

# Computational Bootcamp 5: LaTeX, Markdown, and Formatting Documents for Social Science

Henry Watson

Georgetown University

8/19/22

# Installation: LaTeX

- Installing LaTeX can take a long time

# Installation: LaTeX

- Installing LaTeX can take a long time
- You do not need to install LaTeX to use Overleaf (although you will need to make an account on [www.overleaf.com](https://www.overleaf.com))

# Installation: LaTeX

- Installing LaTeX can take a long time
- You do not need to install LaTeX to use Overleaf (although you will need to make an account on [www.overleaf.com](https://www.overleaf.com))
- My preferred way to install LaTeX, if you want it primarily for RMarkdown, is to install it through R (TinyTex package)

# What is LaTeX?

- In academia, we often want to use complicated symbols and formatting in our documents like:

$$\begin{cases} \frac{\partial}{\partial b_0} \sum_{i=1}^n \{y_i - (b_0 + b_1 x_{1i} + b_2 x_{2i} + \cdots + b_{10} x_{10i})\}^2 = 0 \\ \frac{\partial}{\partial b_1} \sum_{i=1}^n \{y_i - (b_0 + b_1 x_{1i} + b_2 x_{2i} + \cdots + b_{10} x_{10i})\}^2 = 0 \\ \vdots \\ \frac{\partial}{\partial b_{10}} \sum_{i=1}^n \{y_i - (b_0 + b_1 x_{1i} + b_2 x_{2i} + \cdots + b_{10} x_{10i})\}^2 = 0 \end{cases}$$

- This is something normal text editors like word and google docs don't do well and aren't really meant for.
- LaTeX is a tool that provides an array of complicated, professional looking, and useful formatting options at the expense of ease of use.

# What is LaTeX? Continued

```

58 \begin{frame}
59 \frametitle{What is LaTeX? }
60 \begin{tsize}
61 \item In academia, we often want to use complicated symbols and formatting in our documents like:
62 \begin{equation}
63 \begin{cases}
64 \frac{\partial}{\partial b_0} \sum_{i=1}^n \{y_i - (b_0 + b_1 x_{1i} + b_2 x_{2i} + \dots + b_{10} x_{10i})\}^2 = 0 \\
65 \frac{\partial}{\partial b_1} \sum_{i=1}^n \{y_i - (b_0 + b_1 x_{1i} + b_2 x_{2i} + \dots + b_{10} x_{10i})\}^2 = 0 \\
66 \vdots \\
67 \frac{\partial}{\partial b_{10}} \sum_{i=1}^n \{y_i - (b_0 + b_1 x_{1i} + b_2 x_{2i} + \dots + b_{10} x_{10i})\}^2 = 0 \end{cases}
68 \end{cases}
69 \end{equation}
70 \item This is something normal text editors like word and google docs don't do well and aren't really meant for.
71 \item LaTeX is a tool that provides an array of complicated, professional looking, and useful formatting options
    at the expense of ease of use.
72 \end{tsize}
73 \end{frame}
74
75 \begin{frame}
76 \frametitle{What is LaTeX? Continued}
77 \begin{figure}
78 \centering
79 \includegraphics[width=12cm]{LaTeX.png}
80 \end{figure}
81 \begin{tsize}
82 \item Basically, LaTeX turns document preparation into its own form of coding, this coding is less restrictive than R
    (E.G. Copy-pasted text will still show up as unformatted text without breaking anything), but still tricky to learn.
83 \end{tsize}
84 \end{frame}
85
86 \begin{frame}
87 \frametitle{What is LaTeX? Continued}

```

## What is LaTeX?

- In academia, we often want to use complicated symbols and formatting in our documents like:

$$\begin{cases} \frac{\partial}{\partial b_0} \sum_{i=1}^n \{y_i - (b_0 + b_1 x_{1i} + b_2 x_{2i} + \dots + b_{10} x_{10i})\}^2 = 0 \\ \frac{\partial}{\partial b_1} \sum_{i=1}^n \{y_i - (b_0 + b_1 x_{1i} + b_2 x_{2i} + \dots + b_{10} x_{10i})\}^2 = 0 \\ \vdots \\ \frac{\partial}{\partial b_{10}} \sum_{i=1}^n \{y_i - (b_0 + b_1 x_{1i} + b_2 x_{2i} + \dots + b_{10} x_{10i})\}^2 = 0 \end{cases}$$

- This is something normal text editors like word and google docs don't do well and aren't really meant for.
- LaTeX is a tool that provides an array of complicated, professional looking, and useful formatting options at the expense of ease of use.

- Basically, LaTeX turns document preparation into its own form of coding, this coding is less restrictive than R (E.G. Copy-pasted text will still show up as `unformatted` text without breaking anything), but still tricky to learn.

# Text Editor Design and Usefulness

- In general, there are two paradigms of text editing, WYSIWYG (what you see is what you get, often referred to as wizywig), and WYSIWYM (what you see if what you mean, often referred to as wizywim).

# Text Editor Design and Usefulness

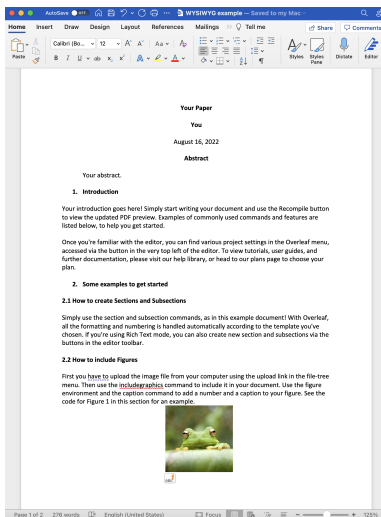
- In general, there are two paradigms of text editing, WYSIWYG (what you see is what you get, often referred to as wizywig), and WYSIWYM (what you see if what you mean, often referred to as wizywim).
- In WYSIWYG editors, like Word, Google Docs, and PowerPoint, you write and format content at the same time.



# Text Editor Design and Usefulness

- In general, there are two paradigms of text editing, WYSIWYG (what you see is what you get, often referred to as wizywig), and WYSIWYM (what you see if what you mean, often referred to as wizywim).
- In WYSIWYG editors, like Word, Google Docs, and PowerPoint, you write and format content at the same time.
- In WYSIWYM editors, formatting is done in a separate step (compiling, knitting, etc) with the aid of style commands and a stylesheet.

# WYSIWYG



# WYSIWYM

```
1 \documentclass{article}
2
3 % Language setting
4 % Replace 'english' with e.g. 'spanish' to change the document language
5 \usepackage[english]{babel}
6
7 % Set page size and margins
8 % Replace 'letterpaper' with 'a4paper' for UK/EU standard size
9 \usepackage[letterpaper,top=2cm,bottom=2cm,left=3cm,right=3cm,marginparwidth=1.75cm]{geometry}
10
11 % Useful packages
12 \usepackage{amsmath}
13 \usepackage{graphicx}
14 \usepackage{colorlinka=true, allcolors=blue}{hyperref}
15
16 \title{Your Paper}
17 \author{You}
18
19 \begin{document}
20 \maketitle
21
22 \begin{abstract}
23 Your abstract.
24 \end{abstract}
25
26 \section{Introduction}
27
28 Your introduction goes here! Simply start writing your document and use the Recompile button to view the
29 updated PDF preview. Examples of commonly used commands and features are listed below, to help you get
30 started.
31
32 Once you're familiar with the editor, you can find various project settings in the Overleaf menu, accessed via
33 the button in the very top left of the editor. To view tutorials, user guides, and further documentation,
34 please visit our \href{https://www.overleaf.com/learn}{help library}, or head to our plans page to
35 \href{https://www.overleaf.com/user/subscription/plans}{choose your plan}.
36
37 \section{Some examples to get started}
38
39 \subsection{How to create Sections and Subsections}
40
41 Simply use the section and subsection commands, as in this example document! With Overleaf, all the formatting
42 and numbering is handled automatically according to the template you've chosen. If you're using Rich Text
43 mode, you can also create new section and subsections via the buttons in the editor toolbar.
44
45 \section{How to include Figures}
46
47 First you have to upload the image file from your computer using the upload link in the file-tree menu. Then
48 use the \includegraphics command to include it in your document. Use the figure environment and the caption
49 command to add a number and a caption to your figure. See the code for Figure \ref{fig:frog} in this section
50 for an example.
51
52 Note that your figure will automatically be placed in the most appropriate place for it, given the surrounding
53 text and taking into account other figures or tables that may be close by. You can find out more about adding
54 images to your documents in this help article on \href{https://www.overleaf.com/learn/how-to/Including_images_on_Overleaf}{including images on Overleaf}.
```

## Your Paper

You  
August 16, 2022

### Abstract

Your abstract.

## 1 Introduction

Your introduction goes here! Simply start writing your document and use the Recompile button to view the updated PDF preview. Examples of commonly used commands and features are listed below, to help you get started.

Once you're familiar with the editor, you can find various project settings in the Overleaf menu, accessed via the button in the very top left of the editor. To view tutorials, user guides, and further documentation, please visit our [help library](https://www.overleaf.com/learn), or head to our plans page to [choose your plan](https://www.overleaf.com/user/subscription/plans).

## 2 Some examples to get started

### 2.1 How to create Sections and Subsections

Simply use the section and subsection commands, as in this example document! With Overleaf, all the formatting and numbering is handled automatically according to the template you've chosen. If you're using Rich Text mode, you can also create new section and subsections via the buttons in the editor toolbar.

### 2.2 How to include Figures

First you have to upload the image file from your computer using the upload link in the file-tree menu. Then use the `\includegraphics` command to include it in your document. Use the figure environment and the caption command to add a number and a caption to your figure. See the code for Figure 1 in this section for an example.

Note that your figure will automatically be placed in the most appropriate place for it, given the surrounding text and taking into account other figures or tables that may be close by. You can find out more about adding images to your documents in this help article on [including images on Overleaf](https://www.overleaf.com/learn/how-to/Including_images_on_Overleaf).



Figure 1: This frog was uploaded via the file-tree menu.

## Quick Aside, Sending Word Files

- Whether you choose to use LaTeX, Markdown, or neither, the goal should be to send .pdf files for final documents unless you are strictly required to send a word document.

## Quick Aside, Sending Word Files

- Whether you choose to use LaTeX, Markdown, or neither, the goal should be to send .pdf files for final documents unless you are strictly required to send a word document.
- This is important for two reasons:

## Quick Aside, Sending Word Files

- Whether you choose to use LaTeX, Markdown, or neither, the goal should be to send .pdf files for final documents unless you are strictly required to send a word document.
- This is important for two reasons:
  - ① .doc/.docx/Google Docs type files save meta-data (E.G: edit history) that can be embarrassing.

## Quick Aside, Sending Word Files

- Whether you choose to use LaTeX, Markdown, or neither, the goal should be to send .pdf files for final documents unless you are strictly required to send a word document.
- This is important for two reasons:
  - ① .doc/.docx/Google Docs type files save meta-data (E.G: edit history) that can be embarrassing.
  - ② PDFs, particularly PDFs generated by Markdown and Latex, but also PDFs exported by Word, have a cleaner more professional look that will improve your confidence in submission and signal to whoever is reading your work that you know what you're doing.

## Quick Aside, Sending Word Files

- Whether you choose to use LaTeX, Markdown, or neither, the goal should be to send .pdf files for final documents unless you are strictly required to send a word document.
- This is important for two reasons:
  - ① .doc/.docx/Google Docs type files save meta-data (E.G: edit history) that can be embarrassing.
  - ② PDFs, particularly PDFs generated by Markdown and Latex, but also PDFs exported by Word, have a cleaner more professional look that will improve your confidence in submission and signal to whoever is reading your work that you know what you're doing.
- Share document files when you are collaborating with someone, share pdfs when they are evaluating your work. Where the line between the two is unclear, ask.



# Markdown and RMarkdown

- Markdown trades some of LaTeX's range of use for significant improvements in ease of use.

# Markdown and RMarkdown

- Markdown trades some of LaTeX's range of use for significant improvements in ease of use.
- RMarkdown combines Markdown with an `r` script editor that is almost as clear as the normal R script editor.

# Installation: LaTeX (for R Markdown)

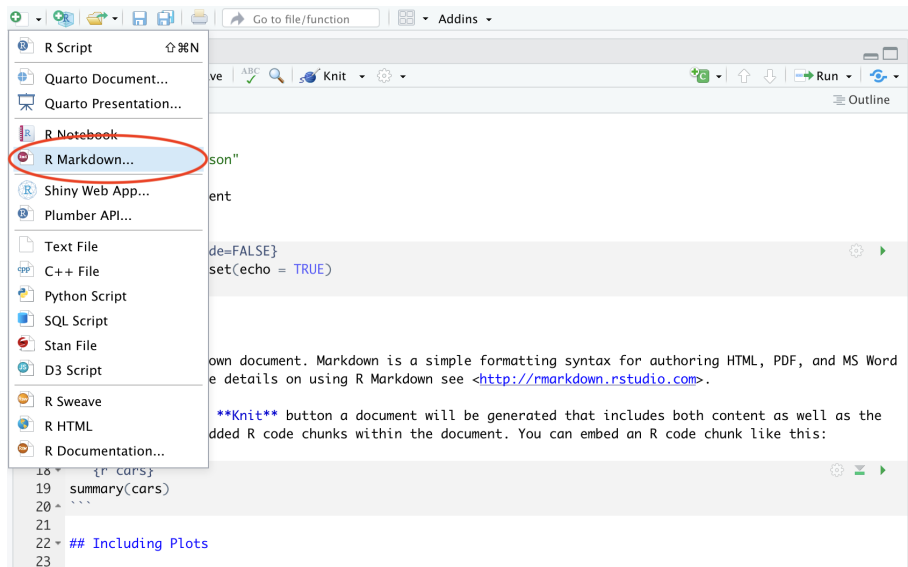
```
install.packages("rmarkdown")
```

```
library(rmarkdown)
```

```
install.packages("tinytex")
```

```
tinytex::install_tinytex()
```

# RMarkdown



# RMarkdown

Knit: Produce a final PDF document

```
9 knitr::opts_chunk$set(echo = TRUE)
10 ```
11
12 ## R Markdown
13
14 This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word
15 documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.
16
17 When you click the **Knit** button a document will be generated that includes both content as well as the
18 output of any embedded R code chunks within the document. You can embed an R code chunk like this:
19
20 ```{r cars}
21 summary(cars)
22 ```
23
24 ## Including Plots
25
26 You can also embed plots, for example:
27
28 ```{r pressure, echo=FALSE}
29 plot(pressure)
30 ```
31
32 Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that
33 generated the plot.
```

These lines (grey background) run as code

These lines (white background) run as text

# For anyone who uses Pages

For a 3x3 matrix  $A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$

- Multiply a by the determinant of the 2x2 matrix that is not in a's row or column
- Likewise for b and for c

$$|A| = a \begin{vmatrix} e & f \\ h & i \end{vmatrix} - b \begin{vmatrix} d & f \\ g & i \end{vmatrix} + c \begin{vmatrix} d & e \\ g & h \end{vmatrix}$$

Edit Equation

```
|A| = a
\begin{vmatrix}
e & f \\
h & i
\end{vmatrix}
-b
\begin{vmatrix}
d & f \\
g & i
\end{vmatrix}
+c
\begin{vmatrix}
d & e \\
g & h
\end{vmatrix}
```

# Problem Sets for GOVT 701

- Because RMarkdown allows you to include code in your writeup, with “`{r include = TRUE}`” it is great for projects where we are evaluating your coding skills like 701 problem sets.

# Problem Sets for GOVT 701

- Because RMarkdown allows you to include code in your writeup, with `"{r include = TRUE}"` it is great for projects where we are evaluating your coding skills like 701 problem sets.
- For those of you who are going on to 701 with some coding background. Writing problem sets in RMarkdown is a great way to make the class a little more challenging while preparing yourself for academic paper writing.



# Problem Sets for GOVT 701

- Because RMarkdown allows you to include code in your writeup, with `"{r include = TRUE}"` it is great for projects where we are evaluating your coding skills like 701 problem sets.
- For those of you who are going on to 701 with some coding background. Writing problem sets in RMarkdown is a great way to make the class a little more challenging while preparing yourself for academic paper writing.
- Don't kill yourself trying to do this if you aren't there! Submitting a word document and r script for problem sets is also fine.