

A Minimal Book Example

Yihui Xie

2023-03-29

Contents

1	Prerequisites	5
2	Introduction	7
3	Literature	9
4	Methods	11
4.1	math example	11
4.2	A non math example over here	11
5	Applications	15
5.1	Example one	15
5.2	Example two	15
6	Final Words	17

Chapter 1

Prerequisites

This is a *sample* book written in **Markdown**. You can use anything that Pandoc's Markdown supports, e.g., a math equation $a^2 + b^2 = c^2$.

The **bookdown** package can be installed from CRAN or Github:

```
install.packages("bookdown")  
# or the development version  
# devtools::install_github("rstudio/bookdown")
```

Remember each Rmd file contains one and only one chapter, and a chapter is defined by the first-level heading #.

To compile this example to PDF, you need XeLaTeX. You are recommended to install TinyTeX (which includes XeLaTeX): <https://yihui.name/tinytex/>.

Chapter 2

Introduction

You can label chapter and section titles using `{#label}` after them, e.g., we can reference Chapter 2. If you do not manually label them, there will be automatic labels anyway, e.g., Chapter 4.

Figures and tables with captions will be placed in `figure` and `table` environments, respectively.

```
par(mar = c(4, 4, .1, .1))  
plot(pressure, type = 'b', pch = 19)
```

Reference a figure by its code chunk label with the `fig:` prefix, e.g., see Figure 2.1. Similarly, you can reference tables generated from `knitr::kable()`, e.g., see Table 2.1.

```
knitr::kable(  
  head(iris, 20), caption = 'Here is a nice table!',  
  booktabs = TRUE  
)
```

You can write citations, too. For example, we are using the **bookdown** package (Xie, 2023) in this sample book, which was built on top of R Markdown and **knitr** (Xie, 2015).



Figure 2.1: Here is a nice figure!

Table 2.1: Here is a nice table!

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3.0	1.4	0.1	setosa
4.3	3.0	1.1	0.1	setosa
5.8	4.0	1.2	0.2	setosa
5.7	4.4	1.5	0.4	setosa
5.4	3.9	1.3	0.4	setosa
5.1	3.5	1.4	0.3	setosa
5.7	3.8	1.7	0.3	setosa
5.1	3.8	1.5	0.3	setosa

Chapter 3

Literature

Here is a review of existing methods.

References

Chapter 4

Methods

We describe our methods in this chapter.

Math can be added in body using usual syntax like this

4.1 math example

p is unknown but expected to be around $1/3$. Standard error will be approximated

$$SE = \sqrt{\left(\frac{p(1-p)}{n}\right)} \approx \sqrt{\frac{1/3(1-1/3)}{300}} = 0.027$$

You can also use math in footnotes like this¹.

We will approximate standard error to 0.027^2

4.2 A non math example over here

If I wanted to add a new example to this page I could simply add it here

¹where we mention $p = \frac{a}{b}$

² p is unknown but expected to be around $1/3$. Standard error will be approximated

$$SE = \sqrt{\left(\frac{p(1-p)}{n}\right)} \approx \sqrt{\frac{1/3(1-1/3)}{300}} = 0.027$$

```
index <- 1:10
```

```
index
```

```
## [1] 1 2 3 4 5 6 7 8 9 10
```

f all you need is the correct name of a function (eg. was it `list_files()`, `list.files()`, or `listFiles()`?), just type the first three characters of the function name and use RStudio’s autocomplete mechanism to find what you need.

If you are wondering about the parameters of a function, just hover the cursor over the function name in the script pane or the command line in the console and wait a moment: a little window with the function signature will pop up.

But for more detailed information, help is available for all commands as well as for the R syntax. As well, help is available to find the names of commands when you are not sure of them. You can get help through the command line, or from a search field in the Help tab of the lower-right pane.

(`help()` is a function in R, arguments to a function are passed in parentheses “()”)

```
help(rnorm)
```

(shorthand for the same thing)

```
?rnorm
```

(what was the name of that again ... ?)

```
?binom
```

```
## No documentation for 'binom' in specified packages and libraries:
## you could try '??binom'
```

```
??binom
```

(I see “Binomial” in the list of keywords...)

```
?Binomial
```

(Alternatively: use the `apropos()` function.

```
?apropos
apropos("med")  # all functions that contain the string "med"

## [1] "elNamed"      "elNamed<-"    "median"       "median.default"
## [5] "medpolish"    "runmed"

apropos("^med") # all functions that begin with the string

## [1] "median"      "median.default" "medpolish"

apropos("med$") # all functions that end with the string

## [1] "elNamed" "runmed"
```

If you need help on operators, place them in quotation marks. Try:

```
? "+"
"? ~"
"? ["
"? %in%"
```

That's all fine, but you will soon notice that **R**'s help documentation is very technical and precise but not all that helpful for newcomers (who need the most help). To illustrate, open the help window for the function `var()`.

```
?var
```

Here's what you might look for:

- The **Description** section describes the function in general technical terms.
- The **Usage** section tells you what arguments are required (these don't have defaults), what arguments have defaults, and what the defaults are, and whether additional arguments ("...") are allowed. Often a function comes in several variants, you will find them here.
- The **Arguments** section provides detailed information . You should read it, especially regarding whether the arguments are single values, vectors, or other objects, and what effect missing arguments will have.
- The **Details** section might provide common usage and context information. It might also not. Often functions have crucial information buried in an innocuous note here.

- You really have to read and understand the **Value** section. It explains the output of the function (its returnvalue). The object a function returns could be a list, a matrix or something else (we'll discuss these data types in detail elsewhere.). The value could also be an object that has special methods defined e.g. for plotting it. In that case, the object is formally a “list”, and its named “components” can be retrieved with the usual list syntax.

If you look at the bottom of the help text, you will usually find examples of the function's usage; these sometimes make matters more clear than the terse and principled help-text above.

What you often won't find:

- Clear commented, examples that relate to the most frequent use cases.
- Explanations why a particular function is done in a particular way (e.g. why the denominator is $n-1$ for `sd()` and `var()`).

Notes on common errors:

- A (reasonably) exhaustive list of alternatives and related functions. There are usually some entries, but there is no guarantee that all alternatives are listed – especially if they are provided by an external package.

Therefore, my first approach for **R** information is usually to Google for what interests me and this is often the quickest way to find working example code. **R** has a very large user base and it is becoming very rare that a reasonable question will not have a reasonable answer among the top three hits of a Google search. Also, as a result of a Google search, it may turn out that something can't be done (easily) – and you won't find things that can't be done in the help system at all. You may want to include “r language” in your search terms, although Google is usually pretty good at figuring out what kind of “r” you are looking for, especially if your query includes a few terms vaguely related to statistics or programming.

Chapter 5

Applications

Some *significant* applications are demonstrated in this chapter.

5.1 Example one

5.2 Example two

Chapter 6

Final Words

We have finished a nice book.

Bibliography

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

Xie, Y. (2023). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.33.