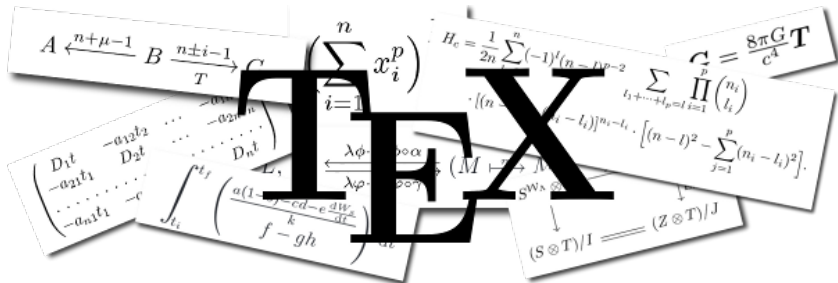


# Introduction to L<sup>A</sup>T<sub>E</sub>X

Jennifer Flegg, September 5 2018

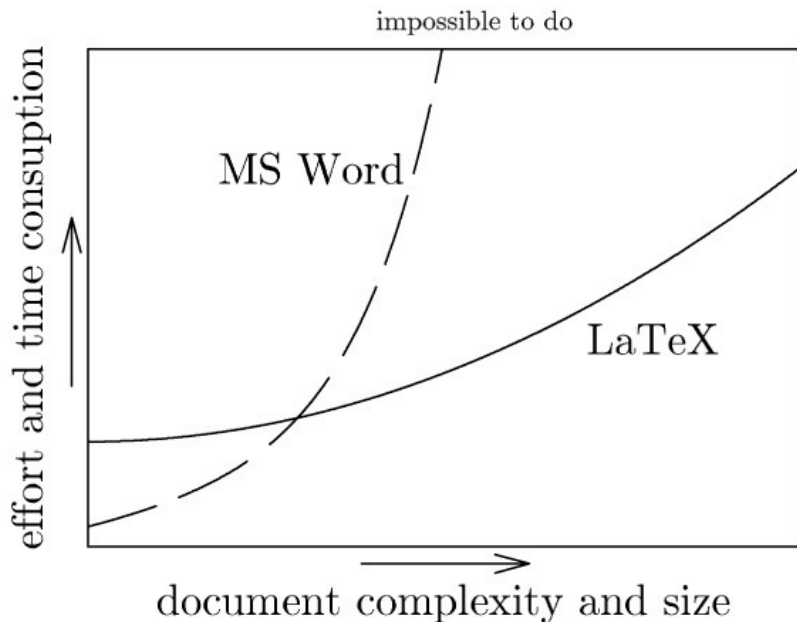


School of Mathematics and Statistics, University of Melbourne

# Why $\text{\LaTeX}$ ?

- ▶  $\text{\LaTeX}$  is the mathematical/statistical standard
- ▶  $\text{\LaTeX}$  looks better than Word (especially for equations)
- ▶  $\text{\LaTeX}$  supported across all platforms equally well
- ▶  $\text{\LaTeX}$  has many (many!) packages and is very flexible
- ▶  $\text{\LaTeX}$  works well for very large documents (stable)
- ▶  $\text{\LaTeX}$  is free
- ▶ The input is plain text. The output can be PostScript or PDF or HTML etc

## Learning curve of $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$



# Getting started in L<sup>A</sup>T<sub>E</sub>X

To start creating documents in L<sup>A</sup>T<sub>E</sub>X, you can work locally on your machine or work with an on-line editor.

1. If working locally, you will need to:
  - ▶ download MikTeX or MacTeX  
(distribution of the TeX/LaTeX typesetting system)
  - ▶ get a TeX editor (TeXworks, TeXmaker, TeXnicCenter etc)
2. If working on-line, options are:
  - ▶ ShareLaTeX
  - ▶ Overleaf

It's my understanding that these two are merging ...

Regardless of where/how you work, you'll want to get a good folder system set up with a folder for your figures, folder for references, etc.

# Basic thesis template

What your thesis looks like in terms of chapters/sections will depend on your project - take direction from your supervisor. There isn't a School template you have to use, so find your own.

There are many available here:

<https://www.overleaf.com/gallery/tagged/thesis>

The types of chapters that you will want to have **might** include:

- ▶ Title page
- ▶ Contents page
- ▶ List of figures
- ▶ List of tables
- ▶ Acknowledgements
- ▶ Abstract
- ▶ Introduction
- ▶ Literature review
- ▶ Methods/model
- ▶ Results
- ▶ Discussion
- ▶ Conclusion
- ▶ References
- ▶ Appendix

# Text writing and compiling

To write in  $\text{\LaTeX}$ , you use plain text in a TeX editor.

A key point of difference between  $\text{\LaTeX}$  and other document preparation software (e.g Word) is that with  $\text{\LaTeX}$  you do NOT work with “what you see is what you get”.

Instead you need to ‘compile’ what you have to see what your document looks like. Usually, you’ll want to compile to PDF, but there are other options.

How you compile depends on what TeX editor you use, but will generally be a ‘play’ or ‘go’ or ‘typeset’ or ‘compile’ button.

You will need to get used to compilation errors (things you have done wrong in your plain text). Debugging these error messages takes a while to get the hang of - be patient!

# Minimal document

At the absolute minimum, your document could look like this:

```
\documentclass{report}
\begin{document}
Your text goes here.
\end{document}
```

Your document begins and ends by `\begin{document}` and `\end{document}`, respectively. Stuff outside this will not appear in your final PDF document, but is part of the preamble.

The preamble is important for defining what packages you want to use (more on this later) and how you want the document to look (fontsize, margins, etc).

The `\documentclass` allows you to do different types of documents (reports, papers, beamer (presentation slides) etc).

# Packages

To get things done in  $\text{\LaTeX}$  you will need to use ‘commands’ and ‘environments’ (more on this later).

The basic  $\text{\LaTeX}$  software does not come with all commands and environments, you need to include the ones you want using `\usepackage{*}`.

For example, if you want to include fancy chemical symbols, then you need to do:

```
\usepackage{chemformula}
```

to be able to use all the commands/environments associated with this package.

The packages you will need depends on your project.

You load packages in the ‘preamble’ (after `\documentclass` but before `\begin{document}`).



# Creating thesis structure

From your  $\text{\LaTeX}$  template, you can start adding structure to the document with:

1. Chapters:

```
\chapter{Results}
```

2. Sections:

```
\section{First result}
```

3. Subsections:

```
\subsection{First  
part of first result}
```

## Chapter 1

### Results

#### 1.1 First result

##### 1.1.1 First part of first result

## Environments and commands

An environment uses `\begin{*}` and `\end{*}` to signal its start and finish:

```
\begin{itemize}
\item Apples
\item Oranges
\end{itemize}
```

- ▶ Apples
- ▶ Oranges

```
\begin{enumerate}
\item Apples
\item Oranges
\end{enumerate}
```

1. Apples
2. Oranges

A command starts with a backslash, for example:

```
\textcolor{red}{Hello there}
```

Hello there

```
\Large Hello; \small Hello
```

Hello; Hello

## Writing equations

Mathematical symbols and equations will be in equation environment or in `$ ... $` signs. Here are some simple equation examples:

<code>\sin^2(x) + (cos(x))^2 = 1</code>	$\sin^2(x) + (\cos(x))^2 = 1$
---	-------------------------------

<code>\ln{\frac{a}{b}} = \log_e</code> <code>\left( \dfrac{a}{b} \right)</code>	$\ln \frac{a}{b} = \log_e \left( \frac{a}{b} \right)$
--	---

<code>\exp(i\pi) + 1 = 0</code>	$\exp(i\pi) + 1 = 0$
<code>exp(i\pi) + 1 = 0</code>	$\exp(i\pi) + 1 = 0$
<code>exp(i\pi) \quad + 1 = 0</code>	$\exp(i\pi) + 1 = 0$

To get syntax for what you want, you can visit [CTAN website](#).

## Numbering and referring to an equation

Here is an equation with an equation number:

```
\begin{equation}\label{eq1}  
1= \int_0^{\infty} f(x) dx  
\end{equation}
```

$$1 = \int_0^{\infty} f(x) dx \quad (1)$$

Here is an equation without an equation number:

```
\begin{equation*}  
1= \int_0^{\infty} f(x) dx  
\end{equation*}
```

$$1 = \int_0^{\infty} f(x) dx$$

Here is a reference to an equation:

I'll refer to Equation (`\ref{eq1}`) to make a point.

I'll refer to Equation (??) to make a point.

## Choose how you want to present equations

Equations can appear on their own line by using double \$\$ signs:

This equation gets its own line `$$r = 1,$$` separate to other text.

This equation gets its own line

$$r = 1,$$

separate to other text.

Equations will appear inline with single \$ signs:

This equation appears inline `$r = 1$` here.

This equation appears inline `r = 1` here.

You will want to choose what is easiest for your reader to follow. In general, put important and/or large equations on their own line.

## Including graphics

You include a figure with the `\includegraphics{*}` command:

```
\includegraphics[width = 4cm]{figures/hqdefault.jpg}
```



You can also:

- ▶ Change the size of the figure with `[width = Xcm]`.
- ▶ Change centering with `center` environment.
- ▶ Include a caption.
- ▶ Refer to the figure in your main text.

## Including graphics

```
\begin{figure}\begin{center}  
\includegraphics[width = 2.5cm]{figures/hqdefault.jpg}  
\caption{Here is my explanation of this figure.}  
\label{myfig}  
\end{center}\end{figure}
```

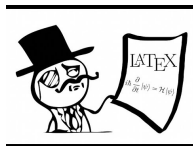


Figure 1: Here is my explanation of this figure.

In Figure `\ref{myfig}`, we can see that I am right.

In Figure ??, we can see that I am right.

## Including tables

Tables can take a lot of work to make them look the way you want.

```
\begin{table}[h]
\begin{tabular}{r r r}\hline
      & Group 1 & Group 2 \\ \hline
Label A &  $\gamma$  &  $\alpha$  \\
Label B &  $\Gamma$  &  $\beta$  \\ \hline
\end{tabular}
\caption{A summary of the table goes here.}
\label{myTable}
\end{table}
```

	Group 1	Group 2
Label A	$\gamma$	$\alpha$
Label B	$\Gamma$	$\beta$

**Table 1:** A summary of the table goes here.



## Including tables

We can refer to the table just as we did for equations and figures, using the `\ref{*}` command:

```
I want to refer to results in Table \ref{myTable}.
```

```
I want to refer to results in Table ??.
```

Change the look of the table using `\hline` and centering:

```
\begin{table}\begin{tabular}{|c|c|c|}\hline
& Group 1 & Group 2 & \\ \hline
Label A &  $\gamma$  &  $\alpha$  & \\ \hline
Label B &  $\Gamma$  &  $\beta$  & \\ \hline
\end{tabular} \end{table}
```

	Group 1	Group 2
Label A	$\gamma$	$\alpha$
Label B	$\Gamma$	$\beta$

## References lists

To reference papers, create a `.bib` file. Lots of packages out there to manage references - I like Jabref.

Once you have your `.bib` file you will need to:

1. choose a bibliography style, here I use a plain style:

```
\bibliographystyle{plain}
```

2. choose your `.bib` file, mine is called 'MyBib.bib':

```
\bibliography{MyBib}
```

3. Start citing papers, using the `\cite{*}` command.

The way to refer to a paper is detailed in  
`\cite{Blogs1998}`.

The way to refer to a paper is detailed in `[?]`.

## Example reference list

NOTE: To get the reference list you need to compile to 'BibTeX'. In fact, you'll need to do this a couple of times to get the files to update. Then compile to PDF (or other format). So, I generally follow these steps to update the reference list:

1. compile BibTeX
2. compile BibTeX
3. compile to PDF

You only need to compile BibTeX if you have changed the .bib file. If not, just go straight to compiling to PDF.

## A few places to get help

- ▶ The Comprehensive TeX Archive Network  
<https://ctan.org/?lang=en>
- ▶ The Not So Short Introduction to  $\LaTeX$   
[tug.ctan.org/info/lshort/english/lshort.pdf](http://tug.ctan.org/info/lshort/english/lshort.pdf)
- ▶ Detexify  
<http://detexify.kirelabs.org/classify.html>
- ▶ Sharelatex tutorial  
[https://www.sharelatex.com/learn/Learn\\_LaTeX\\_in\\_30\\_minutes](https://www.sharelatex.com/learn/Learn_LaTeX_in_30_minutes)
- ▶ School website  
<https://ms.unimelb.edu.au/study/current-masters-and-pgdip-students/latex>