Introduction to LaTeX

Nick Bednar

Vanderbilt Math Camp 2020

Contents

1 What Is LaTeX? 2

2 Choosing a LaTeX Editor 4

2.1 Overleaf 4

2.2 TeXworks 5

2.3 Other Editors 5

3 Your First LaTeX Document 6

3.1 The Basic Set-Up 6

3.2 Adding a Title 6

3.3 Packages 7

3.4 Section Headers and Tables of Contents 8

3.5 Typesetting Commands 8

3.6 Tables 9

3.7 Images and Figures 12

3.8 Equations 14

4 Overleaf Basics 16

4.1 Creating a New Project 16

4.2 Uploading Images 18

4.3 Compiling and Downloading a .PDF 18

5 I Have Broken LaTeX. Now What? 19

1 What Is LaTeX?

LaTeX (pronounced: *LAW-tech* or *LAY-tech*)[1] is a "document preparation system" that uses a descriptive markup language to produce stylized .PDF documents.

"What does *that* mean?"

Google Docs and Microsoft Word are What-You-See-Is-What-You-Get word processors. The words that you type on the page resemble the printed version of the document. They have nice, neat toolbars that let you change font sizes, add pictures, and create tables. In contrast, working in LaTeX feels a lot more like coding. LaTeX does not have toolbars. If you want to change the font size or add pictures, you type a command into the editor. You will not see the change in font size or the picture in the editor itself. Once you compile the document, these stylistic changes will appear in the .PDF document.

"Wait a second, Nick. First, you told me that I would have to learn math and coding to study *politics*. Now you're telling me that I have to abandon Microsoft Word?"

Yes, but it's OK! Let me explain the benefits of LaTeX over Google Docs or Microsoft Word.[2]

First, LaTeX provides you precise control over the way that the document compiles. For example, the commands allow you to control the size and placement of an image without trying to drag-and-drop it exactly where you want it.

Second, LaTeX has better functions for including mathematical equations in

[1] Not *LEI-tex*.

[2] Many people still use Microsoft Word or a similar word processor and reserve LaTeX for writing completed projects. I personally use LaTeX for most writing.

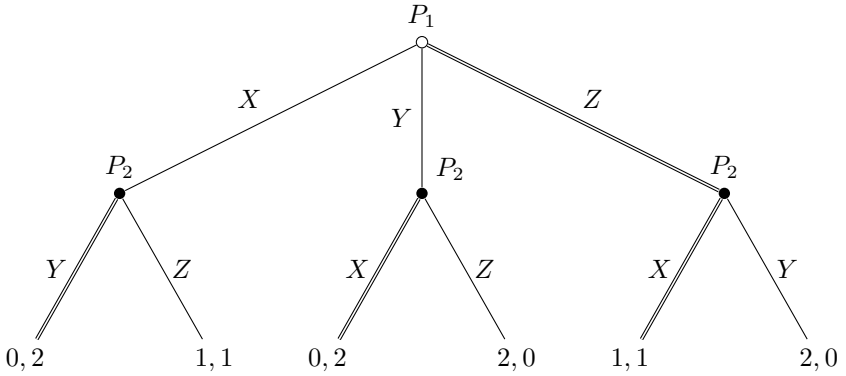
documents. Look at this beauty:

$$Z = \frac{\hat{\theta} - \theta_0}{\sigma_{\hat{\theta}}}$$

Microsoft Word does have an equation editor but it is much clunkier than using LaTeX.

Third, LaTeX allows you to draw images using the TikZ package. This is useful if you will be developing formal models or drawing directed acyclic graphs ("DAGs"). What are formal models? What are DAGs? Fret not! You will learn in time. Just know that you can use LaTeX to draw something like Figure 1.

Figure 1: Example of an Extensive Form Game



Fourth, R has packages that will produce LaTeX-ready code that you can simply copy and paste into your document. Notably, the Stargazer package for R will create beautiful regression tables without much effort.

Fifth and finally, you do not have a choice! Most of your classes will require you to submit final papers and problem sets in LaTeX-produced documents. Many journals have their own LaTeX templates and require scholars to submit their articles using that template. Learn to use LaTeX now.

The remainder of this document walks through LaTeX functionality. You do not need to grasp all of these concepts right now. Rather, this document should serve as a resource when you forget how to do something. This document covers the most basic functions in LaTeX. There are many things that I will not cover. Overleaf has great documentation on LaTeX should you ever get stuck (https://www.overleaf.com/learn).

2 Choosing a LaTeX Editor

The LaTeX base system is free software. There are a number of editors available to use. Everyone has a strong opinion on the best LaTeX editor. The only difference between editors is the ease of use. Ultimately, the documents produced by these editors all look the same. Pick the editor that feels most intuitive to you. I will recommend two editors: Overleaf and TeXworks. For beginners, I highly recommend Overleaf over TeXworks.

2.1 Overleaf

Overleaf (http://www.overleaf.com) is an online LaTeX editor. It is free to use but requires you to make an account. Overleaf allows you to collaborate with other users on projects and backs up all of your projects to the cloud. If you pay for a premium version, you can link Overleaf with Dropbox and GitHub. You can also download your projects and work on them offline (https://www.overleaf.com/learn/how-to/Working_Offline_in_Overleaf).

As an editor, Overleaf has a number of functions that many other editors lack:

- **Cloud-Based Backup:** Automatic backup to the cloud means that you are less likely to lose your projects if your computer crashes.
- **Error Identification:** Overleaf will tell you on what line an error has occurred. Many other editors will produce an error code but will not tell you what the error is or where it has occurred.
- **Spellcheck:** Overleaf includes spellcheck. Many other editors do not have spellcheck.
- **Command Suggestions:** Overleaf will prompt you with commands when you begin typing them. This is useful if you cannot quite remember the appropriate command.
- **Real-Time Compiling:** By default, Overleaf has two windows: the editor and the compiler. This allows you to view your changes in real time. Many other editors lack this functionality.
- **Document History:** Overleaf allows you to examine the document's history and restore previous versions.

What are the downsides to Overleaf? The biggest downside is that it is online. You will need an internet connection to use Overleaf as an editor. Sometimes Overleaf goes down for maintenance, which may limit your access to

documents. Nevertheless, I highly recommend using Overleaf as your primary editor. Its functionality is far superior to many other editors. However, I do recommend that you have an offline editor installed on your computer just in case you find yourself in a situation where you need it. For that, I recommend TeXworks.

2.2 TeXworks

There are countless offline LaTeX editors that work. I use TeXworks. Why do I use TeXworks? It was the first editor that I was instructed to install, I got used to it, and it serves my purposes. It allows you to view both the editor and the compiled document at the same time, but not as seamlessly as Overleaf. It does not have spellcheck, cloud backup, command suggestions, document history, or any of the other nice features of Overleaf. It is a bare-bones editor but it works. You can install TeXworks from http://www.tug.org/texworks/.

2.3 Other Editors

As I stated in the beginning, there are a number of LaTeX editors. If you find that neither Overleaf nor TeXworks meet your demands, then you may want to explore these other options.

- **R Markdown:** Some professors will require you to submit assignments in *R* Markdown. *R* Markdown allows you to produce LaTeX documents in RStudio. The benefit: You can conduct the statistical analysis and write the paper in the same document. *R* Markdown is incredibly powerful and can even be used to build websites. For a resource on *R* Markdown, see Yihui Xie, J.J. Allaire, and Garrett Grolemund's book *R Markdown: The Definitive Guide* at https://bookdown.org/yihui/rmarkdown/.
- **LyX:** LyX is a LaTeX editor that looks a bit more like Microsoft Word than other editors. It has spellcheck and formatting options that do not require the use of LaTeX commands. However, by default, LyX saves files in its own .lyx extension. Fewer people use .lyx and, therefore, when sharing documents you will need to export the document as a .tex file.

This is just the tip of the iceberg when it comes to LaTeX editors. A quick Google search will bring up countless others. You should find the editor that works best for you. However, it will still take time to get used to writing in LaTeX. Do not procrastinate learning LaTeX by spending countless hours searching for an editor that makes everything "easier." Overleaf is probably as easy as it gets.

3 Your First LaTeX Document

3.1 The Basic Set-Up

Every LaTeX document has two parts: the preamble and the document environment. The preamble defines the document class, loads packages, and sets the default formatting parameters for the document. For example, you may use the preamble to tell LaTeX that you want all of your text to appear in a blue, sans-serif, size-16 font. The `document` environment is where you will type the actual substance of your document.

To compile a LaTeX document, you need two things: (1) a document class (`\documentclass{...}`) and (2) the document environment (`\begin{document}` and `\end{document}`). The document class tells LaTeX what kind of document you are working on (e.g. `article`, `book`, `letter`, `Beamer`).These document classes have different functionalities. More often than not, you will want to use `article`. The preamble always begins with the document class command. You will mostly work with `\documentclass{article}`. You write the substance of your project between the `\begin{document}` and `\end{document}` commands. This is the document environment. Let's see an example:

```
\documentclass{article}

\begin{document}
Hello world.
\end{document}
```

If your LaTeX project lacks either of these elements, it will not properly compile. Thankfully, when you begin a new project in Overleaf, it will always include these elements for you.

LaTeX uses environments for many functions. An environment is just a command that tells LaTeX you're trying to format a particular type of item, such as a table, a list, or an image. You start a new environment with `\begin{...}` and end it with `\end{...}`. You place the name of the environment within the curly brackets.

3.2 Adding a Title

LaTeX allows you to add a title to your document. In the preamble, after `\documentclass{article}`, add the `\title{...}`, `\author{...}`, and `\date{...}` commands. At the start of the `document` environment, add `\maketitle`.

```
\documentclass{article}

%This is the title information.
\title{My First Latex Project}
\author{Nick Bednar}
\date{July 30, 2020}

\begin{document}

%This adds the title to the top of the document.
\maketitle

Hello world.
\end{document}
```

This will add a title header like the one that begins this document.

3.3 Packages

One of the most powerful features of LaTeX is the ability to add packages that increase its functionality. There are many many packages that may prove useful to you throughout your career. In the preamble, you use the `\usepackage{}` command to enable a package for use in the current document.

```
\documentclass{article}
\usepackage{amsmath}
\usepackage{tikz}
\usepackage{multicol}

%This is the title information.
\title{My First Latex Project}
\author{Nick Bednar}
\date{July 30, 2020}

\begin{document}
```

We do not have time to cover all of the LaTeX packages that you may find useful. Here is a list of commonly used packages:

- **amsmath:** This package adds a number of commands related to displaying equations, including the align environment.

- **amssymb:** This packages adds a number of math symbols that are missing from the LaTeX base.
- **amsfonts:** This packages adds additional fonts that are missing from the LaTeX base.
- **float:** This packages allow you to place images and tables using the `[H]` parameter.
- **tikz:** This package allows you to draw diagrams in LaTeX.

3.4 Section Headers and Tables of Contents

LaTeX has built-in commands for sections, subsections, and subsubsections. To use section headers, use the following commands:

```
\section{This is a Section}
\subsection{This is a Subsection}
\subsubsection{This is a Subsubsection}
```

This is a Section

This is a Subsection

This is a Subsubsection

LaTeX will automatically number section headers. If you do not want to add section numbers, place an asterisk before the first curly bracket.

```
\section*{This is a Section}
\subsection*{This is a Subsection}
\subsubsection*{This is a Subsubsection}
```

Section headers also allow you to easily add a table of contents to your document. Simply use the command `\tableofcontents` in the document environment. However, the table of contents will only list numbered section headers.

3.5 Typesetting Commands

LaTeX uses commands to format text. As you may have caught on, commands begin with a backslash. The text you want formatted by the command is placed

within a set of curly brackets. Here is a table of common typesetting commands for your reference.

```
\textit{This is italics.}
\textbf{This is bold.}
\textsc{This is smallcaps.}
\underline{This is underlined.}
\tiny{This is tiny.}
\scriptsize{This is script size.}
\footnotesize{This is footnote size.}
\large{This is large.}
\huge{This is huge.}
```

This is italics.
This is bold.
THIS IS SMALLCAPS.
This is underlined.

You can use LaTeX to make lists using either the `enumerate` or `itemize` environments. The `enumerate` environment numbers items in the list:

```
\begin{enumerate}
    \item item 1.
    \item item 2.
    \item item 3.
\end{enumerate}
```

1. item 1.
2. item 2.
3. item 3.

The `itemize` environment makes a bulleted list.

```
\begin{itemize}
    \item item 1.
    \item item 2.
    \item item 3.
\end{itemize}
```

- item 1.
- item 2.
- item 3.

Many punctuation symbols are reserved by LaTeX and will not print on their own. For example, the dollar sign ($) begins the `math` environment. If you want to use one of these reserved symbols, then add a backslash in front of it. For example, `\$` will produce $. The exception to this rule is the backslash. Writing `\\` does not produce a backslash because LaTeX reserves this as a linebreak. Instead, you must use the command `\backslash`.

3.6 Tables

Tables are created using the `tabular` environment:

```
\begin{tabular}{...}
\end{tabular}
```

You must declare the number of columns in the table following the begin command. For each column, add the letter that corresponds to the alignment of the column (`l`=left, `c`=center, `r`=right). For example:

```
\begin{tabular}{lcr}
\end{tabular}
```

will produce a three column table, where the first column is left aligned, the second column is center aligned, and the third column is right aligned.

You separate columns within the table using an ampersand (&). You end a row with two backslashes (\\). For example:

```
\begin{tabular}{lcr}
Item & Class & Cost \\
\hline
Apple & Fruit & \$.99 \\
Orange & Fruit & \$.89 \\
Lettuce & Vegetable & \$1.25
\end{tabular}
```

Item	Class	Cost
Apple	Fruit	$.99
Orange	Fruit	$.89
Lettuce	Vegetable	$1.25

Notice that I use the command `\hline` to add a horizontal line between two rows. If you want to add a line between two columns, then add a vertical bar (`|`) where you declare the columns. (\\). For example:

```
\begin{tabular}{l|c|r}
Item & Class & Cost \\
\hline
Apple & Fruit & \$.99 \\
Orange & Fruit & \$.89 \\
Lettuce & Vegetable & \$1.25
\end{tabular}
```

Item	Class	Cost
Apple	Fruit	$.99
Orange	Fruit	$.89
Lettuce	Vegetable	$1.25

If you want to add a caption to your table, then enclose your `tabular` environment in the table environment. Add the `\caption` command after you begin the `table` environment but before you begin the `tabular` environment. You may also place the `\caption` command after you end the `tabular` environment but before you end the `\table` environment.

```
\begin{table}[H]
\caption{Here is a Table}
```

```
\centering
    \begin{tabular}{l|c|r}
    Item & Class & Cost \\
    \hline
    Apple & Fruit & \$.99 \\
    Orange & Fruit & \$.89 \\
    Lettuce & Vegetable & \$1.25
    \end{tabular}
\end{table}
```

This code produces:

Table 1: Here is a Table

Item	Class	Cost
Apple	Fruit	$.99
Orange	Fruit	$.89
Lettuce	Vegetable	$1.25

The `\caption{}` command names the table. The `\centering` command centers the table within the table environment.

How does LaTeX decide where to place the table? LaTeX uses "floats" to place tables and images. Floats deal with problems when an object would not fit on the current page. As a result, tables and images may not appear exactly where you place them in the editor. Notice the `[H]` after `\begin{table}`. This allows for some control over the placement of the table or the image. There are several commands that can be used to instruct LaTeX about where to place a table or image.

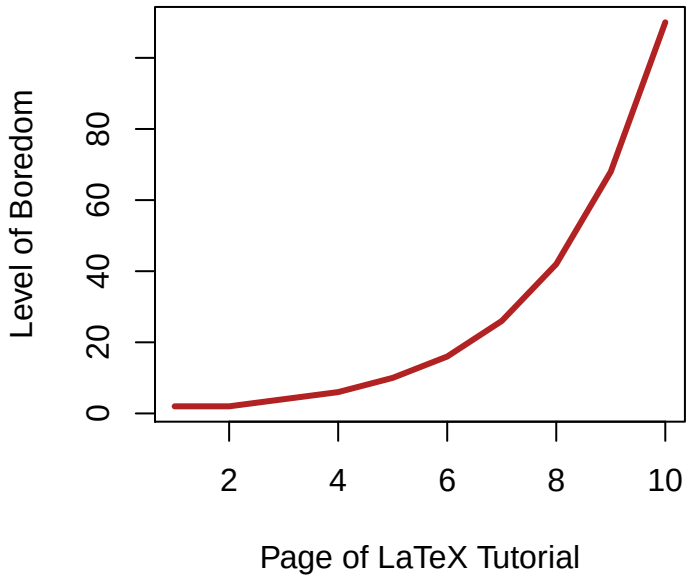
- **!**: Overrides LaTeX's default positioning.
- **h**: Places the table or image approximately here.
- **t**: Places the table or image at the top of the page.
- **b**: Places the table or image at the bottom of the page.
- **p**: Creates a special page for the table or image.
- **H**: Orders LaTeX to place the table or image precisely where it is in the editor. This may result in the table or image running into the margins or getting cut off at the bottom of the page. You should generally avoid using `[H]`. You must use the float package (`\usepackage{float}`) in order for this position to work.

It is possible to put multiple positions in the command in case LaTeX cannot use the first position. The most common order is `[!htbp]`.

3.7 Images and Figures

Images are added to LaTeX using the `\includegraphics[...]{...}` command. The image must be in the same folder as your LaTeX document. In Overleaf, you can upload images to the project. LaTeX will accept .pdf, .png, or .jpg images.

```
\includegraphics[scale = 1]{Rplot03.pdf}
```



LaTeX will allow you to resize an image. There are two options for resizing the image. First, you can use `[scale=X]` to increase or decrease the size by a particular percentage. For example, `[scale=.5]` will resize the image to 50% of its original size. Second, you can use `[width=X, height=X]` to resize the image to a particular width and height. For example, `[width=5in, height=5in]` will resize the image so that it is five inches tall and five inches long.

Like the `table` environment, we can add a caption to the image by enclosing it in the `figure` environment.

You can also place two images side-by-side by using the `minipage` environment. Within the `\begin{minipage}{...}` command, you instruct LaTeX how much space you want the minipage to occupy on the page. In the example below, `{.5 \textwidth}` instructs LaTeX to limit each minipage to half the size of the document.

```
begin{figure}[H]
\caption{A Comparison of My Mom's Dog to Cult-Classic Creatures}
\centering
\begin{minipage}{.5\textwidth}
  \centering
  \includegraphics[trim = 0cm 10cm 0cm 25cm,
  clip = true, width= .9 \linewidth]{zoe1.jpg}
  Zoe, a 10 year old Pomeranian.
  \label{fig:test1}
\end{minipage}%
\begin{minipage}{.5\textwidth}
  \centering
  \includegraphics[width= .9 \linewidth]{Fizzgig.jpg}
  \label{fig:test2}
  Fizzgig from \textit{The Dark Crystal}.
\end{minipage}
\end{figure}
```

Figure 2: A Comparison of My Mom's Dog to Cult-Classic Creatures

Zoe, a 10 year old Pomeranian.

Fizzgig from *The Dark Crystal*.

Within the `\includegraphics[...]{...}` graphics command, I have instructed LaTeX to crop the image of Zoe. Within the brackets, `trim` will crop the image. `Trim` requires four measures to run. It crops left, then bottom, then right, and then top. If you do not want `trim` to crop a particular side, just use `0cm` for that side. You must follow `trim` with `clip=true`.

3.8 Equations

With the `amsmath` package enabled, LaTeX provides several environments for writing equations. You may use either the `equation` environment or the `align` environment. I prefer to use the `align` environment because it allows me to align several equations vertically. Within the `align` environment, the ampersand (`&`) lines up the equations. The backslashes (`\\`) create a linebreak. If you want to add text to your equation, you can use the `\intertext{}` command.

```
\begin{align*}
    y&=2x+3 \\
    y&=4x-1
\end{align*}
```

$$y = 2x + 3$$
$$y = 4x - 1$$

You can also embed a `math` environment within a paragraph by enclosing the equation between two dollar signs (`$`). For example, writing `$\hat{y}=x^{2}$` will produce $\hat{y} = x^2$.

Math mode has countless commands. The following table lists some of the most common symbols that you will need.

Name	Command	Command Example	Compiled Example
Superscript	`^{}`	`x^{2}`	x^2
Subscript	`_{}`	`x_{i}`	x_i
Hat	`\hat{}`	`\hat{x}`	$\hat{x}$
Bar	`\bar{}`	`\bar{x}`	$\bar{x}$
Fraction	`\frac{}{}`	`\frac{1}{2}`	$\frac{1}{2}$
Blackboard Bold	`\mathbb{}`	`\mathbb{E}`	$\mathbb{E}$
$\times$	`\times`	`2 \times 2=4`	$2 \times 2 = 4$
$\div$	`\div`	`4 \div 2 = 2`	$4 \div 2 = 2$
$\neq$	`\neq`	`1+1 \neq 3`	$1 + 1 \neq 3$
$\leq$	`\leq`	`x \leq 2`	$x \leq 2$
$\geq$	`\leq`	`x \leq 2`	$x \leq 2$
$\pm$	`\pm`	`1 \pm 1 =`	$1 \pm 1 =$
Square Root	`\sqrt{}`	`\sqrt{2}`	$\sqrt{2}$
Summation	`\sum`	`\sum_{i=1}^{n}`	$\sum_{i=1}^{n}$
Product	`\prod`	`\prod_{i=1}^{n}`	$\prod_{i=1}^{n}$
∞	`\infty`		
α	`\alpha`		
β	`\beta`		
χ	`\chi`		
ϵ	`\epsilon`		
λ	`\lambda`		
μ	`\mu`		
ω	`\omega`		
π	`\pi`		
σ	`\sigma`		
τ	`\tau`		
Δ	`\Delta`		
$\mathcal{N}$	`\mathcal{N}`		

Let's show some common equations to put this all together.

Probability Mass Function

```
\begin{align*}
    F(x)=P(X \leq x) = \sum_{k \leq x}f(x)
\end{align*}
```

$$F(x) = P(X \leq x) = \sum_{k \leq x} f(x)$$

Law of Large Numbers

```
\begin{align*}
```

```
    \bar{X}_{n} = \frac{1}{n} \sum_{i=1}^{n} X_{i}
    \rightarrow \mathbb{E}(X)
\end{align*}
```

$$\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i \rightarrow \mathbb{E}(X)$$

Central Limit Theorem

```
\begin{align*}
    \frac{\bar{X}_{n}-\mathbb{E(X)}}
    {\sqrt{\frac{\mathbb{V}(X)}{n}}} \sim
    \mathcal{N}(0,1)
\end{align*}
```

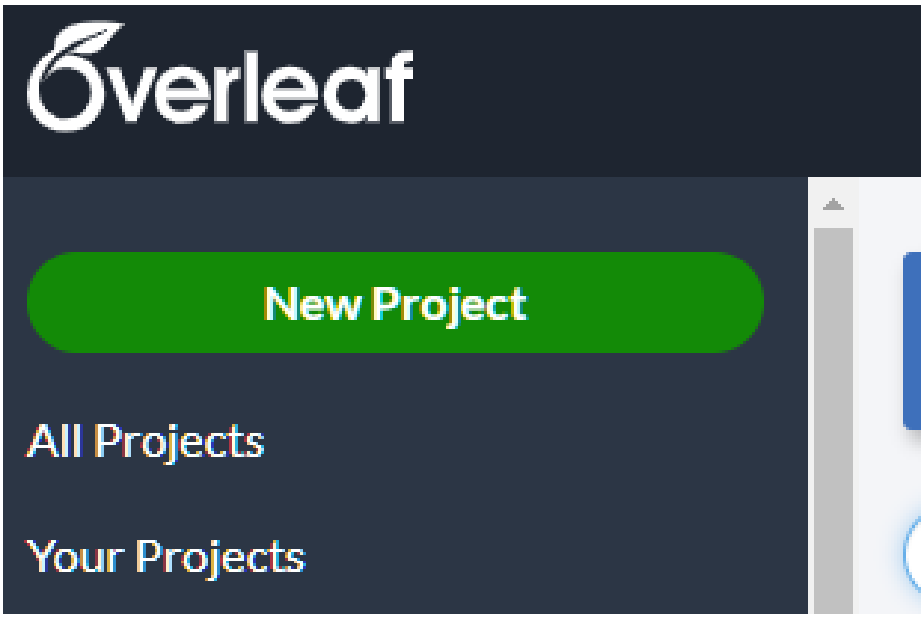
$$\frac{\bar{X}_n - \mathbb{E}(\mathbb{X})}{\sqrt{\frac{\mathbb{V}(X)}{n}}} \sim \mathcal{N}(0,1)$$

4 Overleaf Basics

I recommend that you use Overleaf for all of your LaTeX documents. I described the many benefits of Overleaf over other editors above. You will first need to make an account to use Overleaf. This section provides a few tips for using overleaf.

4.1 Creating a New Project

To start a new project in Overleaf, click the large, green "New Project" button in the upper corner of the homepage.



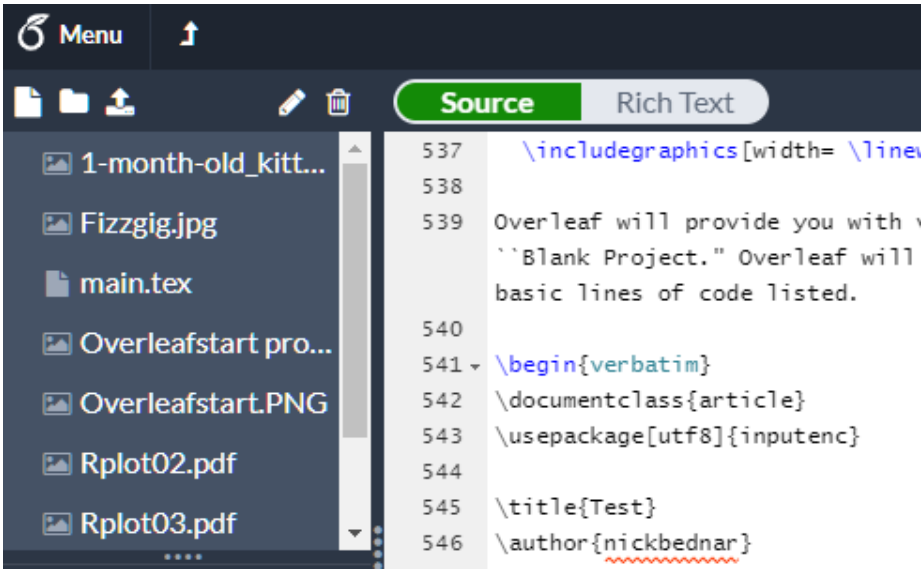
Overleaf will provide you with various options for how to begin your project. If you are starting a new project, select "Blank Project." Overleaf will prompt you to name the project. When you open the project, Overleaf already has some basic lines of code listed.

```
\documentclass{article}
\usepackage[utf8]{inputenc}

\title{Test}
\author{nickbednar}
\date{August 2020}

\begin{document}

\maketitle

\section{Introduction}

\end{document}
```

If you need to upload a project from your computer, first place that project in a .zip file. Then, select "Upload Project." Overleaf will prompt you to upload the .zip file of your project.

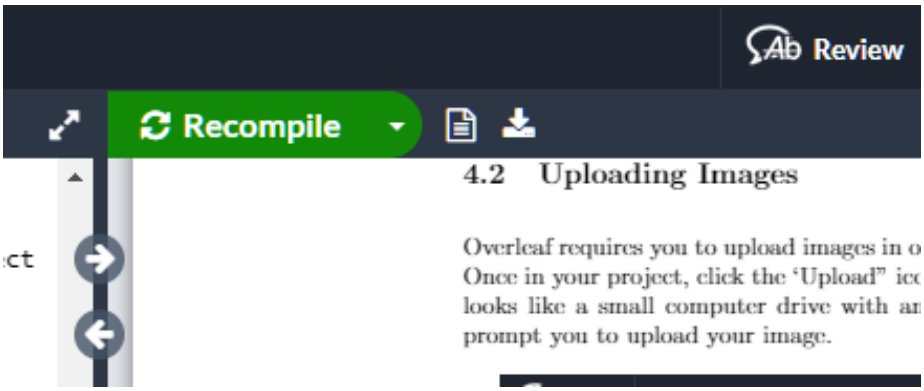
4.2 Uploading Images

Overleaf requires you to upload images in order to include them in your project. Once in your project, click the "Upload" icon in the upper left corner. This icon looks like a small computer drive with an arrow above it. Overleaf will then prompt you to upload your image.



4.3 Compiling and Downloading a .PDF

After you have started working on your project, you will want to compile the code to see what it looks like it. Overleaf automatically compiles the code every so often. To manually compile the code, click the large, green "Recompile" button above the .PDF window. To download a .PDF click the "Download .PDF" icon to the right of the "Recompile" button. This icon looks like a small computer drive with arrow pointing toward it.



5 I Have Broken LaTeX. Now What?

There are plenty of things that may go wrong when working with LaTeX. If you encounter a problem, take a deep breath. Everything will be OK! Here are my generic troubleshooting tips:

1. Check the error code. Does it make sense to you? If it identifies a particular line that produced the error, then look at the line and see if you can spot the error.

2. If you cannot solve the error on your own, skim the list of common errors provided below and see if you can identify the problem.

3. If the list of common errors does not provide you with a solution, copy and paste the error into Google. It is highly unlikely that you are the first person to experience this error. You may also try searching the TeX Stack Exchange (https://tex.stackexchange.com/) to see whether anyone has experienced a similar error.

4. If you are working with an offline compiler, check to make sure that the .pdf document is not open on your computer. LaTeX cannot compile the .pdf while it is open in another program, such as Adobe Reader, because it must save over the existing file. If the document is not open in another program, then try restarting your computer. Sometimes editors just break. I do not know why this happens but restarting the computer often gets them to work again.

5. Ask someone else to examine your code. Sometimes you are so immersed in your code that you become incapable of spotting the error. If you are e-mailing your code to someone, then attach the entire .tex document and describe the error that you are experiencing. Try to avoid giving a vague

description of the error, such as "It won't compile." Describe what you were doing immediately before the error occurred (e.g. "I was trying to include an image, but now it won't compile.") and what you have tried to do to solve the error.

6. Post on the TeX Stack Exchange (https://tex.stackexchange.com/). The users on Stack Exchange are incredibly knowledgeable. However, make sure you thoroughly search the website for your problem before you post. They will expect you to post your code, describe the error, and describe what steps you have taken to try to solve the error. If you fail to provide this information, you may incur their wrath.

Below, I have tried to preempt some common errors that occur when first working with LaTeX.

- **! Undefined Control Sequence**: This error usually appears when you have a typo in one of your commands. Check to ensure that you have spelled all of your commands correctly. Commands are case sensitive. If you have accidentally capitalized a command, then it will trigger this error. This may also happen if you use a command from a package but fail to load the necessary package. Check your packages.
- **[Environment] ended on input line [x] ended by** `\end{document}`: Did you recently add a new environment—such as an `align`, `equation`, `figure`, or `table` environment—to the document? Check whether you ended the environment (i.e. `\end{...}`).
- **Missing $ inserted**: One of two things has happened. Did you start math mode using `$`? You may have forgotten to end math mode with another `$`. Did you intend to use the `$` symbol to display a dollar amount? You may have forgotten to escape the dollar sign by adding a backslash in front of the symbol (i.e. `\$`).
- **Misplaced alignment tab character**: Did you intend to use an ampersand symbol (&) instead of the word "and"? You may have forgotten to escape the ampersand by adding a backslash in front of the symbol (i.e. `\&`).
- **Words do not appear in the document**: Do words appear in your editor but fail to appear in the compiled version? Check whether you have accidentally begun typing after the `\end{document}` command.
- **Misplaced** `\noalign`: This error tends to occur when you have used `\hline` incorrectly. You must use `\hline` in a table environment. If you want a horizontal line outside of a table environment, use `\hrule`.