A Minimal Book Example

Yihui Xie

2021-10-23

Contents

Chapter 1

Prerequisites

Placeholder

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)
```

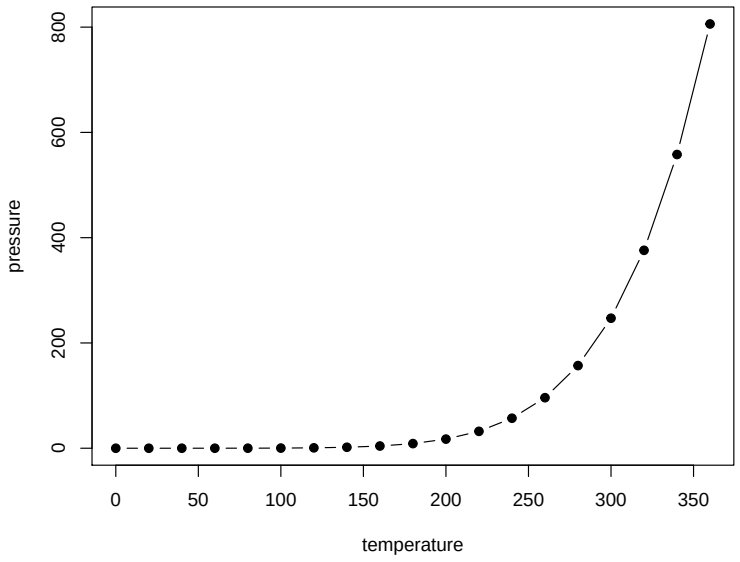

Figure 2.1: Here is a nice figure!

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

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, 2020) in this sample book, which was built on top of R Markdown and **knitr** (Xie, 2015).

Chapter 3

Literature

Here is a review of existing methods.

Chapter 4

Methods

We describe our methods in this chapter.

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.

Chapter 7

Definiciones

Placeholder

7.1 Parametro

7.2 Argumento

7.3 Opcion

7.4 Flags

7.4.1 Concatenacion

7.5 Definicion

Chapter 8

Casos especiales

Placeholder

8.1 help

8.2 Acceso a las variables de entorno

8.3 Valores booleanos

Chapter 9

Ejemplos de codigos

```
int main(int argc, char *argv[]) {
    char* argv2[10];
    std::cout << "Begin" << std::endl;
    argv2[1] = (char*) "default flags";
    cmdline::CmdLine cmdLine(flags);
    /*
    HashMap<std::string> *hmap = new HashMap<std::string>();
    cmdline::ParmItem item1 = cmdline::ParmItem("pepe2");

    hmap->insert(item1.name, std::make_unique<ParmItem>(item1));
    */
    checkDefaultFlags(flags, 2, argv2);

    argv2[1] = (char*)"+o";
    if (testFlagInvalid(flags, 2, (char**)argv2)) return 1;
/*
    argv2[1] = (char *)  "+v";
    if (testFlagDetect(flags, 2, argv2)) return 1;

    argv2[1] = (char*)"+ver";
    if (testFlagDetect(flags, 2, argv2)) return 1;

    argv2[1] = (char*)"+verbo";
    if (testFlagDetect(flags, 2, argv2)) return 1;

    argv2[1] = (char*)"+verbose";
    if (testFlagDetect(flags, 2, (char**)argv2)) return 1;
```

```
    argv2[1] = (char*) "-h";
    if (case01(2, (char**)argv2)) return 1;
    argv2[1] = (char*) "--help";
    if (case02(2, (char**)argv2)) return 1;
    */
    return 0;
}
```

Chapter 10

Verbos

Por definicion los parametros son inmutables, lo cual no implica que se puedan leer.

- getXXX: Obtiene una referencia al parametro
- letXXX: Obtieneuna copia del parametro
- hasXXX: Indica siexiste o no
- addXXX: Permite incluir parametros

Bibliography

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

Xie, Y. (2020). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.21.