

Writing with LaTeX using Overleaf and R Markdown

APSA Workshop
Alejandra López Villegas
September 18, 2024

Today we'll...

1. Understand what LaTeX is
2. Understand how to use LaTeX online editors like Overleaf
3. Understand bibliography management with Overleaf
4. Understand how to incorporate LaTeX and R Markdown

What is LaTeX?

- LaTeX is a typesetting system that produces technical and scientific documentation. Standard to communicate and publish scientific documents.
- Also useful to write math, make presentations, create your CV, make posters, etc.
- There are many editors that call command-line tools:
- For Windows:
 - MiKTeX/proTeXt
 - TeXLive
- For Mac:
 - MacTeX
 - XeTeX
- Or online editors: Overleaf!

What is Overleaf?

- An online LaTeX editor - makes your life easier because it doesn't require you to download any software.
- Also awesome for collaborative work.
- <https://overleaf.com>



What does a LaTeX document look like?

- Two main parts: preamble and body
- Preamble:
 - Sets up packages, define properties of document (author, title, margins, font, font size, spacing, header, page number style, etc.)
 - It contains everything between the type of document (`\documentclass[]{}{}`) and where the document begins (`\begin{document}`).
- Body: where all of the text and code that will be compiled.
 - Begins where the document begins (`\begin{document}`) and ends when your document is done (`\end{document}`)

```
\documentclass[12pt]{article} %there are different document class types
\usepackage{amsmath, amsmath, amssymb, bm} %Math fonts and symbols
\usepackage{dcolumn, multirow} % decimal-aligned columns, multi-row cells
\usepackage[colorlinks=true]{hyperref}
\usepackage{graphicx, subfigure, float} % graphics commands
\usepackage[margin=1in]{geometry} % sets page layout
\usepackage{setspace}% allows toggling of double/single-spacing
\usepackage{verbatim}% defines environment for un-evaluated code
\usepackage{natbib}% defines citation commands and environments.
\singlespace % set document spacing to single
\bibpunct[ , ]{()}{,}{a}{}{,} % sets the punctuation of the bibliography entires.
\newcolumntype{d}[1]{D{.}{.}{#1}} % defines a decimal-aligned column

\title{Introduction to \LaTeX{}} % set the title of the document
```

Some tips for your documents...

- `\\` and `\newline` - both move whatever follows them to a new line. However, if there is “no line to end”, then you will get complaints about this.
- `\newpage` will put everything below starting on a new page.
- ‘‘ is the open double-quote symbol and ’’ is the closed double-quote symbol.
- `\section`, `\subsection` and `\subsubsection` - all generate sequentially nested numbered sections.

```
\section{Section 1 }  
\subsection{Subsection 1}  
\subsubsection{Subsubsection 1}
```

will produce:

1 Section 1

1.1 Subsection 1

1.1.1 Subsubsection 1

- `\footnote{}` puts a footnote with the text between the curly braces at the bottom of the page. You can use the package `endnotes.sty` to convert from footnotes to endnotes.

Some tips for your documents... lists and bullet points

```
\begin{itemize}
```

```
\item First bullet point.
```

```
\end{itemize}
```

- First bullet point.

```
\begin{enumerate}
```

```
\item First bullet point.
```

```
\end{enumerate}
```

1. First bullet point.

```
\begin{enumerate}
```

```
\item First item
```

```
\begin{enumerate}
```

```
\item First sub-item
```

```
\begin{enumerate}
```

```
\item First sub-sub-item
```

```
\begin{enumerate}
```

```
\item First sub-sub-sub-item.
```

```
\end{enumerate}
```

```
\end{enumerate}
```

```
\end{enumerate}
```

```
\end{enumerate}
```

1. First item

- (a) First sub-item

- i. First sub-sub-item

- A. First sub-sub-sub-item.

Some tips for your documents... spacing

- If you install the setspace package in the preamble of your document, you'll be able to alter the spacing of your paragraphs:
 - Can either use `\doublespace` `\singlespace`
 - Or `\setstretch{desiredspacing}`

`\doublespace`

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper,

`\singlespace`

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper,

Tables

- Tables have their own environment.
 - The ampersand (&) separates the content of each cell, while lines end with \\

```
\begin{tabular}{lcc}
```

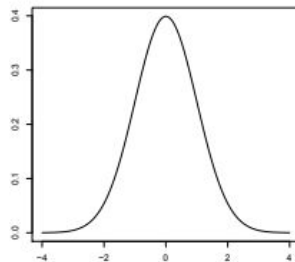
- Once you learn more R, you'll realize that some packages will help you create papers for your own research (such as stargazer and modelsummary).
 - R will automatically produce tables in LaTeX language that can be easily copy pasted into overleaf/the machine of your choosing.
- Nice guides on how to modify things in your tables if using modelsummary/stargazer:
 - <https://modelsummary.com/articles/modelsummary.html>
 - <https://cran.r-project.org/web/packages/stargazer/stargazer.pdf>

Figures

- The graphicx package allows for the inclusion of figures:

```
\begin{figure}[H]
\caption{Normal Density over  $[-4,4]$ ,  $\mu=0$ ,  $\sigma=1$  with more
words to make the caption really long}\label{fig:onefig}
\centerline{\includegraphics[width=2.5in]{norm1.pdf}}
\end{figure}
```

Figure 1: Normal Density over $[-4,4]$, $\mu=0$, $\sigma=1$ with more words to make the caption really long



Equations

- One of the main advantages of LaTeX is that it is very useful to write equations. Here is a list of commonly used symbols:

https://oeis.org/wiki/List_of_LaTeX_mathematical_symbols

- For example:

```
\begin{equation}
y_{i} = b_{0} + b_{1}x_{i} + \varepsilon_{i}\label{eq:linreg}
\end{equation}
```

$$y_i = b_0 + b_1x_i + \varepsilon_i$$

```
\begin{equation}
\bar{X} = \left(\frac{1}{n}\right)\sum X
\end{equation}
```

$$\bar{X} = \left(\frac{1}{n}\right)\sum X$$

```
\begin{align}
x + 11 &= 3 \\
x &= 3-11 \\
x &= -8
\end{align}
```

will produce

$$\begin{aligned}x + 11 &= 3 \\x &= 3 - 11 \\x &= -8\end{aligned}$$

BibTeX and citations

- To use BibTeX, you will need to create a .bib file to store your references and then tell LaTeX how you want those citations to appear on your document. You will also need to use a bib package such as natbib.
- The .bib file is a flat text file that has a series of lines for each entry, and you can refer to them in your main document like this:

- `\citet{MG2004}` produces *Mittelbach and Goosens (2004)*
- `\citep{MG2004}` produces *(Mittelbach and Goosens 2004)*
- `\citep[See] [251] {MG2004}` produces *(See Mittelbach and Goosens 2004, 251)*
- `\citeyearp{MG2004}` produces *(2004)*

Tips for your bibliography (.bib) file

- You can use your citation manager such as Zotero to create your .bib file to export all of your references to BibTeX - here is a [good guide](#) for this.
- If you refuse to use a citation manager, Google scholar usually has a BibTeX version of your desired reference that you can copy and paste to create your own .bib file manually.

R Markdown

- R Markdown allows you to compile a single R Markdown document into different formats such as PDF, HTML or Word.
- Allows you to create notebooks in which you can run code chunks interactively
- Create slides, write journal articles, generate websites, etc.

R Markdown documents

- Made up of two parts:
 - Metadata: syntax is in YAML (<https://en.wikipedia.org/wiki/YAML>). Indentation in this section matters a lot!

```
---  
title: "Hello R Markdown"  
author: "Awesome Me"  
date: "2018-02-14"  
output: html_document  
---
```

- Body
 - Text
 - Code chunks:

```
```{r}  
fit = lm(dist ~ speed, data = cars)
b = coef(fit)
plot(cars)
abline(fit)
```
```

Tips for your R Markdown documents

- Whatever is not inside the metadata or the code chunks is text, you don't need to use a hashtag to indicate it's not code.
- Because it's text, you may want to do specific things with it - this cheat sheet is super useful

(<https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf>)

- Use # or ## or ### to create titles and subtitles
- Use $$$$ for equations
- Use to include figures/images
- Use *** for a break
- Use * and + to create bullet points and sub bullet points.
- You can reference your .bib file by specifying it in your metadata.

Tips for your R Markdown chunks of code

- Use ````{r}` to open a chunk of code and ````` to close them.
- You can include things inside of the curly brackets after the `r` to specify how you want the output of your chunk to look like:



| option | default | effect |
|-------------------------|----------|---|
| <code>eval</code> | TRUE | Whether to evaluate the code and include its results |
| <code>echo</code> | TRUE | Whether to display code along with its results |
| <code>warning</code> | TRUE | Whether to display warnings |
| <code>error</code> | FALSE | Whether to display errors |
| <code>message</code> | TRUE | Whether to display messages |
| <code>tidy</code> | FALSE | Whether to reformat code in a tidy way when displaying it |
| <code>results</code> | "markup" | "markup", "asis", "hold", or "hide" |
| <code>cache</code> | FALSE | Whether to cache results for future renders |
| <code>comment</code> | "##" | Comment character to preface results with |
| <code>fig.width</code> | 7 | Width in inches for plots created in chunk |
| <code>fig.height</code> | 7 | Height in inches for plots created in chunk |

Rendering your R Markdown document

- Click the knit button in R Markdown at the top of the RStudio scripts panel and select the type of document you want (HTML, PDF or Word).
- Doing this will embed code chunk and insert the results into your document, build a new version of your document in the new file type, open up a preview, and save the output file in your working directory.