Preparing your thesis with LaTeX

Jack Walton

October 18, 2019

Newcastle University

Introduction

Follow the leader

- These slides contain links to exercises and further reading
- You can follow along with these slides on my website

jwalton.info/teaching

A little about me...

- I am a 4th year PhD student based in the School of Maths, Stats & Physics
- I have $\sim$4 years of LaTeX experience
- I also teach the PGRDP course Introduction to git and GitHub

A little about you…

- This course is intended for those who already have *some* LaTeX experience
- If you want to brush up on the basics then SAgE offer an introduction to LaTeX workshop (*not* taught by me)

The game plan

- Angela will arrive to check attendance at 1pm
- I am more than happy informing Angela of any skivers (*so don't skive*)
- Exercises (and their solutions) are included to break-up the monotony of me (talking)

Table of contents

Motivation

But I already know LaTeX!

- Many of us learn LaTeX “as we go”
- As such, it is easy to get into bad habits
- It’s even easier to miss out on useful packages and features

For inspiration…

- The Divine Liturgy of Saint John Chrysostom
- PhD thesis, Aaron Turon
- Trees, maps, and theorems, Jean-luc Doumont
- The slides, exercises and solutions produced for this course (source code)

Managing large documents

Modular LaTeX

- For smaller projects it is *okay* to keep everything in a single `.tex` file
- For more involved projects (your thesis) this approach *quickly* becomes cumbersome
- The `\include` command makes it possible to break your document down into smaller chunks
- Working with smaller chunks is more manageable

Structure

An example structure for a thesis project could look like the following:

```
thesis/
├── thesis.tex
├── chapters/
│   ├── chapter_1.tex
│   ├── chapter_2.tex
│   └── chapter_3.tex
├── internal/
│   └── preamble.tex
├── fig/
│   └── science.png
└── references.bib
```

Example thesis.tex

```
\documentclass[12pt]{report}

\include{internal/preamble}

\begin{document}

\include{chapters/chapter_1}
\include{chapters/chapter_2}
\include{chapters/chapter_3}

\bibliography{references}

\end{document}
```

Example internal/preamble.tex

```
% Preamble, packages, commands etc.
\usepackage{microtype}
\usepackage{booktabs}
\usepackage{cleveref}
\usepackage{graphicx}
⋮

% Make it easier to include figures
\graphicspath{{fig/}}
```

Example chapters/chapter_1.tex

```
\chapter{Literature review}
\label{cha:lit_review}

Here's stuff others did which I don't really
understand\ldots
```

Compile a single chapter

`\includeonly` allows the compilation of a single chapter, without messing up references, page numbers etc.

```
\documentclass[12pt]{report}

\include{internal/preamble}
\includeonly{chapters/chapter_2}

\begin{document}

\include{chapters/chapter_1}
\include{chapters/chapter_2}
\include{chapters/chapter_3}
⋮
```

masthesis.sty

- A thesis template for MSP students
- The template is modular and has a structure *similar* to the one given above
- For non-MSP students, or those who would like a different style, the ‘classic thesis’ style is a good option

Version control

- Version control allows you to track and manage changes in code, and collaborate with others
- I'd recommend using version control to help manage your thesis
- Plug: a colleague and I are teaching an upcoming PGRDP workshop Introduction to Git and GitHub

Spell checking

Spell checking `.tex` files is complicated by latex commands.

For those comfortable working at the command line I'd recommend aspell (or ispell or hunspell).

Interactive spell-check:

```
$ aspell -t -c chapters/chapter1.tex
```

Non interactive spell-check (lists mistakes):

```
$ cat chapters/chapter1.tex | aspell list -t
```

Custom dictionary and commands to ignore can be added with `--add-extra-dicts` and `--conf` respectively

Spell checking

Some IDEs have inbuilt spell checkers:

- Texmaker (checks contents of commands still)
- Texstudio (seems to have the best spellchecker)

More generally: here is a list of editors and their features

Word count

For final submission (it will creep up on you, I promise) you *need* to submit a word count.

Counting words in a .tex file is again complicated by the presence of latex commands.

For command line users I'd recommend trying detex and wc:

```
$ detex -le equation,table thesis.tex | wc -w
```

Word count

- Online tool (chapters counted one at at time)
- Texmaker's integrated pdf viewer has word count (right click pdf)
- Texstudio (tools $\rightarrow$ analyse text; chapters one at a time)

Exercise 1

Custom commands

Simple macros

Used to simplify repetitive and/or complex formatting.

Usually specified in the preamble

```
\newcommand{\name}{definition}
```

Simple macros: an example

```
\newcommand{\R}{\mathbb{R}}

The set of real numbers are usually represented
by a blackboard capital r: $\R$.
```

The set of real numbers are usually represented by a blackboard capital r: $\mathbb{R}$.

Macros with parameters

Macros can also be constructed to accept parameters:

```
\newcommand{\name}[# params]{definition}
```

Macros with parameters: an example

```
\newcommand{\bb}[1]{\mathbb{#1}}

Other numerical systems have similar notations.
The complex numbers $\bb{C}$, the rational
numbers $\bb{Q}$ and the integer numbers
$\bb{Z}$.
```

Other numerical systems have similar notations. The complex numbers $\mathbb{C}$, the rational numbers $\mathbb{Q}$ and the integer numbers $\mathbb{Z}$.

Macros with default parameters

It is also possible to define macros which take default parameters:

```
\newcommand{\name}[# params][default #1]{def.}
```

Macros with default parameters

```
\newcommand{\plusbinomial}[3][2]{(#2 + #3)^#1}

We make a new command to save time writing
expressions of the form $\plusbinomial{x}{y}$
and $\plusbinomial[4]{a}{b}$.
```

We make a new command to save time writing expressions of the form $(x + y)^2$ and $(a + b)^4$.

Exercise 2

Managing a bibliography

BibTeX

BibTeX can be used to manage bibliographies. (BibLaTeX is a more sophisticated alternative.)

- BibTeX entries are stored in a `.bib` file
- I recommend maintaining a *single* centralised `.bib` file for the duration of your PhD.

BibTeX entries

A list of entry types which BibTeX understands can be found here.

```
@book{knuth84,
 title="The texbook",
 author="{Donald Ervin} Knuth and Duane Bibby",
 volume="3",
 year="1984",
 publisher="Addison-Wesley Reading"
}
```

Referencing with BibTeX

- References are included as `\cite{knuth84}`, where `knuth84` is the `title` of a BibTeX entry
- Include your `.bib` file with `\bibliography{references}`, where `references` is the name of your file

\usepackage{natbib}

- natbib can be used to implement author-year citations.
- Introduces commands `\citep` and `\citet`, to cite in parenthesis or text.
- `\citep*` and `\citet*` print full author list
- Multiple citations can be made as `\citep{paper1, paper2}`

Compiling with BibTeX

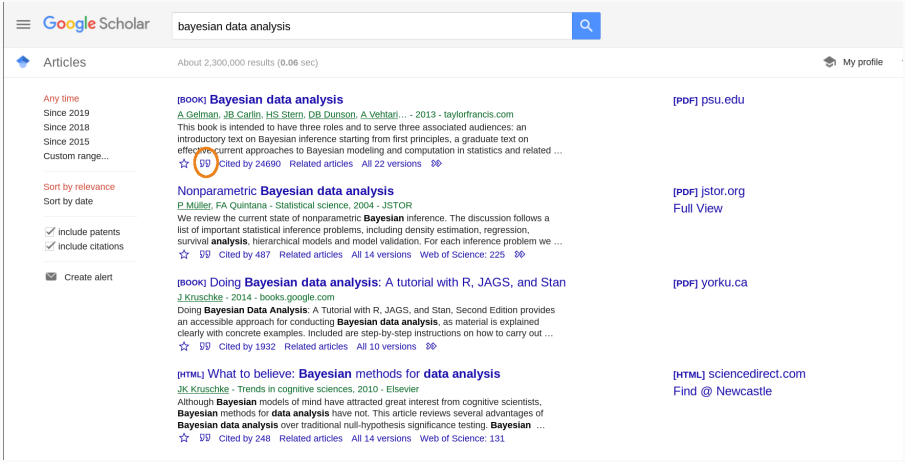
BibTeX adds extra complexity to the processing of your manuscript. You will have to run LaTeX a number of times.

1. pdflatex thesis.tex
2. bibtex thesis.aux
3. pdflatex thesis.tex
4. pdflatex thesis.tex

A Makefile can simplify compilation. However, I'd recommend using latexmk.

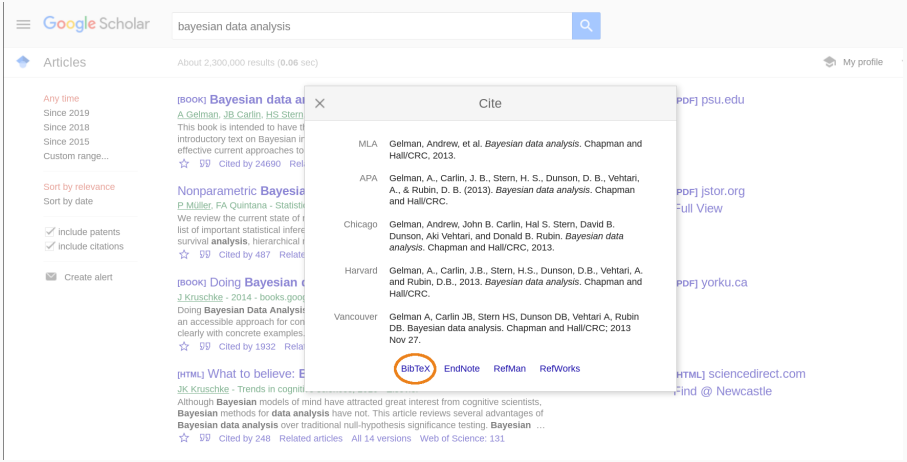
Citations from Google Scholar

Google scholar can be used to export citations easily.



Citations from Google Scholar

Google scholar can be used to export citations easily.



Packages: a few favourites

\usepackage{cleveref}

cleveref formats cross-references automatically

See Figure 1.

Figure 1: TEX the Lion.

\usepackage{cleveref}

```
% Reference as Figure 1, instead of fig. 1
\usepackage[capitalise,noabbrev]{cleveref}
 ⋮
See \cref{fig:lion}.

\begin{figure}
 \centering
 \includegraphics[width=0.4\textwidth]{Lion.png}
 \caption{\TeX\ the Lion.}
 \label{fig:lion}
\end{figure}
```

\usepackage{hyperref}

- Adds hypertext links to cross-references.
- See e.g. this link to the Table of Contents, the links in the table of contents and the external hyperlinks throughout.
- `hyperref` takes many options to alter how links are displayed

\usepackage{booktabs}

Booktabs can be used to enhance default tabular.

Item		
Animal	Sold	Price ($)
Gnat	per gram	13.65
	each	0.01
Gnu	stuffed	92.50
Emu	stuffed	33.33

Table 1: Default LaTeX table.

\usepackage{booktabs}

```
\begin{tabular}{|l|l|r|}
 \hline
 \multicolumn{2}{|c|}{Item} & \\\cline{1-2}
 Animal        & Sold      & Price (\$) \\\hline
 Gnat          & per gram  & 13.65      \\
               & each      & 0.01       \\
 Gnu           & stuffed   & 92.50      \\
 Emu           & stuffed   & 33.33      \\\hline
\end{tabular}
\caption{Default \LaTeX\ table.}
```

\usepackage{booktabs}

Item		
Animal	Sold	Price ($)
Gnat	per gram	13.65
	each	0.01
Gnu	stuffed	92.50
Emu	stuffed	33.33

Table 2: Booktabs table and styling.

\usepackage{booktabs}

```
\begin{tabular}{llr}
 \toprule
 \multicolumn{2}{c}{Item} & \\\cmidrule{1-2}
 Animal       & Sold     & Price (\$) \\\midrule
 Gnat         & per gram & 13.65      \\
              & each     & 0.01       \\
 Gnu          & stuffed  & 92.50      \\
 Emu          & stuffed  & 33.33      \\
 \bottomrule
\end{tabular}
\caption{Booktabs improves table spacing.}
```

\usepackage{pgfplotstable}

pgfplotstable can read data in from file (e.g. a `.csv` file) and automatically format the data as a table.

Consider that I have some `.csv` file:

```
Element,    Number,    Mass
H,          1,         1.00794
He,         2,         4.00260
Li,         3,         6.94100
Be,         4,         9.01218
```

\usepackage{pgfplotstable}

	Atomic	
Element	Number	Mass
H	1	1.00794
He	2	4.00260
Li	3	6.94100
Be	4	9.01218

Table 3: pgfplotstable can read input files.

\usepackage{pgfplotstable}

```
\pgfplotstabletypeset[
  col sep=comma,
  string type,
  every head row/.style={%
    before row={%
      \toprule
      & \multicolumn{2}{c}{Atomic} \\
      \cmidrule{2-3}
    },
    after row={\midrule}
  },
  every last row/.style={after row=\bottomrule}
  ]%
  {assets/elements.csv}
```

\usepackage{pgfplotstable}

pgfplotstable can round numbers as desired:

	Atomic	
Element	Number	Mass
H	1	1.008
He	2	4.003
Li	3	6.941
Be	4	9.012

Table 4: pgfplotstable understands precision and rounding.

\usepackage{pgfplotstable}

```
\pgfplotstabletypeset[
  col sep=comma,
  columns/Number/.style={string type},
  columns/Element/.style={string type},
  columns/Mass/.style={fixed zerofill,
                       precision=3},
⋮
⋮  (As in earlier example)
⋮
  \caption{pgfplotstable understands precision
           and rounding.}
```

Even *more* table generators

In addition to pgfplotstable there are various other table generators:

- pandas.DataFrame.to_latex (Python users)
- xtable (R users)
- Excel2latex (Excel users)
- matrix2latex (Matlab users)

A few more packages…

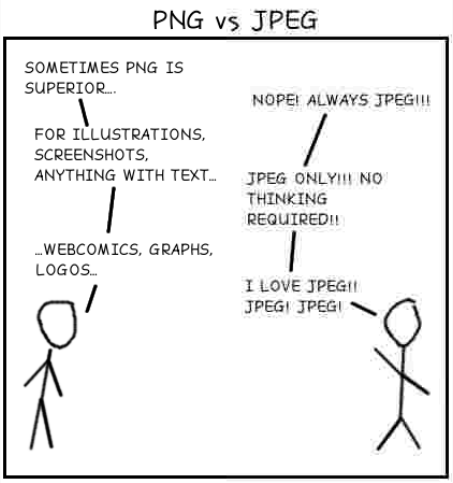
- tikz
- standalone
- fancyhdr
- multirow
- ifdraft
- titlesec
- microtype
- natbib
- geometry
- todonotes

Exercise 3

Common mistakes

Image formats

- Do *not* use `.jpeg` files for plots (`.jpeg` compresses text poorly)
- If you must use a raster format use `.png`
- Ideally use a vector format e.g. `.pdf`



Avoiding image scaling

- Avoid scaling your plots using the `width` argument of `\includegraphics`
- Using `width` will scale the font sizes in your plot, making it difficult to control font size
- Aim to create your plot with the *exact* dimensions you need for your document
- The logic to achieve this is the same for whatever plotting software you use. Here I outline an implementation for python.

Typesetting maths

Brackets should be large enough to completely enclose all they contain.

$$(\sum_{i=1}^{n-1} i) + n$$

```
(\sum_{i=1}^{n-1} i) + n
```

$$\bigg(\sum_{i=1}^{n-1} i \bigg) + n$$

```
\bigg( \sum_{i=1}^{n-1} i \bigg) + n
```

Typesetting maths

```
$a, b, c, d, e \text{ and } f$
```
a, b, c, d, e and f

```
$a$, $b$, $c$, $d$, $e$ and $f$
```
a, b, c, d, e and f

```
$i=1,...,10$
```
$i = 1, ..., 10$

```
$i=1,\ldots,10$
```
$i = 1, \ldots, 10$

```
$sin(x)^2 + cos(x)^2 = 1$
```
$sin(x)^2 + cos(x)^2 = 1$

```
$\sin(x)^2 + \cos(x)^2 =1$
```
$\sin(x)^2 + \cos(x)^2 = 1$

Hyphen, en-dash and em-dash (-, –, —)

- The hyphen (`-`) is used to join words in a compound construction. “A long-term solution”
- An en-dash (`--`) appears in page ranges. “See pages 1–3”
- An em-dash (`---`) is typically used as a stand-in for a comma or parenthesis to separate out phrases. “Against all odds, Boris — the class clown — became prime minister.”

Quotes

LaTeX requires you to use separate markup for opening and closing quotes.

Opening quotes are ``

Closing quotes are ''

Quotes should look “like this” not ”like this”.

Capitalisation in BibTeX

Your BibTeX style will handle most capitalisation. For some words (names, places, ...) capitalisation must be ensured

```
@book{springer57,
  title="Introduction to {R}iemann surfaces",
  author="Springer, George",
  volume="473",
  year="1957",
  publisher="Addison-Wesley Reading"
}
```

Conclusion

Resources

- First point of call: stack exchange
- The not so short introduction to LaTeX 2ε
- More Math into LaTeX 4th edition (hard copies available at library)
- CTAN: comprehensive TeX Archive Network
- You should be aware of: official university guidelines

Exercise solutions

The solutions to today's exercises are included online (and will remain so in the future):

- Solutions 1
- Solutions 2
- Solutions 3

Perspective

Leslie Lamport, initial developer of LaTeX, was asked what three LaTeX mistakes people should stop making:

1. Worrying too much about formatting and not enough about content.
2. Worrying too much about formatting and not enough about content.
3. Worrying too much about formatting and not enough about content.

Source

Feedback and the future

- Please complete workshop evaluation
- Feedback is anonymous