Introduction to LaTeX

Lecture 1

Professor Brian C. Keegan

4 April 2023

Department of Information Science

Interdisciplinary Training in
the Social Sciences
UNIVERSITY OF COLORADO **BOULDER**

Introductions

Introductions

- Name and pronouns
- Affiliations
- Reasons you are here

About me

- Assistant Professor, Department of Information Science
- Computational social science, governing online commons, high-tempo online collaborations, cannabis informatics
- Didn't use LaTeX until *after* dissertation... a huge mistake!
- You are in version 1 of this course, feedback is very welcome!
- brian.keegan@colorado.edu

Outline of lectures

- Two lectures (April 4 & 11) for two hours
- Lecture one
    - Syntax, formatting, lists, tables, figures
- Lecture two
    - Math, document structure, bibliographies, presentations

Resources

- User Guide. *The LaTeX Project*.
- LaTeX. *Wikibooks*.
- Documentation. *Overleaf*.
- Learn LaTeX.
- Getting to Grips with LaTeX.
- LaTeX 2_ε Cheat Sheet.
- StackOverflow.

What is LaTeX?

- LaTeX is a markup language for typesetting documents
- It excels at math notation, multiple figures, & large documents
- Not a WYSIWYG word processor like Microsoft Word, Google Docs, Apple Pages, Adobe InDesign, *etc*.
- The code *describing* your document's style and its content is compiled into a file
    - Typically PDF but PostScript, RTF, HTML, SVG possible
    - Converting to Microsoft Word ".doc" or ".docx" is notoriously painful: LaTeX → PDF → Adobe Acrobat → Microsoft Word

History around LaTeX

- Typography and typesetting are art, technology, and professions originating in 10th-century China
- Early computers could not represent text beyond simple monospaced Latin characters like typewriters or teleprinters
- Donald Knuth developed "TeX" in 1978 to support revisions to his famous book *The Art of Computer Programming*
    - Math formulae, non-Latin characters, multiple fonts
- LaTeX released in 1985 by Leslie Lamport with a focus on more user-friendly syntax and design patterns than TeX

Culture around LaTeX

- Common in math, physics, computer science, economics
- Mispronunciation is a classic out-group marker
  - LAH-tekh *or* LAY-tekh; *not* LAY-teks
- **Costs**: learning and debugging technical syntax, configuring document structure and style
- **Benefits**: managing and customizing complex technical documents, professional-looking camera-ready output
  - Manuscripts, theses, lecture notes, presentations
  - These slides were written in LaTeX!
- Sharing, re-use, & adaptation of LaTeX code is common[1]

[1] Rotabi, R., Danescu-Niculescu-Mizil, C., & Kleinberg, J. (2017a). Competition and selection among conventions; (2017b). Tracing the Use of Practices Through Networks of Collaboration.

Development environment

- LaTeX can be run on your local machine
  - *Lots* of headaches around package management, fonts, *etc.*
  - **Windows** users should use MikTeX
  - **Mac** users should use MacTex
  - Then use modern text editors/IDEs like Sublime, Atom, Eclipse, XCode, *etc.* to develop documents
- We will use an online service called Overleaf
  - The "cloud" handles package management, fonts, *etc.*
  - Free version is sufficient, paid supports advanced features
  - Misses some features of IDEs, but good for $> 90\%$ of users

Customizing and debugging

- Customizing style of document through arcane commands and assembling a collection of libraries is the timesink
- Shifting mindset: Want to change the spacing for a bulleted list?
  - WYSIWYG: Highlight passage and apply desired formatting
  - LaTeX: Find variable, parameter, or operator or import a new library.
- No one knows all of these moving parts and you are *not* a "bad" user for consulting StackOverflow and documentation all the time!
  - Many people have probably had a question like this before
  - A helpful answer on StackOverflow
- Changes may do nothing, break your whole build, or do something close to what you hoped → this is debugging!

Syntax

Minimum viable example

- This is the "Hello World!" of LaTeX

Hello World!

```
\documentclass{article}
\begin{document}
Hello world!
\end{document}
```

- Also the most minimum example of a "minimum viable example" helpful for debugging the where/why/what of bugs
  - a `\documentclass` with type ("article" is a common default)
  - a "document" environment between a `\begin` and `\end`
- All the content exists within the "document" environment

Paragraphs, reserved characters, and comments

- New paragraph created with an empty line
- There are some "reserved" characters with special meanings
  - `# $ % ^ & _ { } ~ \` ...escape them with a backslash
  - A *very* common error, especially when copying from elsewhere
- Comment lines with %

Paragraphs escaping reserved characters and comments

```
\documentclass{article}
\begin{document}
The apple costs \$2 \& the orange is \$1. % Paragraph 1

Option \#2 has a 25\% chance of success. % Paragraph 2
\end{document}
```

Commands and environments

- A command is invoked with a backslash and name of the command
  - Accepts arguments within curly braces `{ }`
  - Accepts optional parameters within square brackets `[ ]`
- An environment alters the behavior of larger parts of a document
  - Exists between the `\begin` and `\end` tags
- Commands and environments can be nested

Commands and environments

```
\documentclass{article}
\begin{document}
\begin{itemize} % Starts an itemize environment for a bulleted list
    \item I am \textbf{very} excited! % The \textbf{} command bolds the word very
\end{itemize} % Stops the itemize environment
\end{document}
```

Front matter

Document environment

- Always the first command because it defines layout and style
- There are many document classes available by default

Class	Description
`article`	Scientific manuscripts and reports
`report` & `book`	Documents with multiple chapters
`letter`	Professional correspondence
`minimal`	The most basic type used for debugging

- More available at LaTeX Templates, Overleaf Gallery, *etc.*
  - Many publishers have their own templates & classes

Customizing document class

```
\documentclass[12pt,letter]{article} % Use 12-point font on a 8.5" x 11" letter paper
\begin{document}
Hello world!
\end{document}
```

Packages

- Packages change and improve how default LaTeX works
    - Downloading and configuring packages locally will quickly make you want to use IDEs or cloud services offering package management
- Import packages at top of the file, before `\begin{document}`
- The `geometry` package customizes page layout

Creating 2-inch margins

```
\documentclass{article}
\usepackage[margin=2in]{geometry} % Import the geometry package and use 2-inch margins
\begin{document}
Hello world!
\end{document}
```

Preamble: Title and Author

- You will likely also want a title and author
- Enter these into the "preamble" of the document before the content begins
  - `\title{}` formats and stores the name of the document
  - `\author{}` formats and stores the name(s) of the authors(s)

Adding a title and author

```
\documentclass{article}
\begin{document}
\title{The Document Title} % Store the name of the document
\author{A Brilliant Writer} % Store the name of the author
\maketitle % Make the title, including the author
Hello world!
\end{document}
```

Preamble: Abstract and date

- Including an abstract and date is also common

Adding an abstract and title

```
\documentclass{article}
\begin{document}
\title{The Document Title}
\author{A Brilliant Writer}
\begin{abstract}
    A quick summary % Indenting within environments is helpful
\end{abstract}
\date{} % Leaving this empty defaults to today
\maketitle
Hello world!
\end{document}
```

Formatting

Sections: Basics

- LaTeX provides a few levels of sections (top to bottom):
  1. `\chapter` (specific to `book` and `report` classes)
  2. `\section` (typically the top-level in an `article` manuscript)
  3. `\subsection`
  4. `\subsubsection`
  5. `\paragraph`
- The table of contents will reference these section names

Adding a section

```
\documentclass{article}
\begin{document}
Hello world!
\section{The next day} % Creates a new section
It's still a beautiful day! % Content within new section
\end{document}
```

Sections: Formatting sections

- Sections can formatted to have consistent styles, numbering, *etc*.
- In the front matter, import the `titlesec` package and invoke the `\titleformat{}` command whose style you want to change
- Font, size, alignment, numbering, *etc*. can be customized

```
\titleformat{ command }[ shape ]{ format }{ label }{ sep }{ before-code }[ after-code ]
```

Changing section formatting

```
\documentclass{article}
\usepackage{titlesec} % Imports the titlesec package
% Update the \section definition to remove numbers, bold and large font, no indent
\titleformat{\section}{\normalfont\Large\bfseries}{}{0pt}{}
\begin{document}
\section{Introduction}
Hello world!
\end{document}
```

Text: Styling

- Styles include: **bold**, *italics*, underline, SMALL CAPS, and `typewriter`
- Can also use $_{\text{textsubscript}}$ and $^{\text{textsuperscript}}$

Changing text style

```
\documentclass{article}
\begin{document}
\textbf{Hello world}, \textsc{what} a \textit{beautiful} \texttt{afternoon} on the
↪  26\textsuperscript{th} day of July!
\end{document}
```

Text: Size

- Text can be scaled to several pre-defined sizes
- List of font size names and examples:
  - `\tiny`, `\footnotesize`, `\large`, `\Huge`
- Different style with command inside braces: `{\size content}`

Changing text size

```
\documentclass{article}
\begin{document}
{\tiny Hello world}, {\small what} a {\Large beautiful} day!
\end{document}
```

Text: Font

- There are hundreds of typefaces available (Overleaf), including Times and Helvetica, but the default is Computer Modern
  - However, it's non-trivial to change fonts mid-document
- Fonts have three coarse categories: serif, sans serif, and monospaced
- Change all the fonts in a document by importing font as a package
  - "fbb" (Bembo) is one of my favorite serif fonts with a fascinating history from the 15th century ...and what this presentation uses!

Changing font

```
\documentclass{article}
\usepackage{fbb} % Change default font for whole document
\begin{document}
Hello world!
\end{document}
```

Text: Quote marks

- LaTeX has a very idiosyncratic syntax for formatting quote marks
  - Probably the second most common bug newbies encounter!
- Use the "grave accent" (tilde key to left of 1) for the left quote marks
- Use single or double quotes (to left of enter) for right quote marks
- Double quotes copied in from other documents may generate errors, unrecognizable characters, or just point wrong way

Quote marks

```
\documentclass{article}
\begin{document}
A "quote" in a phrase. % Error or looks bad
A ``quote'' in a phrase. % Canonical practice
``A `quote' in a phrase.'' % Quotes in quotes
\end{document}
```

Text: Diacritics and special characters

- Encoding non-English characters is a famously hard problem
- LaTeX designed on 128-character ASCII standard but can compose advanced characters with an escaped code
- Alternatively: declare the encoding with `inputenc` package
- Other special characters can still be used by escaping or commands
    - Detexify is a super-cool tool!

Diacritics

```
\documentclass{article}
\usepackage[utf8]{inputenc} % Tell LaTeX what encoding to use
\begin{document}
S\o{}ren Gonz\'{a}lez-Ag\"{u}ero % Use escaped ASCII characters to format
Søren González-Agüero % Copy characters directly in if encoding declared
\end{document}
```

Content: Paragraph alignment and indents

- `\indent` paragraphs that aren't, `\noindent` that are
- Default paragraph alignment is fully justified on both sides
- Paragraphs can be aligned with commands or environments

Alignment	Environment	Command
Left justified	`flushleft`	`\raggedright`
Right justified	`flushright`	`\raggedleft`
Center	`center`	`\centering`

Paragraph alignment

```
\documentclass{article}
\begin{document}
{\centering Centered in page}

\noindent Don't indent me
\end{document}
```

Content: Line spacing

- Non-default line spacing for aesthetic or editorial reasons?
  - In practice, the class controls line spacing and don't mess with this
- Use the `setspace` package

Changing line spacing

```
\documentclass{article}
\usepackage{setspace} % Imports the setspace package
% \singlespacing % single spacing
\onehalfspacing % one-half spacing
% \setstretch{1.25} % 1.25 spacing
\begin{document}
\section{Introduction}
Hello world!
\end{document}
```

Lists

Basics of lists

- Lists are a common kinds of environment within LaTeX
  - `itemize` for bulleted lists
  - `enumerate` for numbered lists
  - `description` for qualitative lists
- Lists can be nested up to a depth of four by default

Lists

```
\documentclass{article}
\begin{document}
\begin{itemize} % An itemized list
    \item The first bulleted item
    \begin{enumerate} % Create a enumerate list beneath the itemize
        \item An indented numeric item % Still uses \item
    \end{enumerate} % Close the environment
    \item A second bulleted item % Add more items
\end{itemize} % Close the environment
\end{document}
```

Helper packages

- Use the `enumitem` package to customize lists formats & styles
  - Spacing, margins, widths, indents, separation can all be customized
  - `enumitem`'s "noitemsep" option makes lists single-spaced
- If creating nested list environments is tiresome, try `easylist`!

List helpers

```
\documentclass{article}
\usepackage{enumitem}
\begin{document}
\begin{itemize}[noitemsep] %
    \item The first bulleted item
    \item A second bulleted item
    \item A third bulleted item
\end{itemize}
\end{document}
```

Marionette me!

- Let's make a non-trivial list together!

Tables

Basics of tables

- Tables in LaTeX are powerful and frustrating
- Defaults requires using two environments
  - `table` contains `tabular`, caption, and label
  - `tabular` contains the content and markup describing the table
- Reserved characters used extensively for tables
  - `&` column separator
  - `\\` new row
- There is a zen to creating LaTeX tables by hand, but...
  - `.to_latex()` for pandas, `stargazer` for R, `excel2latex` for Excel
  - Web generators like tablesgenerator.com and latex-tables.com

A “simple” table

label A	label B
value 1	value 2
value 3	value 4

Caption

A simple table

```
\documentclass{article}\begin{document}
\begin{table} % Open the table environment
    \begin{tabular}{c|c} % Open the tabular environment
        label A & label B \\
        value 1 & value 2  \\
        value 3 & value 4
    \end{tabular}
    \caption{Caption} % Give a caption
    \label{tab:my_label} % Create a reference variable
\end{table}
\end{document}
```

Intermediate tables

```
\begin{tabular}[pos]{table spec}
```

- The `tablespec` is where the number, justification, and size of columns in the table are defined
  - `l` for left, `c` for center, `r` for right, `p{size}` for text, `|` for lines
- A `multicolumn` exists in the base package and often used with `multirow` package
- Common helper packages include `array` (advanced formatting), `tabularx` (better text handling), `booktabs` (prettier table elements), `longtable` (multi-page tables)

Marionette me!

- Let's make a non-trivial table together!

Figures

Basics of figures

- `graphicx` provides `\includegraphics` that reads in image files
  - Vector images (PDF, EPS, SVG) > raster images (JPEG, PNG, GIF)
  - Scaling image size relative to document dimensions
- Like tables, it often makes sense for the graphics to be embedded within a `figure` environment defining alignment, caption, and label

A simple figure

```
\documentclass{article}
\usepackage{graphicx}
\begin{document}
Some text.
\begin{figure}
    \includegraphics[width=.5\textwidth]{image_file.png}
    \caption{Caption}
    \label{fig:my_label}
\end{figure}
\end{document}
```

Intermediate figures

- You may want multiple sub-plots in a single figure
  - Use the `caption` and `subcaption` packages

Side-by-side sub-figures

Side-by-side sub-figures

```
\documentclass{article}
\usepackage{graphicx,subcaption}
\begin{document}
Some text.
\begin{figure}
    \begin{subfigure}{.475\textwidth} % Subfigure width is a hair under half
        \includegraphics{image_file1.png}
        \caption{Caption for image 1}
        \label{fig:image1}
    \end{subfigure}\hfill
    \begin{subfigure}{.475\textwidth}
        \includegraphics{image_file2.png}
        \caption{Caption for image 2}
        \label{fig:image2}
    \end{subfigure}\caption{Caption for both images}
    \label{fig:both_images}
\end{figure}
\end{document}
```

Next week

Next week's topics

- **Math:** symbols and equations
- **Modularity:** Multi-file documents
- **Bibliographies:** BibTeX and citations
- **Intermediate:** variables, counters, functions
- **Presentations:** `beamer`, `tikz`, and posters
- **Office Hours:** Debugging your projects!