Making Lecture Notes with Bookdown

Vicky Scowcroft

2022-08-04

Contents

Introduction

This guide walks through how to create html lecture notes using **R**, **Markdown** and **Bookdown**. This format is preferred as LaTeX doesn't play nicely with accessibility tools like screen readers, especially for maths heavy content.

System Requirements

To use bookdown, you will need to have R and R studio installed (perhaps R studio is not a requirement, but it makes things a lot easier). These can both be installed from the DDAT Self Service app on a Mac, or R can be downloaded directly from here and R Studio from here. You will also need **pandoc**. This may already be installed on your system as part of the Anaconda distribution.

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

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

Caveat

The instructions in this guide are what worked for me, on a Mac using R version 4.0.2, R Studio version 1.3 and pandoc version 2.2.3.2. This guide also assumes that you're familiar with using the command line.

Chapter 1

Getting started

The quickest way to get started is to use the bookdown demo as a template. This page gives instructions for how to do this.

The bookdown: Authoring Books and Technical Documents with R Markdown is the main documentation for bookdown. I won't repeat a lot of the information here.

This guide will walk through the steps I used to convert my LaTeX lecture notes for PH40112 to the bookdown version here. The files used to create the PH40112 notes are available on github and you're welcome to use those as a starting point.

Chapter 2

Converting from LaTeX notes

2.1 Converting tex files

If you already have LaTeX versions of your notes you can convert these to markdown using pandoc Pandoc won't create *perfect* versions of your notes - you'll most likely have to do a bit of tweaking, but it gets most of the way there.

You can convert a tex file to Rmd via the command

```
pandoc -f latex -t markdown input-tex.tex -o output-md.Rmd
```

where `input-tex.tex` is the name of your tex file and `output-md.Rmd` is the name of your output Rmd file.

Pandoc converts one tex file at a time. It doesn't understand how to process a "master" tex file with `input` or `insert` LaTeX commands. Each file will need to be processed separately.

2.2 Batch conversion

If you want to convert a batch of latex files you can script the process.

2.3 Tidying up the Rmd file

Pandoc will have done most of the work in getting your tex file converted to markdown. However, depending on how complex the original file was, you may end up with some tidying to do.

Things that typically will need fixing are:

- Section etc labels - if you used underscores in any of your latex labels you should change these to '-' in the markdown file. The underscores confuse markdown. Section **??** discusses sections, chapters, etc.
- Figures - the default pandoc conversion is quite limited, so here we'll use a more adaptable figure environment. In practice its easiest to just copy-paste the example code in Chapter **??** and edit the relavent bits.
- Equations - the equations themselves should be fine, but if you want to have equation numbers you'll need to do some tidying. See Chapter **??** for the syntax.
- Tables - tables have a different format but it's pretty easy to convert a latex table using find/replace. See Chapter **??**.
- References - generally easy to fix. See Chapter **??**.

2.4   Notes with gaps

Think carefully about whether you want to provide notes with gaps. While they may serve a purpose when the students have a hard-copy of the notes in a live lecture, they will most likely be using your notes on a screen this year.

It is possible to create notes with gaps for things like equations etc for the students to fill in. To do this you need to add some extra code to the `index.Rmd` file:

```
{r setup, include=FALSE}
library(knitr)
knit_engines$set(asis = function(options) {
  if (options$echo && options$eval) knit_child(text = options$code)
})
```

Then to "hide" e.g. an equation but still keep the same numbering, edit your equation to the following format:

```
\begin{equation}
```{asis, echo=FALSE}
\Delta s^2 = \Delta x_{1}^{2} + \Delta x_{2}^{2}
```
\label{eq:dist01}
\end{equation}
```

Here the `asis` command and back-ticks wrap around **only** the equation itself, rather than the whole `equation` object. If you wrap the `asis` around the `\begin{equation}` and `\end{equation}` the equation numbering will disappear too.

2.5 Using the `titlesec` package

If you're using the `titlesec` latex package you'll need to add an additional line to the yaml information on your index.Rmd file:

```
subparagraph: yes
```

This should fix any compilation errors related to the `titlesec` package when you're producing a pdf output.

Source: Stack Overflow

Chapter 3

Chapters, sections, etc

3.1 File Structure

Bookdown organises your notes into a book. The main page you see when you open a bookdown page is the `index.Rmd` file.

Each chapter in the book has it's own markdown file. Each chapter filename should start with a number, and will be ordered numerically. For this book, the files are

```
index.Rmd
01-intro.Rmd
02-latex-conversion.Rmd
03-chapters.Rmd
... etc
```

3.2 Headings

The chapter title and label are given in the first line of the file.

```
# Chapters, sections, etc {#sec:chapters}
```

The `#` at the start of the line denotes a chapter heading. The label is given in `{#sec:chapters}`.

You can make sections either using two `#` at the start of the line, or with dashes under the section title:

```
Headings {#sec:headings}
--------
```

Note that the `{#sec:label}` syntax is still the same - you don't need to say whether it's a chapter, section, subsection etc.

Subsections are done similarly, with either three # at the start of the line.

You can change the section heading from 'Chapter' to e.g. section by editing the following in the `_bookdown.yml` file:

```
language:
  ui:
    chapter_name: "Section "
```

3.3   Cross referencing

You can reference other chapters, sections etc throughout the book with the following syntax:

```
Section \@ref(sec:cross-ref) is this section.
```

Which shows up as "Section **??** is this section.".

When cross-referencing, you don't need the # before `sec`. That's only used to define the label.

Chapter 4

Figures

4.1 Adding figures

Figures and captions can be included, but the syntax is quite different to LaTeX.

Assuming that you would add a figure to a LaTeX document with the following code:

```
\begin{figure}
\begin{centering}
\includegraphics[width=0.7\textwidth]{Images/ho-tension.png}
\caption{This is a figure caption.}
\label{fig:ho-plot}
\end{centering}
\end{figure}
```

To include the figure in markdown the syntax is as follows:

```
```
{r echo=FALSE, ho-plot, out.width='70%', fig.show='hold', fig.cap="This is a figure caption."}
knitr::include_graphics("Images/ho-tension.png")
```
```

Important parts of this command:

- `echo=FALSE` prevents the code being used to display the figure being shown.
- `ho-plot` is your figure label.
- `out-width` is the output width of the figure.
- `fig.cap` is the figure caption.
- `knitr::include_graphics` gives the location of the image file.
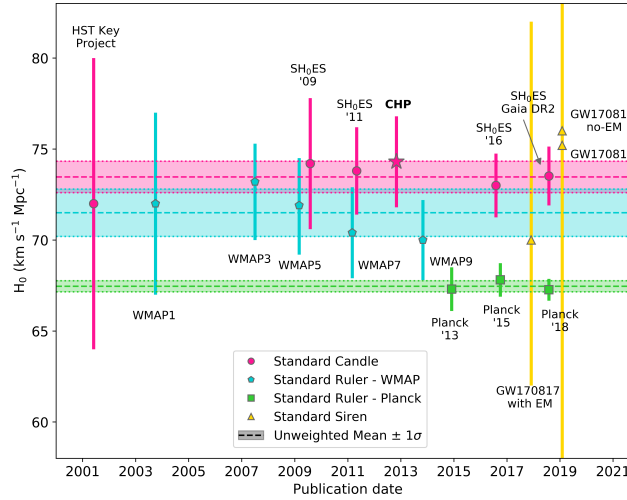
Figure 4.1: This is a figure caption.

4.2   Figure captions

Figure captions are controlled by the `fig.cap` setting. Your caption should be inside the double quotes. Remember to use \" to escape any " characters you may use inside your caption.

You can include cross-references and citations inside figure captions. See Sections **??**, **??**, and **??** for how to label and cross-reference figures, equations, and tables. Citations are covered in Section **??**.

4.3   Figure numbering and referencing

Figures with labels and captions are numbered according to the section (e.g. Fig **??** in this case).

To reference a figure use the syntax

```
\@ref(fig:label)
```

where `fig:label` would be `fig:ho-plot` in this case.

You can cross reference between sections/chapters, so make sure to use unique figure labels.

Chapter 5

Equations

5.1 Equation syntax

The syntax for equations is similar (but not identical) to LaTeX.

LaTeX code:

```
\begin{equation}
\label{eqn:friedman}
    \left(\dfrac{\dot{a}}{a}\right)^2 + \dfrac{kc^2}{a^2} = \dfrac{8\pi G}{3}\rho
\end{equation}
```

Rmd code:

```
\begin{equation}
    \left(\dfrac{\dot{a}}{a}\right)^2 + \dfrac{kc^2}{a^2} = \dfrac{8\pi G}{3}\rho
(\#eq:friedman)
\end{equation}
```

$$\left(\frac{\dot{a}}{a}\right)^2 + \frac{kc^2}{a^2} = \frac{8\pi G}{3}\rho \tag{5.1}$$

You can also use the latex

```
\begin{align}
...
\end{align}
```

format for equations. If you're converting from LaTeX to markdown with pandoc it may convert equations to

```
\begin{aligned}
...
\end{aligned}
```

which also works.

LaTeX subequations and intertext

I haven't been able to get subequations and intertext to work in bookdown. LaTeX equations of the form

```
\begin{subequations}\begin{align}
\vec{E} &= \left( x,t \right)
\intertext{and in 3 dimensional space as}
\vec{E} &= \left( x,y,z,t \right)
\end{align}\end{subequations}
```

should be written as separate equations with the text between written outside the equation environment, e.g.

```
\begin{align}
\vec{E} &= \left( x,t \right)
\end{align}
and in 3 dimensional space as
\begin{align}
\vec{E} &= \left( x,y,z,t \right)
\end{align}
```

5.2   Equation numbers and labels

The syntax for the maths is the same, but the labelling changes. To label and equation add

```
(\#eq:label)
```

just before `end{equation}`. Only equations with labels will be numbered. If you don't want numbers then don't label the equations, but numbers are helpful.

5.3   Cross referencing equations

The syntax for cross-referencing equations is similar to sections and figures, i.e.

```
Eqn. \@ref(eq:friedman) is the Friedman equation
```

will give "Eqn. (**??**) is the Friedman equation".