

L^AT_EX Workshop

Nicholas Berridge-Argent

UNSW Society of Statistics

2019



Society of Statistics
UNSW

“This is not a \LaTeX course”

– *every course using \LaTeX*



Society of Statistics
UNSW

Schedule

- 1 Introduction
- 2 Document Setup and Text
- 3 Equations and Maths
- 4 Tables and Images
- 5 Graphs and Diagrams
- 6 Extra Document Features
- 7 Source Code Listings, Presentations
- 8 Where To Next?



History

- Donald Knuth was writing the second edition of his book *The Art of Computer Programming* in 1977, when he found that the typesetting used by the publishers was inferior.
- As you do, invented a markup language for typesetting called TeX, which was eventually used around the world for typesetting academic documents.
- These days, \LaTeX forms the most commonly-used extension to plain TeX. Industry-standard for academic documents in mathematics, statistics, computer science, physics, etc.



Environments

- **Overleaf** — fully online, compiles in real time. UNSW Students get free Overleaf Pro account with their UNSW e-mail.
- **TeX Live** — can be installed on your computer, so no size limits, requires no online access. **But!** takes a long time to download, takes some skill to set up packages, needs you to compile \LaTeX manually. Available for Windows and Linux, Mac users use **MacTeX**.
- **MiKTeX** — Alternative to TeX Live for Windows, Linux and Mac.
- **Microsoft Word** — Not even once... (although the Word equation editor uses TeX-like syntax under the hood)

Getting Started

```
\documentclass{article}

\begin{document}

Hello, \LaTeX{}!

\end{document}
```

Hello, \LaTeX !



Getting Started II

```
\documentclass{article}

\begin{document}

\[
\sum_{k=1}^n k^2 = \frac{n}{6}(2n+1)(n+1)
\]

\end{document}
```

$$\sum_{k=1}^n k^2 = \frac{n}{6}(2n+1)(n+1)$$



Schedule

- 1 Introduction
- 2 Document Setup and Text
- 3 Equations and Maths
- 4 Tables and Images
- 5 Graphs and Diagrams
- 6 Extra Document Features
- 7 Source Code Listings, Presentations
- 8 Where To Next?



The Preamble

- (Relatively) small section of code for setting the document class, metadata, setting up packages.
- `\documentclass` — declares the type of document and sets up a bunch of formatting, most common is `article`.
- `\usepackage` — includes useful packages, most documents have several.
- `\title`, `\author`, `\date` — used to set document metadata.
- See example `theorem.tex`.

The geometry Package

- Default paper size in \LaTeX is US Letter (215.9×279.4 mm) rather than A4 (210×297 mm).
- The margins are set to be enormous based on readability studies, but you may want to change those too.
- We can use the package geometry for this:
`\usepackage[a4paper,margin=3cm]{geometry}`
- Unrelated to the geometry package, but we can also use two columns with:
`\documentclass[twocolumn]{article}`



Metadata

- Some document metadata can be set in the preamble, then used throughout the document e.g. by the `\maketitle` command.
- `\title` — Sets document title:
`\title{I Like Statistics}`.
- `\author` — Sets document author(s):
`\author{Nicholas Berridge-Argent}\z5208292}`
- `\date` — Sets the date the document was made:
`\date{\today}`



Essential Packages

- `amsmath` and `amssymb` — **very** useful for mathematical typesetting, adds a lot of extra useful environments and symbols.
- `microtype` — Adjusts fonts to improve readability.
- `parskip` — Removes indentation at the start of paragraphs (personal preference).
- See example `theorem-improved.tex`.

Sections and Subsections

- Sometimes useful to separate a document into several sections.
- For this we have `\section`, `\subsection`, `\subsubsection`, ...
- Can be used with an asterisk: `\section*` to suppress numbering.
- Other document classes provide other forms of sectioning, e.g. `book` provides `\chapter`.
- Can be used with `hyperref` package to add links in a PDF viewer.



- We saw before we can add a small abstract with `abstract`.
- We can also define proof / theorem / lemma environments with `\newtheorem`

- Un-numbered lists, such as the one you're reading now, can be made with `itemize`.
- Numbered lists can be made with `enumerate`.
- Lists can be nested inside each other.

Styling Text

- Text can be made bold or italic with `\textbf` and `\textit`.
- \LaTeX supports three default fonts: `roman`, `sans-serif` and `typewriter`.
- Change font temporarily with `\textrm`, `\textsf`, `\texttt`.
- Change font for whole document in the preamble with `\renewcommand{\rmdefault}`, etc.
- Can also change font size: `\tiny`, `\large`, `\Huge`, etc.
- Can also use `\textsuperscript` for superscript, e.g. `1st`.
- See example `theorem-final.tex`



Schedule

- 1 Introduction
- 2 Document Setup and Text
- 3 Equations and Maths**
- 4 Tables and Images
- 5 Graphs and Diagrams
- 6 Extra Document Features
- 7 Source Code Listings, Presentations
- 8 Where To Next?



Basic Equations and Symbols

- Use `\(...\)` or `$. . . $` for inline equations.
- Use `\[. . .\]` or `$$. . . $$` for block equations.
- Most normal symbols are available just as-is: e.g. $(3x+2)^2=0$ becomes $(3x + 2)^2 = 0$.
- Greek letters are available by a backslash followed by their name, e.g. `\alpha` becomes α .
- Fractions are available using `\frac`, e.g. `\frac{\pi}{6}` becomes $\frac{\pi}{6}$.



Basic Equations and Symbols

- Sums are available with `\sum`, e.g.
`\sum_{k=1}^n 3^{-k}` becomes $\sum_{k=1}^n 3^{-k}$.
- Integrals are available with `\int`, e.g.
`\int_{-\infty}^{\infty} e^{-x^2} dx` becomes $\int_{-\infty}^{\infty} e^{-x^2} dx$.
- Special mathematical typefaces are available, e.g. `\mathcal{P}` becomes \mathcal{P} and `\mathbb{R}` becomes \mathbb{R} .
- See example `homework.tex`



Dilemma!

```
\documentclass{article}
\begin{document}
\[
  \sin(x+(\frac{\pi}{2})^n)
\]
\end{document}
```

$$\sin(x + (\frac{\pi}{2})^n)$$

Problem: The sine function is appearing in italics, and the brackets aren't sized properly to the fraction.

The Solution

```
\documentclass{article}
\begin{document}
\[
\sin\left(x+\left(\frac{\pi}{2}\right)^n\right)
\]
\end{document}
```

$$\sin\left(x + \left(\frac{\pi}{2}\right)^n\right)$$

Solution: Most functions have a built in command form, and `\left` and `\right` will automatically size brackets.

If your function doesn't have a command, you can use `\text`, e.g. `\text{cis }(\pi/2)` becomes `\text{cis }(\pi/2)`, make sure to include the space inside the `\text` command for spacing.



AMSMath Equation Environments

- `equation` provides an environment similar to a block equation, but with a unique number on the right-hand margin, useful to refer to multiple equations in text.
- `align` and `align*` (no numbering) provide block equations which can have multiple lines, aligned at a certain position, useful for showing long streams of working out.

Environments for Maths Mode

- `pmatrix` is useful for setting up vectors and matrices.
- `bmatrix` can also be used for matrices.
- `cases` is useful for piecewise functions.
- See example `homework-2.tex`

Schedule

- 1 Introduction
- 2 Document Setup and Text
- 3 Equations and Maths
- 4 Tables and Images
- 5 Graphs and Diagrams
- 6 Extra Document Features
- 7 Source Code Listings, Presentations
- 8 Where To Next?



Tables and Images

- Motto of tables in \LaTeX : Just use `booktabs`.
- Tables are an environment, `tabular`, which is set up with specifications for the columns.
- Horizontal rules can be used to separate rows.
- Vertical rules can be used to separate columns, but please don't.
- Different column types available: `l`, `c`, `r`, `p`.
- Images can be inserted using the `graphicx` package.

Figures

- It is often useful to control the positioning of images and tables, and to give them captions.
- We can use the `figure` and `table` environments for this.
- Takes an optional position specifier: `h`, `t`, `b`, `p`.
- Usually useful with the `\centering` command or `center` environment.
- Can be combined with a `\caption`.
- See example `report.tex`



Schedule

- 1 Introduction
- 2 Document Setup and Text
- 3 Equations and Maths
- 4 Tables and Images
- 5 Graphs and Diagrams**
- 6 Extra Document Features
- 7 Source Code Listings, Presentations
- 8 Where To Next?



Graphs with PGFPlots

- \LaTeX is good for typesetting paragraphs, equations. Not so good for describing graphs.
- We need an entirely new set of commands for describing graphs — PGFPlots.
- Can graph raw data (e.g. scatter plots) or mathematical functions.
- Can be combined with “PGF libraries” to plot graphs more easily.
- Can we do 3D graphs too? **Yes!**
- See example `report-graph.tex`

Graphs with PGFPlots

- Inside a `tikzpicture` environment, have an `axis` environment.
- Use `\addplot` to add a plot based on a function, or with `table` to add a scatterplot.
- Use `\addlegendentry` to add a reference to that to the legend.

Graphs with PGFPlots

- What's the best way to learn the right syntax?
- Google.
- Doesn't this take a long time?
- Yes, feel free to use `gnuplot`, RStudio, LibreOffice Calc, Microsoft Excel, etc. and `\includegraphics`.
- But, if you want your documents to look **really** nice...

Diagrams with TikZ

- \LaTeX is good for typesetting paragraphs, equations. Not so good for describing diagrams.
- We need an entirely new set of commands for describing diagrams — TikZ.
- Uses simple drawing commands to draw simple shapes.
- Can be combined with “TikZ packages” to draw complicated diagrams more easily.
- As before, you can use another software package which generates images and `\includegraphics`, but you won't get the best looking document possible.



Diagrams with TikZ

- Again use a `tikzpicture` environment.
- Use `\draw` to draw outlines, `\filldraw` to draw filled shapes.
- Mostly based on co-ordinates, but can add shapes e.g. circles.
- Use `\node` to add text.
- See example `report-diagram.tex`

Schedule

- 1 Introduction
- 2 Document Setup and Text
- 3 Equations and Maths
- 4 Tables and Images
- 5 Graphs and Diagrams
- 6 Extra Document Features**
- 7 Source Code Listings, Presentations
- 8 Where To Next?



Table of Contents

- Very long documents can be hard to navigate.
- Can also be very long to type up a table of contents by hand.
- Document is already populated with `\sections`, `\subsections`, etc.
- We can automatically generate a table of contents based on that.
- Sometimes takes 2–3 goes to compile the document.



Headers and Footers

- Some assignments specifically require you to add a header and footer.
- Other times it can just make your document look very nice.
- Using the package `fancyhdr` you can style custom headers and footers.

Section Titles

- The default look of section, subsection headings can be a bit boring.
- You might want to use alternative counters e.g. roman numerals.
- Using the package `titlesec` you can style custom section titles.

Links and References

- As mentioned before, `hyperref` can add links to a document in a PDF viewer.
- Can also make e.g. a table of contents clickable.
- Can **also** add clickable hyperlinks to web pages.
- Quickest way to add references: `\footnote`.
- Better way to add references: `thebibliography` environment.
- Automatically manages references in the text and adds a bibliography at the end.



Extra page styling

- Sometimes a page contains a wide table or graph, and could be made landscape. Enter: `pdflscape`.
- Sometimes you want to add colours to text (even in equations!). Enter: `xcolor`.
- What about extra fonts? We can add some extra fonts as packages, however, they may not support all characters. . .
- Using XeTeX (another set of TeX macros similar to \LaTeX) we can use **any** true-type font, but the drawbacks are the same.
- In general: just use the default fonts. You can use sans-serif maths in the default font, if you're one of those people.



Miscellaneous Syntax

- In the preamble, `\renewcommand` will let you define custom commands! **Very** useful to avoid repeating yourself.
- Comments can be added with `%` to explain tricky syntax.

Schedule

- 1 Introduction
- 2 Document Setup and Text
- 3 Equations and Maths
- 4 Tables and Images
- 5 Graphs and Diagrams
- 6 Extra Document Features
- 7 Source Code Listings, Presentations**
- 8 Where To Next?



Source Code Listings

- For computer scientists, or people who like code, you may want to include source code into your documents e.g. in an appendix.
- Not just as simple as using `\texttt`, you want a specific package for that, `listings`.
- Allows for basic syntax highlighting, but not with the typewriter font :(since it doesn't have a bold weight.
- For algorithms, ideally use a different package e.g. `algorithm2e`.



Presentations

- The formatting for presentations is **very** different to the formatting for articles and books.
- But, it would be convenient to be able to use the same typesetting e.g. for equations in a presentation.
- Academics often need to create a presentation to convey the same ideas they would in an article.
- We can use the package `beamer` for this (although it's really a document class).
- See example: this presentation! `presentation.tex`



Schedule

- 1 Introduction
- 2 Document Setup and Text
- 3 Equations and Maths
- 4 Tables and Images
- 5 Graphs and Diagrams
- 6 Extra Document Features
- 7 Source Code Listings, Presentations
- 8 Where To Next?



Where To Next?

- Obviously, this workshop was just a start!
- Official package documentation on CTAN:
<https://www.ctan.org/pkg>.
- More tutorials on Overleaf website:
<https://www.overleaf.com/learn> and Wikibooks:
<https://en.wikibooks.org/wiki/LaTeX>.
- \LaTeX Stack Exchange for questions & answers:
<https://tex.stackexchange.com/>.
- OEIS has a list of symbols:
https://oeis.org/wiki/List_of_LaTeX_mathematical_symbols.
- Slides and examples will be made available on the StatSoc website:
<https://statsoc.unsw.edu.au/>.

