Introduction to LaTeX

Adam Kaplan
akapl@mit.edu

September 30, 2022

These slides build on materials from previous years by Emilia Simison, Dan de Kadt, Elizabeth Dekeyser, Nina McMurry, Weihuang Wong, and Guillermo Toral, 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. Figures, Tables, and Code
6. Troubleshooting
7. Alternative LaTeX Editors

Where we are

What's LaTeX?

- LaTeX is 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. 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. In other words, content entry takes place separately from formatting.
- 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:

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

$$f(y) = \frac{1}{y!\Gamma(\alpha)\beta^{\alpha}} \int_0^{\infty} \mu^{y+\alpha-1} \exp\left(-\mu\left[1+\frac{1}{\beta}\right]\right) d\mu$$

- LaTeX works well with R: you can easily input plots and tables from R, and you can even write LaTeX documents directly from R (stay tuned for the next workshop!)
- LaTeX makes it easy to manage large documents, with references and bibliography, cross-references and links inside the document, etc.
- LaTeX is used widely in the scientific community, and some people see it as a signal of credibility

Things to keep in mind about LaTeX

- LaTeX is 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.
- LaTeX is 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.
- LaTeX is not so good for other purposes, like spellchecks or review tools, although you can also do those in LaTeX.
- LaTeX is a tool – the more you have, the better.

Where we are

Text, formulas, and commands

Consider the following sentence:

> A "source file" has text, formulas (e.g. $\sqrt{5}$), and *commands to* LaTeX

Here's the source for the above sentence:

```
A ``source file'' has text, formulas (e.g. $\sqrt{5}$), and
\emph{commands to} \LaTeX
```

It's made up of:

- text: `A ``source file'' has text, formulas (e.g....`
- formula: `$\sqrt{5}$`
- more text: `), and`
- commands: `\emph{commands to} \LaTeX`

Commands & spaces

Commands:

- LaTeX commands start with \ and continue with letters or with one non-letter (e.g. `\begin{}`, `\end{}`, `\textbf{}`, `\dots`, `\&`).
- Many commands have a "starred" version, where the name of the command is followed by "*". This usually tells LaTeX to omit numbering (e.g. `\section*` means do not number 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 a subscript.
- ^ says: write what follows as a superscript.

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

Where we are

Key components of a LaTeX document

LaTeX documents typically include the following 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 (written in a `.tex` file)

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

LaTeX mechanics

Here are the steps you usually take to create 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 a bibliography in the document.
- A new `.pdf` file (and a number of auxiliary files terminating in `.aux`, `.out`, `.toc`, etc.) will appear in the same folder as your `.tex` file.
- 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 PDF reader side by side, so you can see in real time what your output looks like (this is what TexMaker does).

Let's practice!

Let's start by building a minimum article document.

- Open a `.tex` file and fill it up.
  - Give it a document class e.g. `article`. (You can also try `report`, and insert optional arguments, e.g. `letterpaper`, `a4paper`, `10pt`, `12pt`. You can insert multiple arguments, separated by a comma.)
  - Fill in the titling information.
- Add some text.
  - 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  %\usepackage[•]{•}
3
4  \begin{document}
5
6  \title{•}
7  \author{•}
8  \date{\today}
9
10 \maketitle
11
12 \section{•}
13
14 \end{document}
```

A simple .tex file

```
\documentclass[10pt]{article}
\usepackage[top=1in, bottom=1in, left=1in, right=1in]{geometry}

\begin{document}

\title{My first \LaTeX{} document}
\author{Nina McMurry}
\date{\today}

\maketitle

\section{A first section}

Some text

\section{A second section}

Some more text\footnote{...and a footnote}

\subsection{First subsection}

bla bla bla

\subsection{Second subsection}

bla bla bla

\section*{A third, un-numbered section}

Perhaps an appendix

\end{document}
```

The resulting PDF file

My first LaTeX document

Nina McMurry

September 10, 2018

1 A first section

Some text

2 A second section

Some more text[1]

2.1 First subsection

bla bla bla

2.2 Second subsection

bla bla bla

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.
- We have already used one. Do you remember which one?
- Yes! The biggest environment is the `document` environment, which bounds the entire document you make.
- Environments can be used for lists, for including figures, 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`:

```
Some countries in the Southern Cone
\begin{itemize}
\item Argentina
\item Chile
\item Brazil
\end{itemize}
```

Which produces the following:
Some countries in the Southern Cone

- Argentina
- Chile
- Brazil

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. End the line with `\\` to separate lines.
5. Type a second line:
   ```
   Y_i - \varepsilon_i &= \alpha + \beta x_i
   ```
6. End the `align*` environment.

This produces a set of aligned equations (what if you omit the `*` from `align*`?):

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

Math environments

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

- `$...$` creates in-line math, `\[...\]` creates displayed math.
- Example of in-line math with `$`: writing `$\sum_{i=1}^{3}i = 1 + 2 + 3$` will produce $\sum_{i=1}^{3} i = 1 + 2 + 3$.
- Example of displayed math with `\[`: writing `\[\sum_{i=1}^{3}i = 1 + 2 + 3\]` will produce

$$\sum_{i=1}^{3} i = 1 + 2 + 3$$

If you want display-style, 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
`a+b,a \times b,a \cdot b`	$a+b, a \times b, a \cdot b$	Arithmetic
`y_{ijk},y^{ijk}`	y_{ijk}, y^{ijk}	Subscript, superscript
`\sqrt{x},\sqrt[y]{x}`	$\sqrt{x}, \sqrt[y]{x}$	Square root, y-th root
`\frac{x}{y}`	$\frac{x}{y}$	Fractions
`\left(\frac{x}{y}\right)`	$\left(\frac{x}{y}\right)$	Delimiters
`\binom{n}{n-k}`	$\binom{n}{n-k}$	Binomial coefficients
`\alpha`	α	Greek letters
`\exp,\log,\Pr`	$\exp, \log, \Pr$	Operators
`\bar{X},\hat{\beta},\widehat{XY}`	$\bar{X}, \hat{\beta}, \widehat{XY}$	Accents
`\mathcal{R},\mathbb{E}`	$\mathcal{R}, \mathbb{E}$	Math fonts

More math symbology

LaTeX code	Output	What to use it for
`\sum_{k=0}^{n-1}a r^k`	$\sum_{k=0}^{n-1} ar^k, \sum_{k=0}^{n-1} ar^k$	Sums
`\int_0^{\infty}e^{-x}\,dx`	$\int_0^{\infty} e^{-x}\,dx, \int_0^{\infty} e^{-x}\,dx$	Integrals
`\lim_{n \to \infty}\bar{X}_n`	$\lim_{n\to\infty} \bar{X}_n, \lim_{n\to\infty} \bar{X}_n$	Limits

See `http://detexify.kirelabs.org/classify.html` if you need to look up a symbol.

And nerdiness save Mathpix!

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}`

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

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

Where we are

Adding a figure or table

To insert figures in your document, include \usepackage{graphicx} in your preamble. Then

```
\begin{figure}[hbt]
\begin{center} % To keep the figure centered
\includegraphics[scale=0.8]{figures/my_figure_filename}
\end{center}
\caption{A Pretty Plot}
\end{figure}
```

Notes on figures and tables

- You can specify the file extension, or not.
- The caption is also optional, but it's good practice to caption your figures and tables.
- Options for `\includegraphics` are e.g. `[width=\textwidth]` or `[height=3cm]`
- Labels are also really useful! `\label{tab:MyTab}` combined with `\ref{tab:MyTab}`
- Images should be in the same folder! (or `\graphicspath{{Graphs/}}`)
- A useful LaTeX table generator is `http://www.tablesgenerator.com/`

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 positioning parameter: `\begin{figure}[placement specifier]`.
- The default placement specifier, 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".
- 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`.

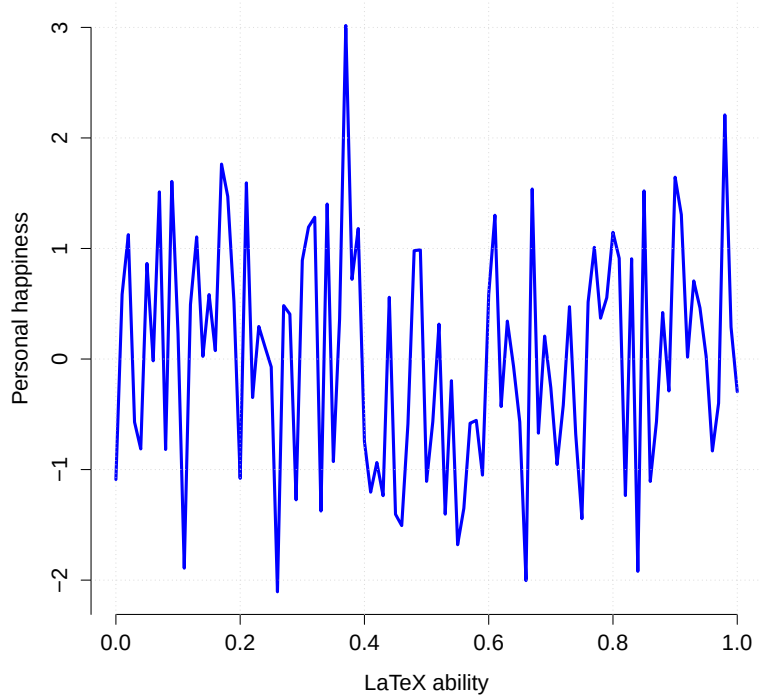
Working with LaTeX and R: Graphics

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

R code:

```
plot(rnorm, ylab=''Personal
     happiness'',
xlab=''LaTeX ability'', lwd=3,
col=''blue'')
grid(lty=''dotted'')
```

Plot produced by R:



Working with LaTeX and R: Tables

R code:

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

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

Output from LaTeX:

	Dependent variable:	
	y	
	(1)	(2)
x1	5.377***	4.922***
	(0.063)	(0.068)
x2	5.916***	0.721
	(0.318)	(0.618)
x1:x2		1.043***
		(0.115)
Constant	−1.747***	0.450
	(0.370)	(0.365)
Observations	100	100
R^2	0.988	0.993
Adjusted R^2	0.988	0.993
Note:	*p<0.1; **p<0.05; ***p<0.01	

Working with LaTeX and R: Code

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. We recommend the `listings` package. To use, include `\usepackage{listings}` in your preamble, then type

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

In your preamble, you can adjust settings, e.g.

```
\lstset{
basicstyle=\ttfamily\footnotesize, % font family/size
breaklines=true                    % sets automatic line
    breaking
}
```

Also: old-fashioned `Verbatim`

Exercise: Export a graph and regression table into LaTeX

1. Open the `R` code file we sent you
2. Run the graphing code in RStudio and export the graph as a pdf file.
3. Run the `R` code to create data
4. Run the `R` code that estimates models 1 and 2
5. Using `stargazer`, export a regression table for both models
6. Import the graph, the table, and the code into your LaTeX document

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.

Where we are

Alternatives to TeXmaker

- TeXworks (www.tug.org/texworks)
- TeXstudio (www.texstudio.org)
- Overleaf (www.overleaf.com)
- LyX (www.lyx.org)
- Emacs (www.gnu.org./software/emacs)

The End?

Thank you!

Not Really!

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.

One run of LaTeX then BibTeX then LaTeX twice...

(1) LaTeX
⇓
(2) BibTeX
⇓
(3) LaTeX
⇓
(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},
  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 I

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 BibTeX format (click "cite"...)
- Copy the BibTeX format citation into `practice.bib`
- Change the `key` to authorYEAR (or similar)

Let's practice using BibTeX II

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}` 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

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*: Wrap your figure in a `figure` environment, then include `\label{fig_KEY}` before `\end{figure}`. `\label` must be placed between `\caption` (if included) and `\end{figure}`
- *Tables*: As above.

To reference the equation, graph, or table in line, call the relevant label with `\ref{fig_KEY}`

- LaTeX will automatically order the objects of each type.
- `Figure~\ref{fig_KEY} 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
- You can also label and reference sections, subsections etc.

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

Why is this not working???!

- Did I make a typo?
- Have I forgotten a package in the preamble?
- Have I closed all the environments I have opened?
- When was the last time this document compiled properly?
- Google to the rescue!

Where we are

Internet resources that will make your life easier

- `https://www.tablesgenerator.com/`
- Mathpix in action
- Zotero export (not an internet friend, but an appreciated one)

Where we are

Getting started with Beamer

```
\documentclass{beamer}

\begin{frame}{frame title}
        Text
%\end{frame}

\pause
```

Now let's try it!

- Create a new document
- Set its class to *beamer*
- Create a slide with a title
- Include some text (Maybe an itemized list?)
- Add some pauses in between the content of your slide

Getting fancier I: Themes and customization

```
\usetheme{CambridgeUS}
```

Multiple themes available! Check:
https://deic-web.uab.cat/~iblanes/beamer_gallery/index.html

```
\setbeamertemplate{navigation symbols}{}
\setbeamercolor{item projected}{bg=red}
```

Getting fancier II: Table of contents

- Normal table of contents
- And a fancier one