Introduction to LaTeX

Lecture 2

Brian C. Keegan, Ph.D.

4 April 2023

Assistant Professor, Department of Information Science

Introduction

- Two lectures (April 4 & 11) for two hours
- Lecture one
  - History, culture, syntax, formatting, lists, tables
- Lecture two
  - Figures, 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.

Review

- History, minimum viable examples, commands & environments
- Common bugs involving reserved characters & quotation marks
- Configuring the document with front matter's packages & preamble
- Formatting sections & styling text
- Working with lists & tables
- Demo Overleaf document is here

Today's agenda

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

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}
```

- 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 % hfill makes the right image right-aligned
    \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}
```

Math

Typesetting math

- A primary motivation for Knuth to develop TeX

- Arbitrarily complex equations possible inline or displayed

  - Dollar signs create the inline environment
  - The equation environment creates the display

Simple and less simple equations

```
\documentclass{article}\begin{document}
Einstein was very smart. He discovered $e=mc^2$.

It required a lot of math that looked like this:
\begin{equation}
    % https://en.wikipedia.org/wiki/Mass%E2%80%93energy_equivalence
    K_{0}-K_{1}=E\left(\frac{1}{\sqrt{1-\frac{v^{2}}{c^{2}}}}-1\right)
\end{equation}
\end{document}
```

Math symbols

- Many reserved characters can be used within a math environment
  - Symbols like: + - = ! / ( ) [ ] < > | ' : *
- Lower- and upper-case Greek letters by cased name:
  \alpha \gamma \Delta becomes $\alpha\gamma\Delta$
  - Annoyingly, \Beta does not resolve automatically, use B explicitly
- Powers/superscripts with carat, indices/subscripts with underscore
  - Enclose longer expressions within braces {}

Sub- and super-scripts

```
\documentclass{article}\begin{document}
\begin{equation}
    \alpha_{\beta+1} = \gamma^2 + \alpha_\gamma^2 - \alpha_{\beta-1}
\end{equation}
\end{document}
```

Intermediate math

- *Many* other symbols and accenting available!
  - List of mathematical symbols and Comprehensive LaTeX Symbol List
  - Technical syntax in linguistics, computer science and programming, and chemistry also supported

- Fractions (\frac), roots (\sqrt), sums (\sum), and integrals (\int), look like command with multiple parameters

  `\frac{n!}{k!(n-k)!} = \binom{n}{k}`

- Matrices are similar to tables: environments with columns delimited by & and rows delimited by newlines

Modularity

Making modular documents

- Good development practice suggests breaking code up into smaller sections rather than a monolithic document

    - Separate "child" .tex files for each chapter or section and a "parent" .tex files where ordering and preamble is managed

- Default "article" documentclass does not support chapters by default, use default documentclasses like book and report instead

- Two options for combining files into a finished documents

    - Use \include to force a page break, but it *cannot* be recursive (no \include within \include) — best for chapters
    - \input does *not* force a page break and *can* be recursive (\input within \input) — better for modularizing manuscript sections

Example of a modularized parent document

- If we had separate files for each chapter of a dissertation, we could compile them all into a completed dissertation with a MVE parent document that looks like:

```
Modularized thesis or book

\documentclass{report} % notice the change in document class
%...other front matter
\begin{document}
\maketitle \tableofcontents \listoffigures \listoftables % make all the front material
\include{Chapter_1} % "Chapter_1.tex" is a file in the same directory
\include{Chapter_2}
\include{Chapter_3}
\include{Chapter_4}
\include{Chapter_5}
\end{document}
```

Collaborative writing and revision tracking

- Overleaf provides interfaces for leaving comments and (premium) services for tracking changes and syncing with GitHub repositories

- Version control can be overkill for smaller collaborations, but nevertheless a good opportunity to practice!

- Outside of Overleaf, packages like `todonotes` can be used to leave comments that are compiled into the document

- Use `latexdiff` or `changebar` to automatically highlight changes between versions for revise and resubmit cycles

Bibliographies

BibTeX

- BibTeX is a database for storing bibliographic information
  - a `.bib` file contains entries like "article", "book", "inproceedings", *etc.*
  - Populate the `.bib` file using entries from your Zotero or Mendeley
  - Also use helper tools like BetterBibTeX to clean up output

- Each entry has a citation key to be referenced in the text, as well as the meta-data defining the reference (title, author, *etc.*)

BibTeX entry in `bibliography.bib`

```
% bibliography.bib
@book{kuhn1970structure, % reference type is "book" and citation key
title={The Structure of Scientific Revolutions},
author={Kuhn, Thomas S},
volume={111},
year={1970},
publisher={Chicago University of Chicago Press}
}
```

Citing references

- In the body of the text, cite the reference using the \cite{} command and pass the citation key

    - Also include a non-breaking space character before the cite command to prevent orphaned citations: ~\cite

- Pass multiple comma-separated keys that will be grouped together, depending on citation style

- Specific pages, chapters, tables, *etc.* can be referenced as an option

---

Citing

```
\documentclass{article}\begin{document}
Scientific knowledge does not accumulate smoothly~\cite{kuhn1970structure}.
\bibliography{bibliography}
\end{document}
```

Citation and bibliography styles

- Reference formatting standards abound, LaTeX supports several by default: apalike, plain, abbrv

- Define these styles with `\bibliographystyle` before referencing the bibliography file at (typically) the end of the document

- More advanced citation and bibliography style management can be customized with the `natbib` package

Citing

```
\documentclass{article}\begin{document}
Scientific knowledge does not accumulate smoothly~\cite{kuhn1970structure}.
\bibliographystyle{apalike} % Define the bibliography style
\bibliography{bibliography}
\end{document}
```

Presentations and figures

- You can also use LaTeX to make presentations (like this one!)

    - There is a "slides" document class, but >99% chance you'd prefer `beamer` or a popular Beamer theme like `metropolis`

- LaTeX can also make posters with the `beamerposter` and `tikzposter` templates

Figures

- `tikz` is a very powerful package for drawing diagrams and figures in LaTeX

- You will definitely want to build on others' MVEs and resources
  - Examples at Awesome TikZ, TeXample, Walmes

- Derivative packages like tikz-network

Office hours