Introduction to LaTeX

Guillermo Toral & Weihuang Wong

MIT

September 15, 2017

These slides build on materials from previous years by Dan de Kadt and Elizabeth Keysner, and on "The Not So Short Introduction to LaTeX".

What we'll cover today

1. Introduction
2. LaTeX Language
3. Setting up a LaTeX document
4. LaTeX environments & commands
5. LaTeX and R
6. Referencing objects
7. Reference management in LaTeX using BibTeX
8. Troubleshooting

Where we are

What's LaTeX?

- LaTeXis a typesetting system that produces high-quality documents – anything from letters, papers, dissertations, posters, slides (like these ones), resumes, etc.
- LaTeX, unlike Word or other normal word processor, is not a WYSIWYG ("what you see is what you get") system. So using it requires a bit of a change of mindset.
- In LaTeX, you write the code, and then the software compiles or creates the document as a PDF file.
- Like R, there are hundreds of add-ons available that you can use.

Why use LaTeX?

- LaTeX makes math much easier to write and to display well:

$$Y_i = \alpha + X_i\beta + \varepsilon_i$$

$$f(y) = \frac{1}{y!\Gamma(\alpha)\beta^{\alpha}} \int_{-\infty}^{\infty} \mu^{y+\alpha-1} exp\left(-\mu\left[1+\frac{1}{\beta}\right]\right)$$

- LaTeXworks well with R: you can easily input plots and tables from R, and you can even write LaTeX documents directly from R, through R Markdown
- LaTeXmakes it easy to manage large documents, with references and bibliography, cross-references and links inside the document, etc.
- LaTeXis used widely in the scientific community, and some people see it as a signal of credibility.

Things to keep in mind about LaTeX

- LaTeXis a skill that takes some effort – and patience – to build.
- Grad school gives you the time and incentives to do so – invest some effort, and it'll pay off.
- LaTeXis a good tool for some purposes – once you master it, it becomes very easy to use, and it allows you to do cool things that would be harder to do without LaTeX.
- LaTeXis not so good for other purposes, like spellchecks or review tools, although you can also do those in LaTeX.
- LaTeXis a tool – the more you have, the better.

Where we are

Commands & spaces

Commands:

- LaTeX commands start with \ and continue with letters or with one non-letter.
- Many commands have a "starred" version, where the name of the command is followed by "*", and usually tells LaTeX to omit numbering (e.g. do not number this equation, or this section).

White spaces:

- "Whitespace" characters (whether one space, one tab, or several spaces / tabs) are all treated as one space.
- An empty line or multiple empty lines are all treated as an empty line (which ends a paragraph).
- Spaces after commands are ignored, so it's good to write { } after them: LaTeX{ }.
- You can insert a line break with \\ or `\newline`. For a page break, use `\newpage`.

Reserved characters

There are some "reserved" characters in LaTeX:

- \ says: read text following this as an operation.
- $ says: open an inline math environment. Put another to close it.
- % says: ignore text that follows – allows you to comment your code.
- & says: position this to line up with other position indicators.
- {} says: group these items.
- _ says: write what follows as underscript.
- ^ says: write what follows as upperscript.

If you actually want to print these characters, you need to use "\" before them (e.g. \&), or "textbacklash" for the backslash.

Where we are

Key components of a LaTeX document

LaTeX documents have several key components:

- Preamble: declares the type of the document, gives its parameters, and loads packages
- Titling: declares the document's title, author(s), and date.
- Document itself, which may include a title page, sections, subsections, bibliography, etc.

A minimal LaTeX document includes the following code:

```
\documentclass[options]{class}
\usepackage[options]{packages}
\begin{document}
Your document.
\end{document}
```

LaTeX mechanics

Here are the steps you usually take for creating a document with LaTeX:

- Write in your `.tex` file, and save it.
- Compile the LaTeX file into a PDF by pressing the corresponding button in the TeX software you use. Sometimes you need to do this several times, especially if you have sections or bibliography in the document.
- A new PDF file (and a number of auxiliary files terminating in .aux, .out, .toc, etc.) appear in the same folder where your .tex file is.
- It is useful to compile your PDF from time to time as you write, to make sure it's compiling correctly.
- You can use your Tex and your PDF reader side by side, and have the PDF refresh automatically each time you compile, so you can see in real time how your output looks like (this is what TexMaker does).

Let's practice!

Let's start by building a minimum article document.

- Open the `template.tex` file and fill it up: give it a document class, fill in the titling information, and add some text in the document. Try structuring your text with sections, by using `\section{Section name}`. You can also use `\subsection{}` and `\subsubsection{}`.
- You can also add some `\footnote{}`.
- When you're ready, save the file and compile it.
- Open the resulting PDF file.

An empty template for a .tex file

```
1   \documentclass[•]{•}
2
3   \begin{•}
4
5   \title{•}
6   \author{•}
7   \date{\today}
8
9   \maketitle
10
11  \section{•}
12
13  \end{document}
14
```

A simple .tex file

```
1   \documentclass{article}
2
3   \begin{document}
4
5   \title{My first \LaTeX{} document}
6   \author{Guillermo Toral \& Weihuang Wong}
7   \date{\today}
8
9   \maketitle
10
11  \section{A first section}
12
13  Some text
14
15  \section{A second section}
16
17  Some more text\footnote{...and a footnote}.
18
19
20  \section*{A third, un-numbered section}
21
22  (Perhaps an appendix)
23
24
25  \end{document}
```

The resulting PDF file

My first LaTeX document

Guillermo Toral & Weihuang Wong

September 10, 2017

1 A first section

Some text

2 A second section

Some more text[1].

A third, un-numbered section

(Perhaps an appendix)

[1] ...and a footnote

1

Where we are

What's an environment?

LaTeX uses "environments" to format content:

- When you "\begin{environment}" a set of rules is applied to the content that appears *within that environment*.
- When you "\end{environment}" those rules no longer apply.
- Environments can be embedded within environments.
- The biggest environment is the "document" environment, which bounds the entire document you make.
- Within the "document" environment you might use a bullet-point environment ("itemize"), and within those bullet points you might you a picture environment ("figure"). Environments can be used for including equations, tables, plots, text in different sizes or alignments, abstracts, long quotes, etc.

Using an environment:

Here's how you can use a basic text environment, "itemize":

```
\begin{itemize}
\item Math camp.
\item \LaTeX {} workshop.
\item Quant I.
\end{itemize}
```

Which produces the following:

- Math camp.
- LaTeX workshop.
- Quant I.

If we substituted "enumerate" for "itemize" (make sure to use the enumitem package!) we'd get a numbered list.

Let's practice!

Now practice creating and filling an "enumerate" environment within your practice `.tex` file:

- Load the "enumitem" package in the file's preamble.
- Begin an"enumerate" environment. It takes some options:
- E.g., [label = (\arabic*)] or [label = \alph*.]
- Create items
- End the "enumerate" environment.
- Optional: Wrap the whole "enumerate" environment in a "center" environment.

Math Environments

Let's now add math to our document:

1. Load the amsmath and amssymb packages in your document's preamble.
2. Begin an "align*" environment
3. Type in some math: Y_i &= \alpha + \beta x_i + \varepsilon_i
4. And a second line: Y_i - \varepsilon_i &= \alpha + \beta x_i
5. End the "align*" environment.

This will produce a set of aligned equations:

$$Y_i = \alpha + \beta x_i + \varepsilon_i$$
$$Y_i - \varepsilon_i = \alpha + \beta x_i$$

Math environments

We can also create a *non-aligned* math environment using $ and $$.

- $ creates in-line math, $$ creates a new line. You must begin and end with the same symbol ($ or $$).
- Example of in-line math with $: writing `$1+1=2$` will produce $1 + 1 = 4$.
- Example of new-line math with $$: writing `$$1+1=2$$` will produce

$$1 + 1 = 2$$

If you want new-line, numbered equations, but you don't want to align them, or you're writing only one equation, you can use the environment "equation".

Some Math Symbology

LaTeX code	Output	What to use it for
`y_x`	y_x	subscript
`y_{xxx}`	y_{xxx}	multiple-letter subscript
`y^x`	y^x	superscript
`y^{xxx}`	y^{xxx}	multiple-letter superscript
`\sqrt{x}`	$\sqrt{x}$	square root
`\sqrt[y]{x}`	$\sqrt[y]{x}$	y-th root
`y^{xxx}`	y^{xxx}	multiple-letter superscript
`\frac{x}{y}`	$\frac{x}{y}$	fraction
`\alpha`	α	Greek letters
`\sum_{k=0}^{n-1}a r^k`	$\sum_{k=0}^{n-1} ar^k$	sums
`\int_0^{\infty}`	$\int_0^{\infty}$	integrals
`\lim_{n \to \infty}`	$\lim_{n \to \infty}$	limits
`\hat{X}, \widehat{XY}`	$\hat{X}, \widehat{XY}$	hats
`\mathcal{R}`	$\mathcal{R}$	math font

Figure & table environments: Floats

The positioning of floats can be frustrating if you don't know how they work or don't use the tools you have to control them.

- Every float has an optional positionning parameter: `\begin{figure}[placement specifier]`.
- The default placement specificier, the one LaTeX uses if we don't specify anything, is "tbp", which stands for "at the top of the page", "at the bottom of the page", and "at a special page containing only floats".
- When LaTeX encounters a float, it tries to place it in the current place and if it can't, it puts it in a figures queue or in a tables queue, and every time it starts a new page, it tries to fill a special "float page" with floats from the queues. If it can't, it tries to place the first float in the queue. LaTeX maintains the order in the queue.
- You can use other placement specificers, like "h" (for here), or "!" (which basically says "override other parameters").
- To avoid floats appearing outside the section where they belong (or say beyond a certain paragraph), you can use `\FloatBarrier`.

LaTeX Commands

Commands are another way of taking in content and create output:

- No need to use "\begin" and "\end" with commands. Instead you simply use \COMMAND{YOUR INPUT}.
- We've already used a couple, like "\section" or "\item".
- Some commands take options in square brackets: \COMMAND[OPTIONS]{YOUR INPUT}

We can use commands to include graphics in our document:

1. Load the `graphicx` package.
2. Begin a "center" environment so the graph shows centered.
3. Use the "includegraphics" command. It takes some options and some arguments:
4. Options in square brackets: [scale = .5]
5. A filename in curly braces: {graph.pdf}
6. End the "center" environment.

Some useful commands

Font face

Command	*Declaration*	*Effect*
`\textrm{`*text*`}`	`{\rmfamily` *text*`}`	Roman family
`\textsf{`*text*`}`	`{\sffamily` *text*`}`	Sans serif family
`\texttt{`*text*`}`	`{\ttfamily` *text*`}`	`Typewriter family`
`\textmd{`*text*`}`	`{\mdseries` *text*`}`	Medium series
`\textbf{`*text*`}`	`{\bfseries` *text*`}`	**Bold series**
`\textup{`*text*`}`	`{\upshape` *text*`}`	Upright shape
`\textit{`*text*`}`	`{\itshape` *text*`}`	*Italic shape*
`\textsl{`*text*`}`	`{\slshape` *text*`}`	*Slanted shape*
`\textsc{`*text*`}`	`{\scshape` *text*`}`	SMALL CAPS SHAPE
`\emph{`*text*`}`	`{\em` *text*`}`	*Emphasized*
`\textnormal{`*text*`}`	`{\normalfont` *text*`}`	Document font
`\underline{`*text*`}`		Underline

The command (t*ttt*) form handles spacing better than the declaration (t*ttt*) form.

Font size

`\tiny` tiny
`\scriptsize` scriptsize
`\footnotesize` footnotesize
`\small` small
`\normalsize` normalsize
`\large` large
`\Large` Large
`\LARGE` LARGE
`\huge` huge
`\Huge` Huge

These are declarations and should be used in the form `{\small ...}`, or without braces to affect the entire document.

Where we are

Working with LaTeX and R: Tables

R code:

```
mod1<-lm(y~x1+x2)
mod2<-lm(y~x1*x2)
xtable(mod1,mod2)
OR
stargazer(mod1,mod2)
```

Note: xtable is a more generic function, will turn any matrix or data frame into a table. Stargazer is very flexible, and highly recommended.

Output from LaTeX:

	Model 1	Model 2
(Intercept)	-2.94^*	-0.24
	(0.40)	(0.36)
x1	5.54^*	5.03^*
	(0.06)	(0.06)
x2	6.32^*	1.50^*
	(0.32)	(0.49)
x1:x2		0.95^*
		(0.09)
N	100	100

Standard errors in parentheses

* indicates significance at $p < 0.05$

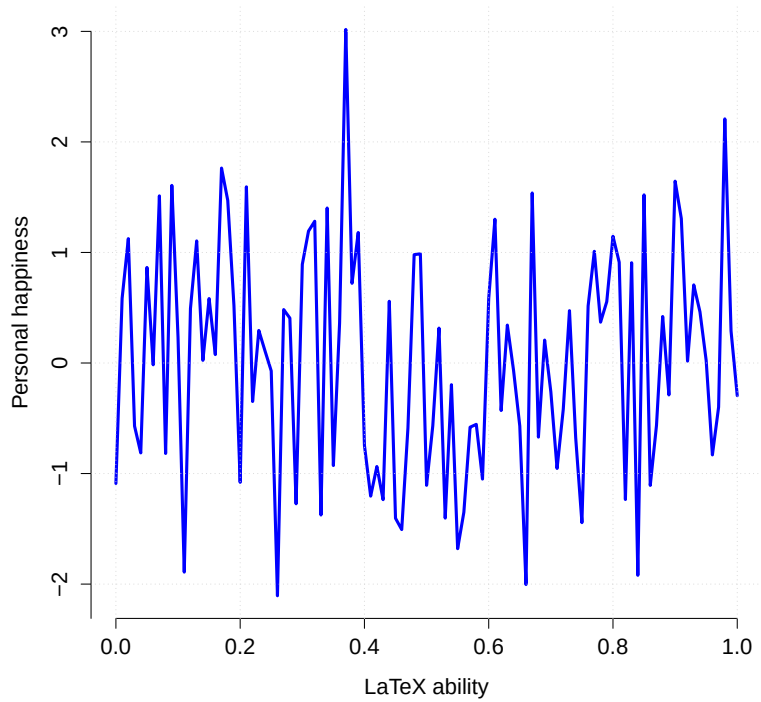
Working with LaTeX and R: Graphics

Normally you want to export R plots as PDF files, not JPEG or PNG (unless you're dealing with a very large file like a map):

R code:

```
pdf(file="FILENAME.pdf")
plot(rnorm, ylab=''Personal happiness'',
xlab=''LaTeX ability'', lwd=3,
col=''blue'')
grid(lty=''dotted'')
dev.off()
```

Plot produced by R:



Working with LaTeX and R: Verbatim

When you want to import R-Code into your LaTeX document, you have to use a special environment that allows the code to appear as is, with lines and special symbols intact. The best use for this is **verbatim**. To use, simply type:

```
\ begin{verbatim}
Paste your R-Code, then...
\ end{verbatim}
```

Exercise: Export a graph and regression table into LaTeX

1. Open the R code file we sent you
2. Change filename, then using the pdf wrapper, run the graphing code
3. Run the R code to create data
4. Run the R code that estimates models 1 and 2
5. Using one of the table commands, export a regression table for both models
6. Import the graph and the table into your LaTeX document

Where we are

Labels & references

Whenever you create a graph, table, or equation, you should label it

- *Equations*: Before you end your equation environment, include \label{eq_KEY}
- *Figures*: You must wrap your figure in a "figure" environment, then include \label{fig_KEY} before you end your figure environment
- *Tables*: You must use a "table" environment, then include \label{tab_KEY} before you end the table environment

Now, whenever you reference the equation, graph, or table in line, call the relevant label with \ref{TYPE_KEY}

- LaTeX will automatically order the objects of each type, giving you nicely numbered labels
- `Figure \ref{fig_treateffect} shows blah blah blah`
  $\Rightarrow$"Figure 7 shows blah blah blah"
- These will also sync with the captions you give to tables or graphs, which are also automatically numbered

Exercise: Label and reference your graph and table

- Make sure you use the right wrapper environments
- Use the \label command
- Give appropriate labels to both objects
- Call the objects in-line

Where we are

What is BibTeX and how does it work?

BibTeX, which is included in most TeX distributions, allows you to compile references throughout your .tex file, and to produce bibliographies automatically with pretty much any style you want.

BibTeX is a compilation procedure for your LaTeX code.

You run it in concert with the regular LaTeX compiler.

Two runs of LaTeX then BibTeX then LaTeX again...

(1) LaTeX
⇓
(2) LaTeX
⇓
(3) BibTeX
⇓
(4) LaTeX
⇓
Voila!

To use BibTex, we need to create a .bib file with our references in the correct format. Then we can "call" that file when compiling our .tex file.

The ".bib" file

```
@article{lipset1959,
  title={Some social requisites of democracy: Economic development
   and political legitimacy},
  author={Lipset, Seymour Martin},
  journal={American political science review},
  volume={53},
  number={1},
  pages={69--105},
  year={1959}
}
@book{dahl1973,
  title={Polyarchy: Participation and opposition},
  author={Dahl, Robert Alan},
  year={1973},
  publisher={Yale University Press}
}
```

Note that `lipset1959`, `dahl1973` are the (unique) keys, which we will use to cite it in our .tex file. You can directly get the formatted code for BibTeX references from Google Scholar, Zotero, etc.

Let's practice using BibTeX

Add a new article to a .bib file:

- Open "practice.bib" in your TEX editor
- Search for the article you want to include in Google scholar (or similar)
- Export the citation in format (click "cite"...)
- Copy the format citation into "practice.bib"
- Change the key to authorYEAR (or similar)

Cite your new article and make a bibliography:

- Use the "natbib" package (Mac users may need package "cite" too)
- Create an inline citation using \citet{key}
- Then a parenthetical citation with \citep{key}
- Finally, include \bibliography{practice.bib}{} and \bibliographystyle{chicago} at the bottom of your document
- Compile: LATEX → (2)LATEX → (3) BibTeX → (4)LATEX (or simply do "Quick build" in TeXMaker after you've configured it to do include BibTex compilation in Quick build going to texmaker > preferences > quick build).

Where we are

What to do when things aren't working

Sometimes LaTeX just won't compile, or compile and then delete the PDF file.

- Delete auxiliary files and compile again
- Think of what you edited since last time you successfully compiled your .tex file, and look at the code you added / edited – the error must be there.
- Try googling your problem
- Check the .tex Stack exchange forum, and post a minimum working example if you can't find anything.
- Practice! Use LaTeX to complete your problem sets for Quant I.
- Reach out to Quant I TAs for help Feel free to post LaTeX questions on Piazza, or raise them at recitation and/or office hours.