# A Minimal Book Example

Yihui Xie

2020-03-01

# Contents

# 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:

```r
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

```
{r eval=FALSE} #
install.packages('tinytex')
#
tinytex::install_tinytex()
# # to uninstall TinyTeX,
run
tinytex::uninstall_tinytex()
#
```

# Chapter 3

# Equations

## 3.1 Inline equations

...are enclosed by simple `$    $`, like this:

`$\tilde h(\omega) = \int_{-\infty}^{\infty}\,e^{i\omega t'} h(t') \, dt'\,$`

which produces this output: $\tilde h(\omega) = \int_{-\infty}^{\infty} e^{i\omega t'} h(t')\, dt'$ .

## 3.2 Display equations

...without numbers can be enclosed by double `$$    $$`, like this:

`$$\tilde h(\omega) = \int_{-\infty}^{\infty}\,e^{i\omega t'} h(t') \, dt'\,.$$`

which produces

$$\tilde h(\omega) = \int_{-\infty}^{\infty} e^{i\omega t'} h(t')\, dt' \,.$$

## 3.3 Equation labels

To label an equation with `name` use the format `(\#eq:name)`. To cite that equation use the format `\@ref(eq:name)`

The equation label has to appear after the body of the equation code, like this:

```
\begin{equation}
  \tilde h(\omega) = \int_{-\infty}^{\infty}\,e^{i\omega t'} h(t') \, dt'
  \label{eq:binom}
\end{equation}
```

$$\tilde{h}(\omega) = \int_{-\infty}^{\infty} e^{i\omega t'} h(t') \, dt' \tag{3.1}$$

Then you can cite the equation, like this: (3.1).

## 3.4  Equation numbering

There is some weirdness about how equation numbering is handled in the PDF versus the HTML versions of the book.

In the PDF output equations are numbered by default. Every line of an equation will get numbered except if it

1. is inside `$$ $$`.
2. has `\notag` at the end of line, before the `\\`.
3. is in an `{equation*}` or `{align*}` environment where there are no labels
4. is in a `{split}` environment with a single label

But in the HTML output an equation is unnumbered by default, except if it contains an explicit equation label. For more details see here.

So, to make sure we get the same numbering in PDF and HTML we should do this:

**An unnumbered display equation** should be enclosed with `$$ $$` or in environments `{equation*}` or `{align*}`.

**A numbered display equation** should include a single label.

Here are some examples:

This equation will get a number in PDF but not in HTML, **which is a mistake!**

$$\tilde{h}(\omega) = \int_{-\infty}^{\infty} e^{i\omega t'} h(t') \, dt' \tag{3.2}$$

This gets no number in PDF or HTML:

$$\tilde{h}(\omega) = \int_{-\infty}^{\infty} e^{i\omega t'} h(t') \, dt'$$

And this gets a number in both PDF and HTML:

$$\tilde{h}(\omega) = \int_{-\infty}^{\infty} e^{i\omega t'} h(t') \, dt' \tag{3.3}$$

## 3.5 Multi-line equation with multiple labels

Here is a long equation stretching over several lines, first with a single number

$$
\begin{aligned}
\text{Var}(\hat{\beta}) &= \text{Var}((X'X)^{-1}X'y) \\
&= (X'X)^{-1}X'\text{Var}(y)((X'X)^{-1}X')' \\
&= (X'X)^{-1}X'\text{Var}(y)X(X'X)^{-1} \\
&= (X'X)^{-1}X'\sigma^2 IX(X'X)^{-1} \\
&= (X'X)^{-1}\sigma^2
\end{aligned}
\tag{3.4}
$$

...then with multiple numbers

$$
\begin{aligned}
\text{Var}(\hat{\beta}) &= \text{Var}((X'X)^{-1}X'y) \\
&= (X'X)^{-1}X'\text{Var}(y)((X'X)^{-1}X')' \tag{3.5} \\
&= (X'X)^{-1}X'\text{Var}(y)X(X'X)^{-1} \tag{3.6} \\
&= (X'X)^{-1}X'\sigma^2 IX(X'X)^{-1} \tag{3.7} \\
&= (X'X)^{-1}\sigma^2 \tag{3.8}
\end{aligned}
$$

I wrote some grep routines that produce automatic equation labels, for example in the study guide which contains several hundred equations. Every one of those has a label, even though we rarely cite those labels.

# Chapter 4

# Python

## 4.1 A normal R code chunk

```r
library(reticulate)
matplotlib <- import("matplotlib")
matplotlib$use("Agg", force = TRUE)
```

```python
x = 42
print(x)
```

```
## [1] 42
```

## 4.2 Modify an R variable

In the following chunk, the value of x on the right hand side is 42, which was defined in the previous chunk.

```python
x = x + 12
print(x)
```

```
## [1] 54
```

## 4.3 A Python chunk

This works fine and as expected.

```python
print('Python version = ', sys.version)
```

```
## Python version =  3.5.2 (default, Nov 12 2018, 13:43:14)
## [GCC 5.4.0 20160609]
```

```
x = 42 * 2
print(x)
```

```
## 84
```

The value of `x` in the Python session is 84. It is not the same `x` as the one in R.

## 4.4  Modify a Python variable

```
x = x + 18
print(x)
```

```
## 102
```

Retrieve the value of `x` from the Python session again:

```
py$x
```

```
## [1] 102
```

Assign to a variable in the Python session from R:

```
py$y = 1:5
```

See the value of `y` in the Python session:

```
print(y)
```

```
## [1, 2, 3, 4, 5]
```

## 4.5  Python graphics

You can draw plots using the **matplotlib** package in Python.

```
import matplotlib.pyplot as plt
plt.plot([0, 2, 1, 4])
plt.show()
```