Beginning Data Science in R

Data Analysis, Visualization, and Modelling for the Data Scientist

Thomas Mailund

Apress

Beginning Data Science in R: Data Analysis, Visualization, and Modelling for the Data Scientist

Thomas Mailund
Aarhus, Denmark

ISBN-13 (pbk): 978-1-4842-2670-4 ISBN-13 (electronic): 978-1-4842-2671-1
DOI 10.1007/978-1-4842-2671-1

Library of Congress Control Number: 2017934529

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/9781484226704. For more detailed information, please visit http://www.apress.com/source-code.

Contents at a Glance

Contents

Introduction

Welcome to *Introduction to Data Science with R*. This book was written as a set of lecture notes for two classes I teach, *Data Science: Visualization and Analysis* and *Data Science: Software Development and Testing*. The book is written to fit the structure of these classes, where each class consists of seven weeks of lectures and project work. This means that there are 14 chapters with the core material, where the first seven focus on data analysis and the last seven on developing reusable software for data science.

What Is Data Science?

Oh boy! That is a difficult question. I don't know if it is easy to find someone who is entirely sure what data science is, but I am pretty sure that it would be difficult to find two people with fewer than three opinions about it. It is certainly a popular buzzword, and everyone wants to have data scientists these days, so data science skills are useful to have on the CV. But what *is* it?

Since I can't really give you an agreed upon definition, I will just give you my own: *Data science is the science of learning from data*.

This is a very broad definition—almost too broad to be useful. I realize this. But then, I think data science is an incredibly general field. I don't have a problem with that. Of course, you could argue that any *science* is all about getting information out of data, and you might be right. Although I would say that there is more to science than just transforming raw data into useful information. The sciences are focusing on answering specific questions about the world while data science is focusing on how to manipulate data efficiently and effectively. The primary focus is not which questions to ask of the data but how we can answer them, whatever they may be. It is more like computer science and mathematics than it is like natural sciences, in this way. It isn't so much about studying the natural world as it is about how to compute data efficiently.

Included in data science is the design of experiments. With the right data, we can address the questions we are interested in. With a poor design of experiments or a poor choice of which data we gather, this can be difficult. Study design might be the most important aspect of data science, but is not the topic of this book. In this book I focus on the analysis of data, once gathered.

Computer science is also mainly the study of computations—as is hinted at in the name—but is a bit broader in this focus. Although *datalogy*, an earlier name for data science, was also suggested for computer science, and for example in Denmark it *is* the name for computer science, using the name "computer science" puts the focus on computation while using the name "data science" puts the focus on data. But of course, the fields overlap. If you are writing a sorting algorithm, are you then focusing on the computation or the data? Is that even a meaningful question to ask?

There is a huge overlap between computer science and data science and naturally the skillsets you need overlap as well. To efficiently manipulate data you need the tools for doing that, so computer programming skills are a must and some knowledge about algorithms and data structures usually is as well. For data science, though, the focus is always on the data. In a data analysis project, the focus is on how the data flows from its raw form through various manipulations until it is summarized in some useful form. Although the difference can be subtle, the focus is not about what operations a program does during the analysis, but about how the data flows and is transformed. It is also focused on *why* we do certain transformations of the

data, what purpose those changes serve, and how they help us gain knowledge about the data. It is as much about deciding what to do with the data as it is about how to do it efficiently.

Statistics is of course also closely related to data science. So closely linked, in fact, that many consider data science just a fancy word for statistics that looks slightly more modern and sexy. I can't say that I strongly disagree with this—data science *does* sound sexier than statistics—but just as data science is slightly different from computer science, data science is also slightly different from statistics. Just, perhaps, somewhat less different than computer science is.

A large part of doing statistics is building mathematical models for your data and fitting the models to the data to learn about the data in this way. That is also what we do in data science. As long as the focus is on the data, I am happy to call statistics data science. If the focus changes to the models and the mathematics, then we are drifting away from data science into something else—just as if the focus changes from the data to computations we are drifting from data science to computer science.

Data science is also related to machine learning and artificial intelligence, and again there are huge overlaps. Perhaps not surprising since something like machine learning has its home both in computer science and in statistics; if it is focusing on data analysis, it is also at home in data science. To be honest, it has never been clear to me when a mathematical model changes from being a plain old statistical model to becoming machine learning anyway.

For this book, we are just going to go with my definition and, as long as we are focusing on analyzing data, we are going to call it data science.

Prerequisites for Reading this Book

In the first seven chapters in this book, the focus is on data analysis and not programming. For those seven chapters, I do not assume a detailed familiarity with topics such as software design, algorithms, data structures, and such. I do not expect you to have any experience with the R programming language either. I do, however, expect that you have had *some* experience with programming, mathematical modeling, and statistics.

Programming R can be quite tricky at times if you are familiar with a scripting language or object-oriented languages. R is a functional language that does not allow you to modify data, and while it does have systems for object-oriented programming, it handles this programming paradigm very differently from languages you are likely to have seen such as Java or Python.

For the data analysis part of this book, the first seven chapters, we will only use R for very straightforward programming tasks, so none of this should pose a problem. We will have to write simple scripts for manipulating and summarizing data so you should be familiar with how to write basic expressions like function calls, `if` statements, loops, and so on. These things you will have to be comfortable with. I will introduce every such construction in the book when we need them so you will see how they are expressed in R, but I will not spend much time explaining them. I mostly will just expect you to be able to pick it up from examples.

Similarly, I do not expect you to know already how to fit data and compare models in R. I do expect that you have had enough introduction to statistics to be comfortable with basic terms like parameter estimation, model fitting, explanatory and response variables, and model comparison. If not, I expect you to be at least able to pick up what we are talking about when you need to.

I won't expect you to know a lot about statistics and programming, but this isn't *Data Science for Dummies,* so I do expect you to be able to figure out examples without me explaining everything in detail.

After the first seven chapters is a short description of a data analysis project, one of my students did in an earlier class. It shows how such a project could look, but I suggest that you do not wait until you have finished the first seven chapters to start doing such analysis yourself. To get the most benefit out of reading this book, you should be applying what you learn continuously. Already when you begin reading, I suggest that you find a dataset that you would be interested in finding out more about and then apply what you learn in each chapter to that data.

For the final seven chapters of the book, the focus *is* on programming. To read this part you should be familiar with object-oriented programming. I will explain how it is handled in R and how it differs from languages such as Python, Java or C++ but I expect you to be familiar with terms such as class hierarchies, inheritance, and polymorphic methods. I will not expect you to be already familiar with functional programming (but if you are, there should still be plenty to learn in those chapters if you are not already familiar with R programming as well).

Plan for the Book

In the book, we cover basic data manipulation—filtering and selecting relevant data; transforming data into shapes readily analyzable; summarizing data; visualizing data in informative ways both for exploring data and presenting results; and model building. These are the key aspects of doing analysis in data science. After this we will cover how to develop R code that is reusable and works well with existing packages, and that is easy to extend, and we will see how to build new R packages that other people will be able to use in their projects. These are the essential skills you will need to develop your own methods and share them with the world.

We will do all this using the programming language R (`https://www.r-project.org/about.html`). R is one of the most popular (and open source) data analysis programming languages around at the moment. Of course, popularity doesn't imply quality, but because R is so popular it has a rich ecosystem of extensions (called "packages" in R) for just about any kind of analysis you could be interested in. People who develop statistical methods often implement them as R packages, so you can quite often get the state of the art techniques very easily in R. The popularity also means that there is a large community of people who can help if you have problems. Most problems you run into can be solved with a few minutes on Google because you are unlikely to be the first to run into any particular issue. There are also plenty of online tutorials for learning more about R and specialized packages, there are plenty of videos with talks about R and popular R packages, and there are plenty of books you can buy if you want to learn more.

Data Analysis and Visualization

The topics focusing on data analysis and visualization are covered in the first seven chapters:

- Chapter 1, Introduction to R programming. In which you learn how to work with data and write data pipelines.
- Chapter 2, **Reproducible** analysis. In which you find out how to integrate documentation and analysis in a single document and how to use such documents to produce reproducible research.
- Chapter 3, Data manipulation. In which you learn how to import, tidy up, and transform data, and compute summaries from data.
- Chapter 4, **Visualizing** and exploring data. In which you learn how to make plots for exploring data features and for presenting data features and analysis results.
- Chapter 5, **Working** with large datasets. In which you learn how to deal with data where the number of observations make the usual approaches too slow.
- Chapter 6, **Supervised** learning. In which you learn how to train models when you have datasets with known classes or regression values.
- Chapter 7, **Unsupervised** learning. In which you learn how to search for patterns you are not aware of in data.

These chapters are followed by the first project, where you see the various techniques in use.

Software Development

Software and package development is then covered in the following seven chapters:

- Chapter 8, **More** R programming. In which you'll return to the basics of R programming and get a few more details than the tutorial in Chapter 1.
- Chapter 9, **Advanced** R **programming**. In which you explore more advanced features of the R programming language, in particular, functional programming.
- Chapter 10, **Object** oriented programming. In which you learn how R models object orientation and how you can use it to write more generic code.
- Chapter 11, B**uilding** an R package. In which you learn the necessary components of an R package and how to program your own.
- Chapter 12, T**esting** and checking. In which you learn techniques for testing your R code and checking the consistency of your R packages.
- Chapter 13, **Version** control. In which you learn how to manage code under version control and how to collaborate using GitHub.
- Chapter 14, **Profiling** and optimizing. In which you learn how to identify hotspots of code where inefficient solutions are slowing you down and techniques for alleviating this.

These chapters are then followed by the second project, where you'll build a package for Bayesian linear regression.

Getting R and RStudio

You will need to install R on your computer to do the exercises in this book. I suggest that you get an integrated environment since it can be slightly easier to keep track of a project when you have your plots, documentation, code, etc., all in the same program.

I personally use RStudio (`https://www.rstudio.com/products/RStudio`), which I warmly recommend. You can get it for free—just follow the link—and I will assume that you have it when I need to refer to the software environment you are using in the following chapters. There won't be much RStudio specifics, though, and most tools for working with R have the same features, so if you want to use something else you can probably follow the notes without any difficulties.

Projects

You cannot learn how to analyze data without analyzing data, and you cannot learn how to develop software without developing software either. Typing in examples from the book is nothing like writing code on your own. Even doing exercises from the book—which you really ought to do—is not the same as working on your own projects. Exercises, after all, cover small isolated aspects of problems you have just been introduced to. In the real world, there is not a chapter of material presented before every task you have to deal with. You need to work out by yourself what needs to be done and how. If you only do the exercises in this book, you will miss the most important lessons in analyzing data. How to explore the data and get a feeling for it; how to do the detective work necessary to pull out some understanding from the data; and how to deal with all the noise and weirdness found in any dataset. And for developing a package, you need to think through how to design and implement its functionality so that the various functions and data structures fit well together.

In this book, I go through a data analysis project to show you what that can look like. To actually learn how to analyze data, you need to do it yourself as well, and you need to do it with a dataset that I haven't analyzed for you. You might have a dataset lying around you have worked on before, a dataset from something you are just interested in, or you can probably find something interesting at a public data repository, e.g., one of these:

- RDataMining.com
- UCI machine learning repository (`http://archive.ics.uci.edu/ml/`)
- KDNuggets (`http://www.kdnuggets.com/datasets/index.html`)
- Reddit r/datasets (`https://www.reddit.com/r/datasets`)
- GitHub awesome public datasets (`https://github.com/caesar0301/awesome-public-datasets`)

I suggest that you find yourself a dataset and that after each lesson, you use the skills you have learned to explore this dataset. Pick data that is structured as a table with observations as rows and variables as columns, since that is the form of the data we consider in this book. At the end of the first seven chapters, you will have analyzed this data, you can write a report about your analysis that others can evaluate to follow and maybe modify it. You will be doing reproducible science.

For the programming topics, I describe another project illustrating the design and implementation issues involved in making an R package. There, you should be able to learn from just implementing your own version of the project I use, but you will, of course, be more challenged by working on a project without any of my help at all. Whichever you do, to get the full benefit of this book you should make your own package while reading the programming chapters.

Introduction to R Programming

We will use R for our data analysis so we need to know the basics of programming in the R language. R is a full programming language with both functional programming and object oriented programming features. Learning the language is far beyond the scope of this chapter and is something we return to later. The good news, though, is that to use R for data analysis, you rarely need to do much programming. At least, if you do the *right* kind of programming, you won't need much.

For manipulating data—and how to do this is the topic of the next chapter—you mainly just have to string together a couple of operations. Operations such as "group the data by this feature" followed by "calculate the mean value of these features within each group" and then "plot these means". This used to be much more complicated to do in R, but a couple of new ideas on how to structure such data flow—and some clever implementations of these in a couple of packages such as `magrittr` and `dplyr`—has significantly simplified it. We will see some of this at the end of this chapter and more in the next chapter. First, though, you need to get a taste for R.

Basic Interaction with R

Start by downloading RStudio if you haven't done so already (https://www.rstudio.com/products/RStudio). If you open it, you should see a window similar to Figure 1-1. Well, except that you will be in an empty project while the figure shows (on the top right) that this RStudio is opened in a project called "Data Science". You always want to be working on a project. Projects keep track of the state of your analysis by remembering variables and functions you have written and keep track of which files you have opened and such. Choose File ➤ New Project to create a project. You can create a project from an existing directory, but if this is the first time you are working with R you probably just want to create an empty project in a new directory, so do that.