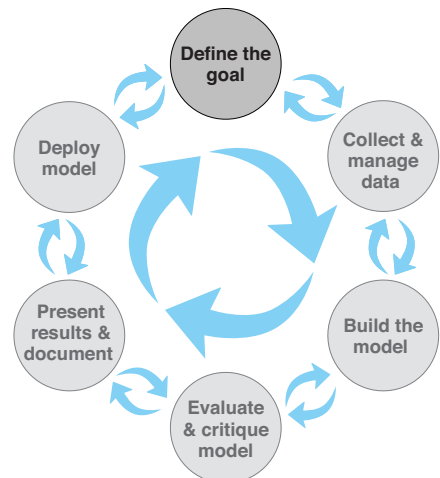
Even after you complete a project and deploy a model, new issues and questions can arise from seeing that model in action. The end of one project may lead into a follow-up project.

Let's look at the different stages shown in figure 1.1.

1.2.1 Defining the goal

The first task in a data science project is to define a measurable and quantifiable goal. At this stage, learn all that you can about the context of your project:

- Why do the sponsors want the project in the first place? What do they lack, and what do they need?
- What are they doing to solve the problem now, and why isn't that good enough?
- What resources will you need: what kind of data and how much staff? Will



you have domain experts to collaborate with, and what are the computational resources?

- How do the project sponsors plan to deploy your results? What are the constraints that have to be met for successful deployment?

Let's come back to our loan application example. The ultimate business goal is to reduce the bank's losses due to bad loans. Your project sponsor envisions a tool to help loan officers more accurately score loan applicants, and so reduce the number of bad loans made. At the same time, it's important that the loan officers feel that they have final discretion on loan approvals.

Once you and the project sponsor and other stakeholders have established preliminary answers to these questions, you and they can start defining the precise goal of the project. The goal should be specific and measurable; not "We want to get better at finding bad loans," but instead "We want to reduce our rate of loan charge-offs by at least 10%, using a model that predicts which loan applicants are likely to default."

A concrete goal leads to concrete stopping conditions and concrete acceptance criteria. The less specific the goal, the likelier that the project will go unbounded, because no result will be "good enough." If you don't know what you want to achieve, you don't know when to stop trying—or even what to try. When the project eventually terminates—because either time or resources run out—no one will be happy with the outcome.

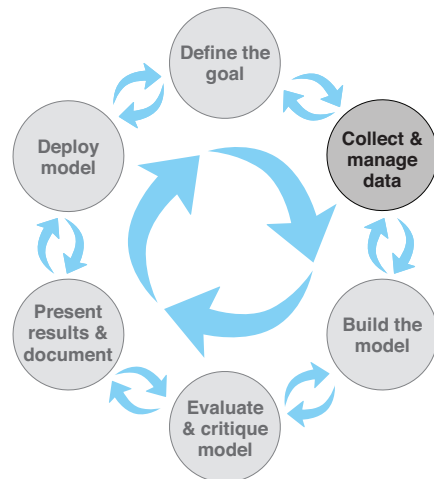
Of course, at times there is a need for looser, more exploratory projects: "Is there something in the data that correlates to higher defaults?" or "Should we think about reducing the kinds of loans we give out? Which types might we eliminate?" In this situation, you can still scope the project with concrete stopping conditions, such as a time limit. For example, you might decide to spend two weeks, and no more, exploring the data, with the goal of coming up with candidate hypotheses. These hypotheses can then be turned into concrete questions or goals for a full-scale modeling project.

Once you have a good idea of the project goals, you can focus on collecting data to meet those goals.

1.2.2 *Data collection and management*

This step encompasses identifying the data you need, exploring it, and conditioning it to be suitable for analysis. This stage is often the most time-consuming step in the process. It's also one of the most important:

- What data is available to me?
- Will it help me solve the problem?
- Is it enough?
- Is the data quality good enough?



Imagine that, for your loan application problem, you've collected a sample of representative loans from the last decade. Some of the loans have defaulted; most of them (about 70%) have not. You've collected a variety of attributes about each loan application, as listed in table 1.2.

Table 1.2 Loan data attributes

Status_of_existing_checking_account	(at time of application)
Duration_in_month	(loan length)
Credit_history	
Purpose	(car loan, student loan, and so on)
Credit_amount	(loan amount)
Savings_Account_or_bonds	(balance/amount)
Present_employment_since	
Installment_rate_in_percentage_of_disposable_income	
Personal_status_and_sex	
Cosigners	
Present_residence_since	
Collateral	(car, property, and so on)
Age_in_years	
Other_installment_plans	(other loans/lines of credit—the type)
Housing	(own, rent, and so on)
Number_of_existing_credits_at_this_bank	
Job	(employment type)
Number_of_dependents	
Telephone	(do they have one)
Loan_status	(dependent variable)

In your data, `Loan_status` takes on two possible values: `GoodLoan` and `BadLoan`. For the purposes of this discussion, assume that a `GoodLoan` was paid off, and a `BadLoan` defaulted.

TRY TO DIRECTLY MEASURE THE INFORMATION YOU NEED As much as possible, try to use information that can be directly measured, rather than information that is inferred from another measurement. For example, you might be tempted to use income as a variable, reasoning that a lower income implies more difficulty paying off a loan. The ability to pay off a loan is more directly measured by considering the size of the loan payments relative to the borrower's disposable income. This information is more useful than income alone; you have it in your data as the variable `Installment_rate_in_percentage_of_disposable_income`.

This is the stage where you initially explore and visualize your data. You'll also clean the data: repair data errors and transform variables, as needed. In the process of exploring and cleaning the data, you may discover that it isn't suitable for your problem, or that you need other types of information as well. You may discover things in the data that raise issues more important than the one you originally planned to address. For example, the data in figure 1.2 seems counterintuitive.