Corrado Caudek

Data Science per psicologi

To Jung Jae-sung (1982 – 2018),

a remarkably hard-working badminton player with a
remarkably simple playing style

Contents

List of Figures

List of Tables

Preface

The document format "R Markdown" was first introduced in the **knitr** package (Xie, 2015, 2021a) in early 2012. The idea was to embed code chunks (of R or other languages) in Markdown documents. In fact, **knitr** supported several authoring languages from the beginning in addition to Markdown, including LaTeX, HTML, AsciiDoc, reStructuredText, and Textile. Looking back over the five years, it seems to be fair to say that Markdown has become the most popular document format, which is what we expected. The simplicity of Markdown clearly stands out among these document formats.

However, the original version of Markdown invented by John Gruber[1] was often found overly simple and not suitable to write highly technical documents. For example, there was no syntax for tables, footnotes, math expressions, or citations. Fortunately, John MacFarlane created a wonderful package named Pandoc (`http://pandoc.org`) to convert Markdown documents (and many other types of documents) to a large variety of output formats. More importantly, the Markdown syntax was significantly enriched. Now we can write more types of elements with Markdown while still enjoying its simplicity.

In a nutshell, R Markdown stands on the shoulders of **knitr** and Pandoc. The former executes the computer code embedded in Markdown, and converts R Markdown to Markdown. The latter renders Markdown to the output format you want (such as PDF, HTML, Word, and so on).

The **rmarkdown** package (Allaire et al., 2021) was first created in early 2014. During the past four years, it has steadily evolved into a relatively complete ecosystem for authoring documents, so it is a good time for us to provide a definitive guide to this ecosystem now. At this point, there are a large number of tasks that you could do with R Markdown:

[1] `https://en.wikipedia.org/wiki/Markdown`

- Compile a single R Markdown document to a report in different formats, such as PDF, HTML, or Word.
- Create notebooks in which you can directly run code chunks interactively.
- Make slides for presentations (HTML5, LaTeX Beamer, or PowerPoint).
- Produce dashboards with flexible, interactive, and attractive layouts.
- Build interactive applications based on Shiny.
- Write journal articles.
- Author books of multiple chapters.
- Generate websites and blogs.

There is a fundamental assumption underneath R Markdown that users should be aware of: we assume it suffices that only a limited number of features are supported in Markdown. By "features", we mean the types of elements you can create with native Markdown. The limitation is a great feature, not a bug. R Markdown may not be the right format for you if you find these elements not enough for your writing: paragraphs, (section) headers, block quotations, code blocks, (numbered and unnumbered) lists, horizontal rules, tables, inline formatting (emphasis, strikeout, superscripts, subscripts, verbatim, and small caps text), LaTeX math expressions, equations, links, images, footnotes, citations, theorems, proofs, and examples. We believe this list of elements suffice for most technical and non-technical documents. It may not be impossible to support other types of elements in R Markdown, but you may start to lose the simplicity of Markdown if you wish to go that far.

Epictetus once said, *"Wealth consists not in having great possessions, but in having few wants."* The spirit is also reflected in Markdown. If you can control your preoccupation with pursuing typesetting features, you should be much more efficient in writing the content and can become a prolific author. It is entirely possible to succeed with simplicity. Jung Jae-sung was a legendary badminton player with a remarkably simple playing style: he did not look like a talented player and was very short compared to other players, so most of the time

you would just see him jump three feet off the ground and smash like thunder over and over again in the back court until he beats his opponents.

Please do not underestimate the customizability of R Markdown because of the simplicity of its syntax. In particular, Pandoc templates can be surprisingly powerful, as long as you understand the underlying technologies such as LaTeX and CSS, and are willing to invest time in the appearance of your output documents (reports, books, presentations, and/or websites). As one example, you may check out the PDF report[2] of the 2017 Employer Health Benefits Survey[3]. It looks fairly sophisticated, but was actually produced via **bookdown** (Xie, 2016), which is an R Markdown extension. A custom LaTeX template and a lot of LaTeX tricks were used to generate this report. Not surprisingly, this very book that you are reading right now was also written in R Markdown, and its full source is publicly available in the GitHub repository `https://github.com/rstudio/rmarkdown-book`.

R Markdown documents are often portable in the sense that they can be compiled to multiple types of output formats. Again, this is mainly due to the simplified syntax of the authoring language, Markdown. The simpler the elements in your document are, the more likely that the document can be converted to different formats. Similarly, if you heavily tailor R Markdown to a specific output format (e.g., LaTeX), you are likely to lose the portability, because not all features in one format work in another format.

Last but not least, your computing results will be more likely to be reproducible if you use R Markdown (or other **knitr**-based source documents), compared to the manual cut-and-paste approach. This is because the results are dynamically generated from computer source code. If anything goes wrong or needs to be updated, you can simply fix or update the source code, compile the document again, and the results will be automatically updated. You can enjoy reproducibility and convenience at the same time.

[2] `http://files.kff.org/attachment/Report-Employer-Health-Benefits-Annual-Survey-2017`

[3] `https://www.kff.org/health-costs/report/2017-employer-health-benefits-survey/`

How to read this book

This book may serve you better as a reference book than a textbook. It contains a large number of technical details, and we do not expect you to read it from beginning to end, since you may easily feel overwhelmed. Instead, think about your background and what you want to do first, and go to the relevant chapters or sections. For example:

- I just want to finish my course homework (Chapter **??** should be more than enough for you).
- I know this is an R Markdown book, but I use Python more than R (Go to Section **??**).
- I want to embed interactive plots in my reports, or want my readers to be able change my model parameters interactively and see results on the fly (Check out Section **??**).
- I know the output format I want to use, and I want to customize its appearance (Check out the documentation of the specific output format in Chapter **??** or Chapter **??**). For example, I want to customize the template for my PowerPoint presentation (Go to Section **??**).
- I want to build a business dashboard highlighting some key figures and indicators (Go to Chapter **??**).
- I heard about `yolo = TRUE` from a friend, and I'm curious what that means in the **xaringan** package (Go to Chapter **??**).
- I want to build a personal website (Go to Chapter **??**), or write a book (Go to Chapter **??**).
- I want to write a paper and submit to the Journal of Statistical Software (Go to Chapter **??**).
- I want to build an interactive tutorial with exercises for my students to learn a topic (Go to Chapter **??**).
- I'm familiar with R Markdown now, and I want to generate personalized reports for all my customers using the same R Markdown template (Try parameterized reports in Chapter **??**).

- I know some JavaScript, and want to build an interface in R to call an interested JavaScript library from R (Learn how to develop HTML widgets in Chapter **??**).
- I want to build future reports with a company branded template that shows our logo and uses our unique color theme (Go to Chapter **??**).

If you are not familiar with R Markdown, we recommend that you read at least Chapter **??** to learn the basics. All the rest of the chapters in this book can be read in any order you desire. They are pretty much orthogonal to each other. However, to become familiar with R Markdown output formats, you may want to thumb through the HTML document format in Section **??**, because many other formats share the same options as this format.

Structure of the book

This book consists of four parts. Part I covers the basics: Chapter 1 introduces how to install the relevant packages, and Chapter **??** is an overview of R Markdown, including the possible output formats, the Markdown syntax, the R code chunk syntax, and how to use other languages in R Markdown.

Part II is the detailed documentation of built-in output formats in the **rmarkdown** package, including document formats and presentation formats.

Part III lists about ten R Markdown extensions that enable you to build different applications or generate output documents with different styles. Chapter **??** introduces the basics of building flexible dashboards with the R package **flexdashboard**. Chapter **??** documents the **tufte** package, which provides a unique document style used by Edward Tufte. Chapter **??** introduces the **xaringan** package for another highly flexible and customizable HTML5 presentation format based on the JavaScript library remark.js. Chapter **??** documents the **revealjs** package, which provides yet another appealing HTML5 presentation format based on the JavaScript library reveal.js. Chapter **??** introduces a few output formats created by the R community, such as the **pret-**

tydoc package, which features lightweight HTML document formats. Chapter **??** teaches you how to build websites using either the **blogdown** package or **rmarkdown**'s built-in site generator. Chapter **??** explains the basics of the **pkgdown** package, which can be used to quickly build documentation websites for R packages. Chapter **??** introduces how to write and publish books with the **bookdown** package. Chapter **??** is an overview of the **rticles** package for authoring journal articles. Chapter **??** introduces how to build interactive tutorials with exercises and/or quiz questions.

Part IV covers other topics about R Markdown, and some of them are advanced (in particular, Chapter **??**). Chapter **??** introduces how to generate different reports with the same R Markdown source document and different parameters. Chapter **??** teaches developers how to build their own HTML widgets for interactive visualization and applications with JavaScript libraries. Chapter **??** shows how to create custom R Markdown and Pandoc templates so that you can fully customize the appearance and style of your output document. Chapter **??** explains how to create your own output formats if the existing formats do not meet your need. Chapter **??** shows how to combine the Shiny framework with R Markdown, so that your readers can interact with the reports by changing the values of certain input widgets and seeing updated results immediately.

Note that this book is intended to be a guide instead of the comprehensive documentation of all topics related to R Markdown. Some chapters are only overviews, and you may need to consult the full documentation elsewhere (often freely available online). Such examples include Chapters **??**, **??**, **??**, **??**, and **??**.

Software information and conventions

The R session information when compiling this book is shown below:

```
xfun::session_info(c(
  'blogdown', 'bookdown', 'knitr', 'rmarkdown', 'htmltools',
  'reticulate', 'rticles', 'flexdashboard', 'learnr', 'shiny',
```

```
  'revealjs', 'pkgdown', 'tinytex', 'xaringan', 'tufte'
), dependencies = FALSE)
```

```
## R version 4.1.2 (2021-11-01)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Big Sur 10.16
##
##   Locale:   it_IT.UTF-8   /   it_IT.UTF-8   /   it_IT.UTF-
8 / C / it_IT.UTF-8 / it_IT.UTF-8
##
## Package version:
##   blogdown_1.7        bookdown_0.24.4
##   flexdashboard_0.5.2 htmltools_0.5.2
##   knitr_1.37          learnr_0.10.1
##   pkgdown_2.0.2       reticulate_1.22
##   revealjs_0.9        rmarkdown_2.11
##   rticles_0.22        shiny_1.7.1
##   tinytex_0.36        tufte_0.11
##   xaringan_0.22
##
## Hugo version: 0.89.1
##
## Pandoc version: 2.17
##
## LaTeX version used:
##   TeX Live 2021 (TinyTeX) with tlmgr 2021-10-04
```

We do not add prompts (> and +) to R source code in this book, and we comment out the text output with two hashes ## by default, as you can see from the R session information above. This is for your convenience when you want to copy and run the code (the text output will be ignored since it is commented out). Package names are in bold text (e.g., **rmarkdown**), and inline code and filenames are formatted in a typewriter font (e.g., `knitr::knit('foo.Rmd')`). Function names are followed by parentheses (e.g., `blogdown::serve_site()`). The double-colon operator `::` means accessing an object from a package.

"Rmd" is the filename extension of R Markdown files, and also an abbreviation of R Markdown in this book.

Acknowledgments

I started writing this book after I came back from the 2018 RStudio Conference in early February, and finished the first draft in early May. This may sound fast for a 300-page book. The main reason I was able to finish it quickly was that I worked full-time on this book for three months. My employer, RStudio, has always respected my personal interests and allowed me to focus on projects that I choose by myself. More importantly, I have been taught several lessons on how to become a professional software engineer since I joined RStudio as a fresh PhD, although the initial journey turned out to be painful.[4] It is a great blessing for me to work in this company.

The other reason for my speed was that JJ and Garrett had already prepared a lot of materials that I could adapt for this book. They had also been offering suggestions as I worked on the manuscript. In addition, Michael Harper[5] contributed the initial drafts of Chapters **??**, **??**, **??**, **??**, and **??**. I would definitely not be able to finish this book so quickly without their help.

The most challenging thing to do when writing a book is to find large blocks of uninterrupted time. This is just so hard. Both others and myself could interrupt me. I do not consider my willpower to be strong: I read random articles, click on the endless links on Wikipedia, look at random Twitter messages, watch people fight on meaningless topics online, reply to emails all the time as if I were able to reach "Inbox Zero", and write random blog posts from time to time. The two most important people in terms of helping keep me on track are Tareef Kawaf (President of RStudio), to whom I report my progress on the weekly basis, and Xu Qin[6], from whom I really learned[7] the importance of making plans on a daily basis (although I still fail to do so sometimes). For interruptions from other people, it is impossible to isolate myself from the outside world, so I'd like to thank those who did not email me or ask me questions in the past few months

[4] https://yihui.name/en/2018/02/career-crisis/

[5] http://mikeyharper.uk

[6] https://www.education.pitt.edu/people/XuQin

[7] https://d.cosx.org/d/419325

and used public channels instead as I suggested[8]. I also thank those who did not get mad at me when my responses were extremely slow or even none. I appreciate all your understanding and patience. Besides, several users have started helping me answer GitHub and Stack Overflow questions related to R packages that I maintain, which is even better! These users include Marcel Schilling[9], Xianying Tan[10], Christophe Dervieux[11], and Garrick Aden-Buie[12], just to name a few. As someone who works from home, apparently I would not even have ten minutes of uninterrupted time if I do not send the little ones to daycare, so I want to thank all teachers at Small Miracle for freeing my daytime.

There have been a large number of contributors to the R Markdown ecosystem. More than 60 people[13] have contributed to the core package, **rmarkdown**. Several authors have created their own R Markdown extensions, as introduced in Part III of this book. Contributing ideas is no less helpful than contributing code. We have gotten numerous inspirations and ideas from the R community via various channels (GitHub issues, Stack Overflow questions, and private conversations, etc.). As a small example, Jared Lander, author of the book *R for Everyone*, does not meet me often, but every time he chats with me, I will get something valuable to work on. "How about writing books with R Markdown?" he asked me at the 2014 Strata conference in New York. Then we invented **bookdown** in 2016. "I really need fullscreen background images in ioslides. Look, Yihui, here are my ugly JavaScript hacks,[14]" he showed me on the shuttle to dinner at the 2017 RStudio Conference. A year later, background images were officially supported in ioslides presentations.

As I mentioned previously, R Markdown is standing on the shoulders of the giant, Pandoc. I'm always amazed by how fast John MacFarlane, the main author of Pandoc, responds to my GitHub issues. It is hard to imagine a person dealing with 5000 GitHub issues[15] over the years

[8] https://yihui.name/en/2017/08/so-gh-email/

[9] https://yihui.name/en/2018/01/thanks-marcel-schilling/

[10] https://shrektan.com

[11] https://github.com/cderv

[12] https://www.garrickadenbuie.com

[13] https://github.com/rstudio/rmarkdown/graphs/contributors

[14] https://www.jaredlander.com/2017/07/fullscreen-background-images-in-ioslides-presentations/

[15] https://github.com/jgm/pandoc

while maintaining the excellent open-source package and driving the Markdown standards forward. We should all be grateful to John and contributors of Pandoc.

As I was working on the draft of this book, I received a lot of helpful reviews from these reviewers: John Gillett (University of Wisconsin), Rose Hartman (UnderstandingData), Amelia McNamara (Smith College), Ariel Muldoon (Oregon State University), Yixuan Qiu (Purdue University), Benjamin Soltoff (University of Chicago), David Whitney (University of Washington), and Jon Katz (independent data analyst). Tareef Kawaf (RStudio) also volunteered to read the manuscript and provided many helpful comments. Aaron Simumba[16], Peter Baumgartner[17], and Daijiang Li[18] volunteered to carefully correct many of my typos. In particular, Aaron has been such a big helper with my writing (not limited to only this book) and sometimes I have to compete with him[19] in correcting my typos!

There are many colleagues at RStudio whom I want to thank for making it so convenient and even enjoyable to author R Markdown documents, especially the RStudio IDE team including J.J. Allaire, Kevin Ushey, Jonathan McPherson, and many others.

Personally I often feel motivated by members of the R community. My own willpower is weak, but I can gain a lot of power from this amazing community. Overall the community is very encouraging, and sometimes even fun, which makes me enjoy my job. For example, I do not think you can often use the picture of a professor for fun in your software, but the "desiccated baseR-er"[20] Karl Broman is an exception (see Section **??**), as he allowed me to use a mysteriously happy picture of him.

Lastly, I want to thank my editor, John Kimmel, for his continued help with my fourth book. I think I have said enough about him and his team at Chapman & Hall in my previous books. The publishing experience has always been so smooth. I just wonder if it would be possible someday that our meticulous copy-editor, Suzanne Lassandro,

[16] https://asimumba.rbind.io
[17] http://peter.baumgartner.name
[18] https://daijiang.name
[19] https://github.com/rbind/yihui/commit/d8f39f7aa
[20] https://twitter.com/kwbroman/status/922545181634768897

would fail to identify more than 30 issues for me to correct in my first draft. Probably not. Let's see.

Yihui Xie
Elkhorn, Nebraska

About the Authors

This book is primarily put together by me (Yihui Xie), making use of the existing R documentation of the **rmarkdown** package and the **rmarkdown** website, which were mainly contributed by J.J. Allaire and Garrett Grolemund.

Yihui Xie

Yihui Xie (https://yihui.name) is a software engineer at RStudio (https://www.rstudio.com). He earned his PhD from the Department of Statistics, Iowa State University. He is interested in interactive statistical graphics and statistical computing. As an active R user, he has authored several R packages, such as **knitr**, **bookdown**, **blogdown**, **xaringan**, **tinytex**, **animation**, **DT**, **tufte**, **formatR**, **fun**, **xfun**, **mime**, **highr**, **servr**, and **Rd2roxygen**, among which the **animation** package won the 2009 John M. Chambers Statistical Software Award (ASA). He also co-authored a few other R packages, including **shiny**, **rmarkdown**, and **leaflet**.

He has authored two books, *Dynamic Documents with knitr* (Xie, 2015), and *bookdown: Authoring Books and Technical Documents with R Markdown* (Xie, 2016), and co-authored two book, *blogdown: Creating Websites with R Markdown* (Xie et al., 2017), and *R Markdown: The Definitive Guide* (Xie et al., 2018).

In 2006, he founded the Capital of Statistics (https://cosx.org), which has grown into a large online community on statistics in China. He initiated the Chinese R conference in 2008, and has been involved in organizing R conferences in China since then. During his PhD training at Iowa State University, he won the Vince Sposito Statisti-

cal Computing Award (2011) and the Snedecor Award (2012) in the Department of Statistics.

He occasionally rants on Twitter (https://twitter.com/xieyihui), and most of the time you can find him on GitHub (https://github.com/yihui).

He enjoys spicy food as much as classical Chinese literature.

J.J. Allaire

J.J. Allaire is the founder of RStudio and the creator of the RStudio IDE. J.J. is an author of several packages in the R Markdown ecosystem including **rmarkdown**, **flexdashboard**, **learnr**, and **radix**.

Garrett Grolemund

Garrett Grolemund is the co-author of *R for Data Science* and author of *Hands-On Programming with R*. He wrote the **lubridate** R package and works for RStudio as an advocate who trains engineers to do data science with R and the Tidyverse. If you use R yourself, you may recognize Garrett from his video courses on Datacamp.com and O'Reilly media, or for his series of popular R cheatsheets distributed by RStudio.

Garrett earned his PhD in Statistics from Rice University in 2012 under the guidance of Hadley Wickham. Before that, he earned a Bachelor's degree in Psychology from Harvard University and briefly attended law school. Garrett has been one of the foremost promoters of Shiny, R Markdown, and the Tidyverse, documenting and explaining each in detail.

Part I

Get Started

1

Installation

We assume you have already installed R (https://www.r-project.org) (R Core Team, 2021) and the RStudio IDE (https://www.rstudio.com). RStudio is not required but recommended, because it makes it easier for an average user to work with R Markdown. If you do not have RStudio IDE installed, you will have to install Pandoc (http://pandoc.org), otherwise there is no need to install Pandoc separately because RStudio has bundled it. Next you can install the **rmarkdown** package in R:

```
# Install from CRAN
install.packages('rmarkdown')

# Or if you want to test the development version,
# install from GitHub
if (!requireNamespace("devtools"))
  install.packages('devtools')
devtools::install_github('rstudio/rmarkdown')
```

If you want to generate PDF output, you will need to install LaTeX. For R Markdown users who have not installed LaTeX before, we recommend that you install TinyTeX (https://yihui.name/tinytex/):

```
install.packages('tinytex')
tinytex::install_tinytex()  # install TinyTeX
```

TinyTeX is a lightweight, portable, cross-platform, and easy-to-maintain LaTeX distribution. The R companion package **tinytex** (Xie, 2021b) can help you automatically install missing LaTeX packages when compiling LaTeX or R Markdown documents to PDF, and also ensures a LaTeX document is compiled for the correct number of times to resolve all cross-references. If you do not understand what

these two things mean, you should probably follow our recommendation to install TinyTeX, because these details are often not worth your time or attention.

With the **rmarkdown** package, RStudio/Pandoc, and LaTeX, you should be able to compile most R Markdown documents. In some cases, you may need other software packages, and we will mention them when necessary.

Part II

Nozioni preliminari

2

Concetti chiave

La *data science* si pone all'intersezione tra statistica e informatica. La statistica è un insieme di metodi ugilizzati per estrarre informazioni dai dati; l'informatica implementa tali procedure in un software. In questo Capitolo vengono introdotti i concetti fondamentali.

2.1 Popolazioni e campioni

Popolazione. L'analisi dei dati inizia con l'individuazione delle unità portatrici di informazioni circa il fenomeno di interesse. Si dice popolazione (o universo) l'insieme Ω delle entità capaci di fornire informazioni sul fenomeno oggetto dell'indagine statistica. Possiamo scrivere $\Omega = \{\omega_i\}_{i=1,\dots,n} = \{\omega_1, \omega_2, \dots, \omega_n\}$, oppure $\Omega = \{\omega_1, \omega_2, \dots\}$ nel caso di popolazioni finite o infinite, rispettivamente.

L'obiettivo principale della ricerca psicologica è conoscere gli esiti psicologici e i loro fattori trainanti nella popolazione. Questo è l'obiettivo delle sperimentazioni psicologiche e della maggior parte degli studi osservazionali in psicologia. È quindi necessario essere molto chiari sulla popolazione a cui si applicano i risultati della ricerca. La popolazione può essere ben definita, ad esempio, tutte le persone che si trovavano nella città di Hiroshima al momento dei bombardamenti atomici e sono sopravvissute al primo anno, o può essere ipotetica, ad esempio, tutte le persone depresse che hanno subito o saranno sottoporsi ad un intervento di psicoterapia. Il ricercatore deve sempre essere in grado di determinare se un soggetto appartiene alla popolazione oggetto di interesse.

Una *sottopopolazione* è una popolazione in sé e per sé che soddisfa proprietà ben definite. Negli esempi precedenti, potremmo essere interessati alla sottopopolazione di uomini di età inferiore ai 20 anni

o di pazienti depressi sottoposti ad uno specifico intervento psicologico. Molte questioni scientifiche riguardano le differenze tra sottopopolazioni; ad esempio, confrontando i gruppi con o senza psicoterapia per determinare se il trattamento è vantaggioso. I modelli di regressione, introdotti nel Capitolo **??** riguardano le sottopopolazioni, in quanto stimano il risultato medio per diversi gruppi (sottopopolazioni) definiti dalle covariate.

Campione. Gli elementi ω_i dell'insieme Ω sono detti *unità statistiche*. Un sottoinsieme della popolazione, ovvero un insieme di elementi ω_i, viene chiamato *campione*. Ciascuna unità statistica ω_i (abbreviata con u.s.) è portatrice dell'informazione che verrà rilevata mediante un'operazione di misurazione.

Un campione è dunque un sottoinsieme della popolazione utilizzato per conoscere tale popolazione. A differenza di una sottopopolazione definita in base a chiari criteri, un campione viene generalmente selezionato tramite un procedura casuale. Il *campionamento casuale* consente allo scienziato di trarre conclusioni sulla popolazione e, soprattutto, di quantificare l'incertezza sui risultati. I campioni di un sondaggio sono esempi di campioni casuali, ma molti studi osservazionali non sono campionati casualmente. Possono essere *campioni di convenienza*, come coorti di studenti in un unico istituto, che consistono di tutti gli studenti sottoposti ad un certo intervento psicologico in quell'istituto. Indipendentemente da come vengono ottenuti i campioni, il loro uso al fine di conoscere una popolazione target significa che i problemi di rappresentatività sono inevitabili e devono essere affrontati.

2.2 Variabili e costanti

Definiamo *variabile statistica* la proprietà (o grandezza) che è oggetto di studio nell'analisi dei dati. Una variabile è una proprietà di un fenomeno che può essere espressa in più valori sia numerici sia categoriali. Il termine "variabile" si contrappone al termine "costante" che descrive una proprietà invariante di tutte le unità statistiche.

Si dice *modalità* ciascuna delle varianti con cui una variabile statistica può presentarsi. Definiamo *insieme delle modalità* di una variabile sta-

tistica l'insieme M di tutte le possibili espressioni con cui la variabile può manifestarsi. Le modalità osservate e facenti parte del campione si chiamano *dati* (si veda la Tabella 1.1).

Example 2.1. Supponiamo che il fenomeno studiato sia l'intelligenza. In uno studio, la popolazione potrebbe corrispondere all'insieme di tutti gli italiani adulti. La variabile considerata potrebbe essere il punteggio del test standardizzato WAIS-IV. Le modalità di tale variabile potrebbero essere $112, 92, 121, ...$ Tale variabile è di tipo quantitativo discreto.

Example 2.2. Supponiamo che il fenomeno studiato sia il compito Stroop. La popolazione potrebbe corrispondere all'insieme dei bambini dai 6 agli 8 anni. La variabile considerata potrebbe essere il reciproco dei tempi di reazione in secondi. Le modalità di tale variabile potrebbero essere $1/2.35, 1/1.49, 1/2.93, ...$ La variabile è di tipo quantitativo continuo.

Example 2.3. Supponiamo che il fenomeno studiato sia il disturbo di personalità. La popolazione potrebbe corrispondere all'insieme dei detenuti nelle carceri italiane. La variabile considerata potrebbe essere l'assessment del disturbo di personalità tramite interviste cliniche strutturate. Le modalità di tale variabile potrebbero essere i Cluster A, Cluster B, Cluster C descritti dal DSM-V. Tale variabile è di tipo qualitativo.

2.2.1 Variabili casuali

Il termine *variabile* usato nella statistica è equivalente al termine *variabile casuale* usato nella teoria delle probabilità. Lo studio dei risultati degli interventi psicologici è lo studio delle variabili casuali che misurano questi risultati. Una variabile casuale cattura una caratteristica specifica degli individui nella popolazione e i suoi valori variano tipicamente tra gli individui. Ogni variabile casuale può assumere in teoria una gamma di valori sebbene, in pratica, osserviamo un valore specifico per ogni individuo. Quando faremo riferiremo alle variabili casuali considerate in termini generali useremo lettere maiuscole come X e Y; quando faremo riferimento ai valori che una variabile casuale assume in determinate circostanze useremo lettere minuscole come x e y.

2.2.2 Variabili indipendenti e variabili dipendenti

Un primo compito fondamentale in qualsiasi analisi dei dati è l'identificazione delle variabili dipendenti (Y) e delle variabili indipendenti (X). Le variabili dipendenti sono anche chiamate variabili di esito o di risposta e le variabili indipendenti sono anche chiamate predittori o covariate. Ad esempio, nell'analisi di regressione, che esamineremo in seguito, la domanda centrale è quella di capire come Y cambia al variare di X. Più precisamente, la domanda che viene posta è: se il valore della variabile indipendente X cambia, qual è la conseguenza per la variabile dipendente Y? In parole povere, le variabili indipendenti e dipendenti sono analoghe a "cause" ed "effetti", laddove le virgolette usate qui sottolineano che questa è solo un'analogia e che la determinazione delle cause può avvenire soltanto mediante l'utilizzo di un appropriato disegno sperimentale e di un'adeguata analisi statistica.

Se una variabile è una variabile indipendente o dipendente dipende dalla domanda di ricerca. A volte può essere difficile decidere quale variabile è dipendente e quale è indipendente, in particolare quando siamo specificamente interessati ai rapporti di causa/effetto. Ad esempio, supponiamo di indagare l'associazione tra esercizio fisico e insonnia. Vi sono evidenze che l'esercizio fisico (fatto al momento giusto della giornata) può ridurre l'insonnia. Ma l'insonnia può anche ridurre la capacità di una persona di fare esercizio fisico. In questo caso, dunque, non è facile capire quale sia la causa e quale l'effetto, quale sia la variabile dipendente e quale la variabile indipendente. La possibilità di identificare il ruolo delle variabili (dipendente/indipendente) dipende dalla nostra comprensione del fenomeno in esame.

Example 2.4. Uno psicologo convoca 120 studenti universitari per un test di memoria. Prima di iniziare l'esperimento, a metà dei soggetti viene detto che si tratta di un compito particolarmente difficile; agli altri soggetti non viene data alcuna indicazione. Lo psicologo misura il punteggio nella prova di memoria di ciascun soggetto.

In questo esperimento, la variabile indipendente è l'informazione sulla difficoltà della prova. La variabile indipendente viene manipolata dallo sperimentatore assegnando i soggetti (di solito in maniera causale) o alla condizione (modalità) "informazione assegnata" o "in-

formazione non data". La variabile dipendente è ciò che viene misurato nell'esperimento, ovvero il punteggio nella prova di memoria di ciascun soggetto.

2.2.3 La matrice dei dati

Le realizzazioni delle variabili esaminate in una rilevazione statistica vengono organizzate in una *matrice dei dati*. Le colonne della matrice dei dati contengono gli insiemi dei dati individuali di ciascuna variabile statistica considerata. Ogni riga della matrice contiene tutte le informazioni relative alla stessa unità statistica. Una generica matrice dei dati ha l'aspetto seguente:

$$D_{m,n} = \begin{pmatrix} \omega_1 & a_1 & b_1 & \cdots & x_1 & y_1 \\ \omega_2 & a_2 & b_2 & \cdots & x_2 & y_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \omega_n & a_n & b_n & \cdots & x_n & y_n \end{pmatrix}$$

dove, nel caso presente, la prima colonna contiene il nome delle unità statistiche, la seconda e la terza colonna si riferiscono a due mutabili statistiche (variabili categoriali; A e B) e ne presentano le modalità osservate nel campione mentre le ultime due colonne si riferiscono a due variabili statistiche (X e Y) e ne presentano le modalità osservate nel campione. Generalmente, tra le unità statistiche ω_i non esiste un ordine progressivo; l'indice attribuito alle unità statistiche nella matrice dei dati si riferisce semplicemente alla riga che esse occupano.

2.3 Parametri e modelli

Ogni variabile casuale ha una *distribuzione* che descrive la probabilità che la variabile assuma qualsiasi valore in un dato intervallo.[1] Senza ulteriori specificazioni, una distribuzione può fare riferimento a un'intera famiglia di distribuzioni. I parametri, tipicamente indicati con lettere greche come μ e α, ci permettono di specificare di quale

[1] In questo e nei successivi Paragrafi di questo Capitolo introduco gli obiettivi della *data science* utilizzando una serie di concetti che saranno chiariti solo in seguito. Questa breve panoramica risulterà dunque solo in parte comprensibile ad una prima lettura e serve solo per definire la *big picture* dei temi trattati in questo insegnamento. Il significato dei termini qui utilizzati sarà chiarito nei Capitoli successivi.

membro della famiglia stiamo parlando. Quindi, si può parlare di una variabile casuale con una distribuzione Normale, ma se viene specificata la media μ = 100 e la varianza σ^2 = 15, viene individuata una specifica distribuzione Normale – nell'esempio, la distribuzione del quoziente di intelligenza.

I metodi statistici parametrici specificano la famiglia delle distribuzioni e quindi utilizzano i dati per individuare, stimando i parametri, una specifica distribuzione all'interno della famiglia di distribuzioni ipotizzata. Se f è la PDF di una variabile casuale Y, l'interesse può concentrarsi sulla sua media e varianza. Nell'analisi di regressione, ad esempio, cerchiamo di spiegare come i parametri di f dipendano dalle covariate X. Nella regressione lineare classica, assumiamo che Y abbia una distribuzione normale con media $\mu = E(Y)$, e stimiamo come $E(Y)$ dipenda da X. Poiché molti esiti psicologici non seguono una distribuzione normale, verranno introdotte distribuzioni più appropriate per questi risultati. I metodi non parametrici, invece, non specificano una famiglia di distribuzioni per f. In queste dispense faremo riferimento a metodi non parametrici quando discuteremo della statistica descrittiva.

Il termine *modello* è onnipresente in statistica e nella *data science*. Il modello statistico include le ipotesi e le specifiche matematiche relative alla distribuzione della variabile casuale di interesse. Il modello dipende dai dati e dalla domanda di ricerca, ma raramente è unico; nella maggior parte dei casi, esiste più di un modello che potrebbe ragionevolmente usato per affrontare la stessa domanda di ricerca e avendo a disposizione i dati osservati. Nella previsione delle aspettative future dei pazienti depressi che discuteremo in seguito (**?**), ad esempio, la specifica del modello include l'insieme delle covariate candidate, l'espressione matematica che collega i predittori con le aspettative future e qualsiasi ipotesi sulla distribuzione della variabile dipendente. La domanda di cosa costituisca un buon modello è una domanda su cui torneremo ripetutamente in questo insegnamento.

2.4 Effetto

L'*effetto* è una qualche misura dei dati. Dipende dal tipo di dati e dal tipo di test statistico che si vuole utilizzare. Ad esempio, se viene lanciata una moneta 100 volte e esce testa 66 volte, l'effetto sarà 66/100. Diventa poi possibile confrontare l'effetto ottenuto con l'effetto nullo che ci si aspetterebbe da una moneta bilanciata (50/100), o con qualsiasi altro effetto che può essere scelto. La *dimensione dell'effetto* si riferisce alla differenza tra l'effetto misurato nei dati e l'effetto nullo (di solito un valore che ci si aspetta di ottenere in base al caso soltanto).

2.5 Stima e inferenza

La stima è il processo mediante il quale il campione viene utilizzato per conoscere le proprietà di interesse della popolazione. La media campionaria è una stima naturale della media della popolazione e la mediana campionaria è una stima naturale della mediana della popolazione. Quando parliamo di stimare una proprietà della popolazione (a volte indicata come parametro della popolazione) o di stimare la distribuzione di una variabile casuale, stiamo parlando dell'utilizzo dei dati osservati per conoscere le proprietà di interesse della popolazione. L'inferenza statistica è il processo mediante il quale le stime campionarie vengono utilizzate per rispondere a domande di ricerca e per valutare specifiche ipotesi relative alla popolazione. Discuteremo le procedure bayesiane dell'inferenza nell'ultima parte di queste dispense.

2.6 Metodi e procedure della psicologia

Un modello psicologico di un qualche aspetto del comportamento umano o della mente ha le seguenti proprietà:

1. descrive le caratteristiche del comportamento in questione,
2. formula predizioni sulle caratteristiche future del comportamento,
3. è sostenuto da evidenze empiriche,
4. deve essere falsificabile (ovvero, in linea di principio, deve potere fare delle predizioni su aspetti del fenomeno considerato che non sono ancora noti e che, se venissero indagati, potrebbero portare a rigettare il modello, se si dimostrassero incompatibili con esso).

L'analisi dei dati valuta un modello psicologico utilizzando strumenti statistici.

Questa dispensa è strutturata in maniera tale da rispecchiare la suddivisione tra i temi della misurazione, dell'analisi descrittiva e dell'inferenza. Nel prossimo Capitolo sarà affrontato il tema della misurazione e, nell'ultima parte della dispensa verrà discusso l'argomento più difficile, quello dell'inferenza. Prima di affrontare il secondo tema, l'analisi descrittiva dei dati, sarà necessario introdurre il linguaggio di programmazione statistica R (un'introduzione a R è fornita in Appendice). Inoltre, prima di potere discutere l'inferenza, dovranno essere introdotti i concetti di base della teoria delle probabilità, in quanto l'inferenza non è che l'applicazione della teoria delle probabilità all'analisi dei dati.

Bibliography

Allaire, J., Xie, Y., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wickham, H., Cheng, J., Chang, W., and Iannone, R. (2021). *rmarkdown: Dynamic Documents for R*. R package version 2.11.

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Xie, Y. (2015). *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition. ISBN 978-1498716963.

Xie, Y. (2016). *bookdown: Authoring Books and Technical Documents with R Markdown*. Chapman and Hall/CRC, Boca Raton, Florida. ISBN 978-1138700109.

Xie, Y. (2021a). *knitr: A General-Purpose Package for Dynamic Report Generation in R*. R package version 1.37.

Xie, Y. (2021b). *tinytex: Helper Functions to Install and Maintain TeX Live, and Compile LaTeX Documents*. R package version 0.36.

Xie, Y., Allaire, J., and Grolemund, G. (2018). *R Markdown: The Definitive Guide*. Chapman and Hall/CRC, Boca Raton, Florida. ISBN 9781138359338.

Xie, Y., Hill, A. P., and Thomas, A. (2017). *blogdown: Creating Websites with R Markdown*. Chapman and Hall/CRC, Boca Raton, Florida. ISBN 978-0815363729.

Index