A Minimal Book Example

John Doe

2023-09-06

Contents

Chapter 1

About

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

1.1 Usage

Each **bookdown** chapter is an .Rmd file, and each .Rmd file can contain one (and only one) chapter. A chapter *must* start with a first-level heading: `# A good chapter`, and can contain one (and only one) first-level heading.

Use second-level and higher headings within chapters like: `## A short section` or `### An even shorter section`.

The `index.Rmd` file is required, and is also your first book chapter. It will be the homepage when you render the book.

1.2 Render book

You can render the HTML version of this example book without changing anything:

1. Find the **Build** pane in the RStudio IDE, and
2. Click on **Build Book**, then select your output format, or select "All formats" if you'd like to use multiple formats from the same book source files.

Or build the book from the R console:

```
bookdown::render_book()
```

To render this example to PDF as a `bookdown::pdf_book`, you'll need to install XeLaTeX. You are recommended to install TinyTeX (which includes XeLaTeX): https://yihui.org/tinytex/.

1.3 Preview book

As you work, you may start a local server to live preview this HTML book. This preview will update as you edit the book when you save individual .Rmd files. You can start the server in a work session by using the RStudio add-in "Preview book", or from the R console:

```
bookdown::serve_book()
```

Chapter 2

Learning R

This is notes about learning R.

Please build this simple boook!!!!! ahhhhhhhhhhhha

2.1 Environment configuration

- install a package: install.packages("tidyverse") Once a package installed you don't need to install it again
- load a library Once the package is installed, you need to "load" it in your current environment libary(tidyverse) It was a think that was disturbing me when starting R/ install.packages takes a double quote surrounding the package name
  whereas library doesn't take double quote around the package name
- options(digits =3) # 3 significant digits
- installing two or more packages at once with c():
  install.packages(c("tidytext","gutenbergr"))
- question mark + name of the function to get the help of the function
  It opens in the bottom right panel in most configurations of R Studio.
  example : ?gutenberg_metadata()

2.2 Basic manipulations

- create a vector with rep (repeat)

my_vec = rep(c(1,2), times = 7) or

my_vec2 = rep(c(1,2), times=c(3,8))

2.3 How to explore a data frame?

- df %>% head()
- df %>% tail()

In both cases, you can specify the number of rows at the top or at the bottom you want to see.
For example head(3) or tail(3)

2.4 About stringr package

Stringr is more coherent than base R functions for strings treatments.
Stringr functions always begin with prefix **str_** ; the first argument is always the string you want to treat. And then comes the pattern you want to identify.

Most common and useful functions in Stringr :

- str_detect() -> returns a logical vector (a vector of TRUE and FALSE)
- str_subset()
- str_view()
- str_view_all()
- str_replace()
- str_replace_all()
- str_split()
- str_trim()
- str_to_lower()

2.5 About Regex in R

2.5.1 Special characters

- \\d stands for **one of any digit 0,1,2, up to 9**
- \\s stands for **one** charater whitespace
- The dot "." **matches any character**
- So, to match a literal dot "." in regex, we need two backslashes then dot \\.
- The star "*" stands for **0 or more** instances of the previous character

- The plus sign "+" stands for **1 or more** instances of the previous character
- The question mark "?" stands for **0 or one** instance of the previous character
- () () "\\1" capture le groupe de la parenthèse 1 et "\\2" capture le groupe de la parenthèse 2

Separate and extract function are from tidyr package.
In **extract**, you can use regex to split a string.

```
library(dplyr)
library(tidyr)
s <- c("5'6", "6'4")
tab <- data.frame(x = s)

tab %>% separate(x,c("feet","inches"),sep="'")
```

```
##   feet inches
## 1    5      6
## 2    6      4
```

```
tab %>% extract(x,c("feet","inches"), regex = "(\\d)'(\\d{1,2})")
```

```
##   feet inches
## 1    5      6
## 2    6      4
```

2.6 Creating samples

set.seed(1) sample()

2.7 About dates

Sys.time() from base R returns current date/time.

2.8 Useful packages or datasets

- gapminder library(gapminder)
  data("gapminder")

2.9 Useful libraries

- It maybe possible to extract a table from a pdf with pdftools
  Not tested myself library("pdftools") temp_file <- tempfile() url <- "https://www.pnas.org/action/downloadSupplement?doi=10.1073%2Fpnas.1510159112&file=pnas.201510159SI.pdf" download.file(url, temp_file) txt <- pdf_text(temp_file) file.remove(temp_file)

Chapter 3

Learning Python

- shebang line #!/usr/bin/env python3

Chapter 4

Learning Git/Github

- git config user.name “my_name”
- git config user.email “me@example.com”
- git config – global user.name “my_name”
  -> set the value of the username for all git repos
  whereas if “git config” without global you set it up for the current directory
- git init -> when in the directory which you want to set under git control (initialize a new repo)
- git add myfile -> stagge myfile (place it in the stagging area)
- git commit -m “my message for this commit”
- git config -l
- git status -> check current state
- three status for tracked files : modified/stagged/commited
- in order to vizualize all **the commits** (not all the modifications) which were made :
  git log
- git add -p a way to review changes before adding them git will show us which files were not stagged and ask us if we want to commit
- git log -p gives more informations in a viewer
  the -p comes from patch you can see differences line by line you can quit the viewer typing q as with less viewer
- git log –stat extra info (how many lines you have added or remove)

- git show ‘commit_id’
  git show takes a commit id as a parameter
- git -stats
- Admit you modified a file readme.txt which is under version control.
  You can see the modifications since the previous version with this command line:
  git diff readme.txt
- Add a file to .gitignore in order it is not tracked anymore(?) echo 01-Learning-R.Rmd > .gitignore echo .RData » .gitignore after modifying .gitignore you need to stagge (git add) and commit (git commit) it.
- git commit -a -m ‘message for the commit’ when you want to commit only the modifications (a is for only modified files / m is for message)
- git rm filename
  after this you must commit in order the changes to be taken into account
- git mv filename in order to move or rename a file
  git mv old_name new_name after this need to commit
- git diff -u
- git diff only shows unstagged changes by default
- git diff –stagged to see the changes that are staged but not commited
- git
- git checkout “commit_id” roll back to a previous version
- git reset remove from staging area
- git commit –amend to modify a commit

Chapter 5

Footnotes and citations

5.1 Footnotes

Footnotes are put inside the square brackets after a caret `^[]`. Like this one [1].

5.2 Citations

Reference items in your bibliography file(s) using `@key`.

For example, we are using the **bookdown** package [Xie, 2023] (check out the last code chunk in index.Rmd to see how this citation key was added) in this sample book, which was built on top of R Markdown and **knitr** [Xie, 2015] (this citation was added manually in an external file book.bib). Note that the `.bib` files need to be listed in the index.Rmd with the YAML `bibliography` key.

The RStudio Visual Markdown Editor can also make it easier to insert citations: https://rstudio.github.io/visual-markdown-editing/#/citations

[1]This is a footnote.

Chapter 6

Learning Markdown

In markdown you need to escape twice the backslash in order to display two backslashes
So what you see here, I wrote it with **four** not just three : \\

To write a list, you must write a star * followed by a coma a the beginning of a line. Before the list starts you need a blankline and same at the end of the list otherwise Mardown won't recognize it.

To introduce a return to the ligne, you need not only to type return in Markdown, but also to make the line followed by two spaces.

To make a few words bold you need to surrender it with two stars both sides.
It is **bold** gives : It is **bold**

Chapter 7

Learning linux commands

- git –version
- mkdir -> create a directory
- cat to read a file
- or less (type q in order to exit less viewer) why less? because previous version of less was more :)
- write in a file :
  echo toto et tata > toto.txt

echo toto et titi > titi.txt

- differences between two files:
  diff toto.txt titi.txt
  or diff -u toto.txt titi.txt
- diff -u is more readable than simple diff command.
- Create a diff file:
  diff -u toto.txt titi.txt > change.diff
- Patch the .diff file:
  patch titi.txt < change.diff
- Clear the console:
  just as in Rstudio ctrl+l or typing "clear" and then enter in the console. both works
- Content of a directory:
  dir or ls : both works.

- Content of a directory including hidden files:
  dir -a ls -a from the help of ls : " -a, –all do not ignore entries starting with."
- Add the options l to see rights on the files: ls -la
- Get the help in git bash on windows:
  function –help example: ls –help
- Make a file executable:
  chmod +x filename
- Open a file with nano :
  nano my_file.txt
- Save changes made to a file in nano:
  ctrl+o + Enter + ctrl+x
- 'cd -' in order to come back to previous directory

Chapter 8

Statistics with R

```r
beads <- rep(c("red","blue"), times = c(2,3))
beads
```

```
## [1] "red"  "red"  "blue" "blue" "blue"
```

```r
# pick a bead at random
sample(beads,1)
```

```
## [1] "blue"
```

```r
# evaluate the probability of drawing a blue at random
mean(beads == "blue")
```

```
## [1] 0.6
```

```r
B <- 10000

# Nota : if you want the "same random" each time
# you draw the sample, you need to set a seed
set.seed(1)
# for example



events <- replicate(B,sample(beads,1))
tab <- table(events)
# calculate probability of each events
prop.table(tab)
```

```
## events
##   blue    red
## 0.6028 0.3972
```

```
# note that with this monte carlo simulation we have
# quite the same probability for blue events as with
# mean(beads =="blue")
```

by default, sample samples without replacement. it means, it you try

```
sample(beads,5)
```

```
## [1] "blue" "red"  "red"  "blue" "blue"
```

```
# it works, but if you try
# sample(beads,6)
# it fails because there are only 6 beads
```

so sample by default is equivalent to sample(beads,5, replace = FALSE) but we can use sample **with** replacement.

```
# so this is working:
sample(beads,6,replace = TRUE)
```

```
## [1] "blue" "red"  "red"  "red"  "blue" "red"
```

```
# and we can use simply sample with replacement
# juste like with replicate

events2 <- sample(beads,B, replace = TRUE)
tab2 <- table(events2)
prop.table(tab2)
```

```
## events2
##   blue    red
## 0.6026 0.3974
```

```
# the result is almost the same
```

discrete and continuous variables example: discrete : flip of a coin outcome from the roll of a die web site traffic on a given day > particular discrete variable (no upper bound > poisson) same for the number of people clicking on an add

continuous : BMI (body mass index) of a subject four years after a baseline measurement hypertension status but could be modelled as discrete (1 hypertension/0 no hypertension)

- probability mass function (pmf) for discrete variable > assign a probability for each value they can take 1-must always be larger or equal to 0 (prob is number bet 0 abd 1) 2- the sum of the poss values has to add up to 1
- examples of pmf : binomial, canonical (flipping a cooin)

Bernouli = flip a coin X = 0 tails X = 1 represents heads

Two broad flavors of inference : * frequency, which uses "long run proportion of times an event occurs in **independent**, identically distributed repetitions" * he second is Bayesian in which the probability estimate for a hypothesis is updated as additional evidence is acquired

- If A and B are two **independent** events then the probability of them both occurring is the product of the probabilities. P(A&B) = P(A) * P(B)
- Suppose you rolled the fair die twice. What is the probability of rolling the same number two times in a row? Since we don't care what the outcome of the first roll is, its probability is 1. The second roll of the dice has to match the outcome of the first, so that has a probability of 1/6. The probability of both events occurring is 1 * 1/6.
- The probability of at least one of two events, A and B, occurring is the sum of their individual probabilities minus the probability of their intersection. P(A U B) = P(A) + P(B) - P(A&B).

Chapter 9

Learning Shiny

library(shiny)

- To have a look on which html tags are available in Shiny: ?builder
- Ctrl + U to see the html content of a web page > it opens the html code in a new page
- Alt + s + w + s or via the menu Session > setwdir > To source file loc

runApp()

shinyUI()

shinyServer()

Chapter 10

Learning html

Install visual studio and live server?

```
#<DOCTYPE! html>
#<html lang="en">


#<\html>
```

Bibliography

Yihui Xie. *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition, 2015. URL http://yihui.org/knitr/. ISBN 978-1498716963.

Yihui Xie. *bookdown: Authoring Books and Technical Documents with R Markdown*, 2023. URL https://CRAN.R-project.org/package=bookdown. R package version 0.35.