An Introduction to LaTeX

Tacye Hong

2021-2022

1 Overview

There are many LaTeX software for you to pick, many (if not all) are free. During this course, we will focus on Overleaf as it is very user-friendly, especially for beginners. Other LaTeX software includes Texmaker and TeXstudio, among others (there is a whole Wikipedia page on this). You can access Overleaf by creating an account with `overleaf.com`. Everything is transferable so the LaTeX file you created on Overleaf can be exported, and then imported to Texmaker, for example.

1.1 Benefits of Overleaf

Overleaf usually prompts the commands as you are typing them, so it saves time and is especially helpful for more complicated commands, such as the ones for figures and tables. Overleaf also provides the `end` command as you type the `begin` command, lowering the chance of you forgetting to end your commands properly. Errors in Overleaf are very easy to spot (they sometimes highlight the whole thing in red! Even someone sitting 5 rows behind you can see it), so you can correct them easily enough.

Overleaf created many guides for people to learn LaTeX,[1] as well as many templates, such as CV and cover letter templates. You can also share your Overleaf document with one other person, and it will be updated in real time, much like Google Doc, so collaboration would be less of a pain.

[1]There guides can be accessed here : `https://www.overleaf.com/learn`. They should be sufficient for beginners, though one might occasionally want to read the full documentation on CTAN.

2 Setting Up

To start, you should always start a new project, as shown in Figure 1. Your project will contain your main `.tex` file, any images and your bibliography library. You have the options of starting a blank project, or uploading a project. The latter is useful if you downloaded a template from somewhere, or someone sent you their project in `.zip` file format (different from sharing).

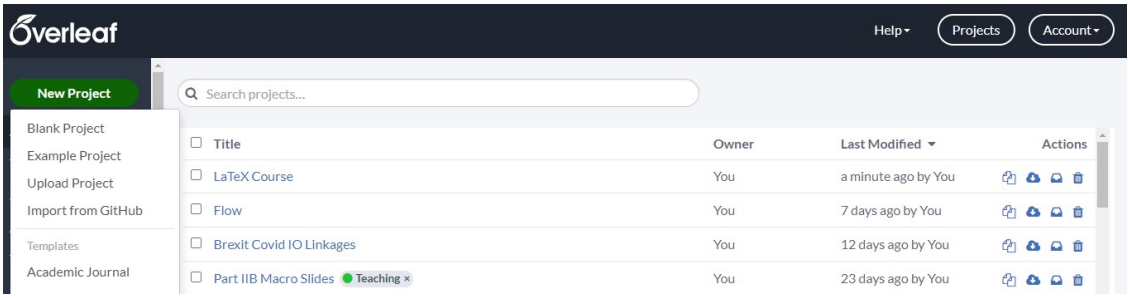


Figure 1: Starting a Project

Once we started a project, we will automatically have a `.tex` file ready, titled `main.tex`. We can see that there are 3 different panels, as shown in Figure 2. From left to right: 1) The folder directory – where you can see all the files within this project – and the file outline for the `main.tex` file; 2) Main "command window" – where you type up your document; 3) PDF preview – where you can preview the file after recompiling the file (every 5 seconds, if you are anything like me).

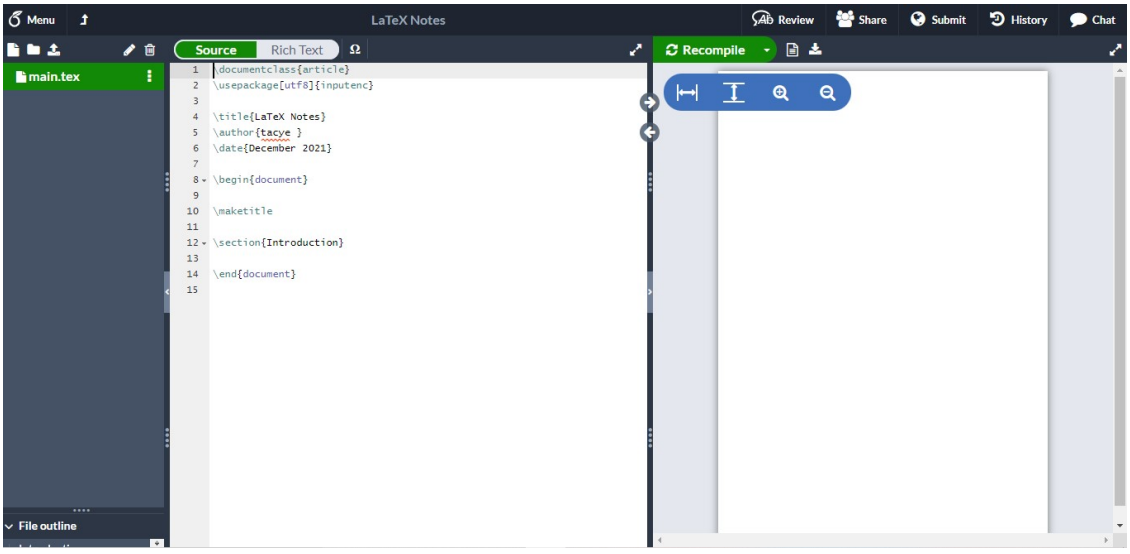


Figure 2: The different panels of Overleaf

Other buttons here include

- New file button – for creating new `.tex` files

- New folder button – for if you want to have your figures all in a folder or if you want to create different (maybe similar) subprojects within the same project
- Upload button – for uploading figures and bibliography
- Edit – for renaming file names
- Delete – try and make a wild guess!

Above all these, there is a Menu. Surprisingly, there aren't too many buttons you will find useful. The only one worth mentioning is perhaps "Main Document", which allows you to set the `.tex` you are currently working with – great if you have multiple versions of a document.

We will be working mostly with the command window during the class. As you can see, there are already some lines ready:

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

\title{ }
\author{Tacye}
\date{December 2021}
\begin{document}
\maketitle
\section{Introduction}
\end{document}
```

We need to first declare a document class. By default, this is `article`. This is what you should use for your dissertation. Other document classes include `letter` and `beamer`, the latter is LaTeX's version of slides.[2]

We can type the title of the document, as well as the author and date in the provided brackets. You can either type a set date or type `\today`, and Overleaf will automatically update the document's date when you recompile the file. The command tells LaTeX to create the title.

The default setup puts the title at the top of the first page. For dissertations, it is likely that you would want to have the title and the abstract on a separate page. In that case, we would need to define a `titlepage` properly `within` the document, using `\begin{titlepage}` and `\end{titlepage}`. This `titlepage` will now contain `title`, `author`, and `date`, and they all need to go after `\begin{document}`.

[2]You are welcome to write your own class, but it is unlikely that you will need to do so. You can also import a user-made class by declaring it in the `documentclass` and uploading the `.cls` file.

The following is what the new structure should look like. Try to figure out yourself why the ordering of the command makes sense:

```
\begin{document}

\begin{titlepage}
\title{LaTeX Course 2022}
\author{Tacye Hong}
\date{\today}
\maketitle
\begin{abstract}
    This is where we type the abstract
\end{abstract}
\thispagestyle{empty}
\end{titlepage}
```

As you can see, the abstract is contained between the \begin{abstract} and \end{abstract} commands. To remove the page number from the title page, we use \thispagestyle{empty}.

2.1 Preamble

We call everything that goes before \begin{document} the "preamble".

We will need to load many packages in the preamble. All the packages can be loaded using \usepackage. You can load all the packages you want in the same bracket, or load them one at a time, but it is better if we group the related packages to keep them tractable. By default, \usepackage[utf8]{inputenc} has been added. You can ignore this.

One of the first packages you will want to load is geometry, which allows you to declare the margin. Once you have loaded the package, you can select your desired margins like this:

```
\usepackage{geometry}
\geometry{left=1.0in,right=1.0in,top=1.0in,bottom=1.0in}
```

The geometry package is very flexible and you can read about it here: https://www.overleaf.com/learn/latex/Page_size_and_margins.

We can also change the indentation of each paragraph, and the space between different paragraphs. We can change it for the entire document by declaring these in the preamble:

```
\setlength{\parindent}{ em}
\setlength{\parskip}{ em}
```

where you will need to fill in how many `em` you want them to be. `em` is one of LaTeX's measurement system and you will soon be able to estimate how big it is. As a rule of thumb, `1em` is sufficient for both `parindent` and `parskip`. By declaring `parindent` to have `1em`, you will have a `1em` indentation for every paragraphs starting from the second paragraph in each section. If there is a particular paragraph that you do not want any indentation, you can add a `\noindent` in front of that paragraph. If you want there to be no indentation for the entire paper, you can input `0em`.

More about formatting paragraphs can be found here: `https://www.overleaf.com/learn/latex/Paragraph_formatting`

3 Main Document

In this section, we explore how to type up the main text of a LaTeX document. In general, it is very simple – you just start typing.

3.1 Creating Sections

In any paper that we write, we need to break it down in sections and subsections. To create a new section, we can simply type

```
\section{ }
```

where the bracket contains the title of the section. Overleaf will number the sections for you. A subsection is created similarly, by typing

```
\subsection{ }
```

We can cross-reference a section by creating labels and referencing them (either before or after the section appears). This is achieved by typing

```
\label{ }
```

next to the section name, where we need to define a label in the brackets. Whenever we want to reference that section, we just need to type

```
\ref{ }
```

where the bracket contains the defined label.

Hence, for example, we discussed how to set up an Overleaf document in Section 2. This cross-reference is done by first labelling the section

```
\section{Setting Up} \label{setup}
```

and then referencing the label here:

```
...set up an Overleaf document in Section \ref{setup}.
```

If you do not want to number a section or subsection for any reason, you can do so by adding an asterisk as such:

```
\section*{ }
```

This, however, would mean that you cannot reference it.

3.2 Footnote

To add a footnote, we can just type

```
\footnote{}
```

right where we want it to appear.[3] It will then appear at the bottom of the page, in a smaller font size. To demonstrate, the above is achieved by typing exactly this:

```
...want it to appear.\footnote{Example footnote.}
```

3.3 Text Colors, Bold, Highlight

To change the colors of a text, we need to first install the `xcolor` package by typing

```
\usepackage{xcolor}
```

in the preamble. We can then type in different colors, such as red using

[3] Example footnote.

```
\textcolor{ }{ }
```

where the first set of brackets is for the desired color and the second set is for the text. Hence, for example, to type the above bit in red, we will type this:

```
\textcolor{red}{different colors, such as red}
```

Other colors in the `xcolor` package can be found here: `https://www.overleaf.com/learn/latex/Using_colours_in_LaTeX#Named_colours_provided_by_the_xcolor_package`[4]

To highlight texts, we will need to install the `soul` package in the preamble, and highlight using:

```
\hl{}
```

Changing words into **bold** or *italic* is very simple. The commands are, respectively,

```
\textbf{}
```

and

```
\textit{}
```

Underlining words is also very simple:

```
\underline{}
```

3.4 Making Lists

In this section, we discuss making a list (and checking it twice! Sorry I wrote this during the holidays). We can either make an ordered list by using `enumerate` or make an unordered list by using `itemize`. Both of these commands need to follow a `\begin` and use `\item` to indicate a new, well, item. Hence, the structure looks like this:

```
\begin{enumerate}
    \item
```

[4]Yes, this is American so y'all, it's gray, not grey.

```
\end{enumerate}

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

You should add an `\item` for each additional item. These will show up, respectively, as

1.

and

•

We can also make nested lists, and alternating between `enumerate` and `itemize`:

1. Top 3 things to do in Cambridge
    - your dissertation
2. Top 3 things to NOT do in Cambridge
    - miss your dissertation deadline

Note the careful placement of the `begin` and `end` environments:

```
\begin{enumerate}
    \item Top 3 things to do in Cambridge
    \begin{itemize}
        \item your dissertation
    \end{itemize}
    \item Top 3 things to NOT do in Cambridge
    \begin{itemize}
        \item miss your dissertation deadline
    \end{itemize}
\end{enumerate}
```

We can also customize how these lists appear, such as using Roman numerals, and changing the style of the bullet but this is left as further reading: `https://www.overleaf.com/learn/latex/Lists#Customizing_lists`.

3.5 Other Notes

We can write comments using `%`. This does, however, mean that if you meant to type the symbol, you will need to type `\%`. Also, as you might have noticed, we often use the curly brackets {} to include items, commands etc. Hence, we also need to include backslash when wanting to type curly brackets, i.e. `\{\}`. The last one is `&`. This is usually used in LaTeX to align tables and equations, as we will discuss later. Hence, you will need to type `\&`.[5]

4 Equations

To type equations, we should install the `amsmath` package and the `amssymb` package in our preamble:

```
\usepackage{amsmath, amssymb}
```

There are different ways to type an equation, depending on how we want it to appear in our text. First, let's consider putting an equation in a separate line, and centered.

```
\begin{equation}
    y = 2x
\end{equation}
```

This will then appear as

$$y = 2x \tag{1}$$

We can also label and reference our equations, using `label` and `ref`, or `eqref`. This shows up as either 1 or (1), respectively. If we do not wish to number a particular equation, we can put an asterisk next to the word `equation` in the command:

```
\begin{equation*}
\end{equation*}
```

The equation will then appear as:

$$y = 2x$$

Alternatively, we can include an equation in a line of text, by using either `$ $` or `\(\)`. For example, we can include $y = 2x$ in the line of text by typing

```
...can include \(y = 2x\) in the line...
```

[5] If you accidentally typed & without the slash, Overleaf is very sassy when pointing it out for you.

or

```
...can include $y = 2x$ in the line...
```

As you might have noticed, these x and y look differently from the usual x and y in the text mode. Hence, if there is an equation that holds only if a certain condition is true, such as the one below,

$$y = 2x, \quad \text{if} \quad x > 0 \tag{2}$$

we will need to figure out how to type the 'if' properly, as I'm sure no one wants to be writing

$$y = 2x, \quad if \quad x > 0$$

To write text mode in the equation environment, we can use `textrm`. Hence, the correct code for Equation 2 is:

```
\begin{equation}
    y = 2x, \quad \textrm{if} \quad x > 0
\end{equation}
```

You might have noticed the unfamiliar `\quad` there. In fact, spaces do not work in math mode. If we haven't added the `\quad`, our equation would have looked like this even if we have fixed the 'if':

```
\begin{equation}
    y = 2x, \textrm{if} x > 0
\end{equation}
```

produces

$$y = 2x, \text{if} x > 0$$

Thus, we have to add spaces manually, using one of the spacing options allowed in LaTeX. There are different spacing options, such as `\,` `\quad` `\qquad`. The full list can be found here: `https://www.overleaf.com/learn/latex/Spacing_in_math_mode`. It is also possible to brute force it, by inserting (horizontal) spaces using `\hspace{ em}`, where you can adjust the desired length of the space using `em`.[6]

Bracket sizes need to be adjusted, otherwise your fractions might end up looking like this:

$$(\frac{(7y + 2x)^2}{2})^2$$

LaTeX will automatically adjust for sizes if you use `\left` and `\right` with the brackets. For example, we can adjust the size of the brackets above by typing:

[6]The PhD student next to me saw it as I'm typing this and said "What do you mean by brute force. This is *the* way to go. It's flexible, okay??"

```
\begin{equation*}
    \left(\frac{(7y+2x)^2}{2}\right)^2
\end{equation*}
```

The fraction now becomes this:

$$\left(\frac{(7y+2x)^2}{2}\right)^2$$

Another method is to declare the size explicitly using one of the following options before any brackets are used:

```
\big \Big \bigg \Bigg
```

These correspond to the following sizes (if parenthesis are used):

$$(\Big(\bigg(\Bigg($$

$$\left(\frac{(7y+2x)^2}{2}\right)^2$$

The `left` and `e`nvironment has the benefit of letting Overleaf picking the correct size of the bracket, but they cannot be split into different rows. This is because they come in a pair, hence LaTeX will not like it if you make them do a long distance relationship. On the other hand, `bigg` etc. can be used in different rows but we need to determine the ideal size ourselves.

4.1 Subscript, Superscript, and Symbols

We can add subscripts, such as a time subscript, to our equations. Let's try adding time t subscript to our variables, such that the equation now reads $y_t = 2x_t$:

```
y_{t} = 2x_{t}
```

Now, let's consider adding a quadratic term to our equation $y_t = 2x_t + 5x_t^2$

```
y_{t} = 2x_{t} + 5x_{t}^{2}
```

The curly brackets are used such that LaTeX understands which part needs to typed as a superscript/subscript. Otherwise, LaTeX will only apply it for the first character. We are not allowed to do a double superscript or a double subscript, meaning that if you want to add a superscript A to the variables above to indicate say country A, you cannot type:

```
y^{A}_{t} = 2x^{A}_{t} + 5x^{A}_{t}^{2}
```

Instead, you need to include curly brackets such that the last term looks like `5{x^{A}_{t}}^{2}`. LaTeX will now be able to interpret it properly:

$$y_t^A = 2x_t^A + 5x_t^{A^2}$$

You can also include parenthesis instead of curly bracket

$$y_t^A = 2x_t^A + 5(x_t^A)^2$$

The two codes for the above are, respectively,

```
\begin{equation*}
    y^{A}_{t} = 2x^{A}_{t} + 5{x^{A}_{t}}^{2}
\end{equation*}
\begin{equation*}
    y^{A}_{t} = 2x^{A}_{t} + 5(x^{A}_{t})^{2}
\end{equation*}
```

There are often many Greek letters and math symbols we need to include in the equations, most, if not all, start with a \. For example, while we can easily type $>$ and $<$, using $<$ and $>$ found on the keyboard, $\geq$ and $\leq$ need to be typed using `\geq` and `\leq`.

Here, let's consider adding an error term ε_t to our equation such that it now reads: $y_t = 2x_t + 5x_t^2 + \varepsilon_t$. To type this, we will need to type the Greek letter `\varepsilon`.

```
y_{t} = 2x_{t} + 5x_{t}^{2} + \varepsilon_{t}
```

For those who are wondering, there is a `var` because `\varepsilon` gives you ε and `\epsilon` gives you ϵ. Also, be careful when capitalizing the Greek letter, as, for example, `\Sigma` gives you Σ while `\sigma` produces σ.

You can find the main list of Greek letters and symbols here: `https://www.overleaf.com/learn/latex/List_of_Greek_letters_and_math_symbols`, with further links on that site. There's no easy way – you will need to keep consulting this page until you start slowly memorizing them.

We also sometimes need to type different fonts in our equations. For example, $\mathbb{R}$ is often used to indicate real numbers while $\mathcal{A}$ is used to define an event space. These fonts are typed using `mathbb` and `mathcal` respectively. Hence, $\mathbb{R}$, $\mathbb{E}_t$ and $\mathcal{A}$ can be typed up using, respectively,

```
\mathbb{R} \mathbb{E}_{t} \mathcal{A}
```

Other fonts in the math mode can be found here: `https://www.overleaf.com/learn/latex/Mathematical_fonts`.

4.2 Aligning

Our equations are sometimes too long to fit in a single line, or we may wish to present it in multiple lines for better clarity. There are a number of ways to do this:

The first is `split`, which should be used within the `equation` environment. You will then use `&` to align the equations and `\\` to define a new line. For example, to get to this format

$$\begin{aligned} y &= 2x + 3x \\ &= 5x \end{aligned} \tag{3}$$

you need to type

```
\begin{equation}
\begin{split}
y & = 2x + 3x \\
 & = 5x
\end{split}
\end{equation}
```

When we split the equation this way, it is ultimately still "one equation", hence there is only one number. If we instead want a number per line, we will need to use `align`. `align` exists outside of the `equation` environment, so you can use

```
\begin{align}
\end{align}
```

without the \begin{equation} and \end{equation}.

Hence,

$$y = 2x + 3x \tag{4}$$
$$= 5x \tag{5}$$

is typed using the following command:

```
\begin{align}
    y & = 2x + 3x \\
    & = 5x
\end{align}
```

This now gives you one number per line.

If we only want to number a specific line, for example, only the final derivation such as the following:

$$\begin{aligned} y &= 2x + 3x \\ &= 5x \end{aligned} \tag{6}$$

we can do so by adding `\nonumber` on the lines we do not want any numbering:

```
\begin{align}
    y & = 2x + 3x \nonumber\\
    & = 5x
\end{align}
```

We can also align multiple equations such as:

$$x = 2y \qquad\qquad y = \frac{5x^2}{2} + 5 \tag{7}$$

$$y = \frac{x}{2} \qquad\qquad x = \sqrt{\frac{2(y-5)}{5}} \tag{8}$$

by using multiple `&`:

```
\begin{align}
    x & = 2y & y & = \frac{5x^2}{2} + 5 \\
    y & = \frac{x}{2} & x & = \sqrt{\frac{2(y-5)}{5}}
\end{align}
```

Notice that we have three pairs of `&`, one for evenly distributing the two sets of equations across the page, and two for aligning the equations within the sets.

5 Tables and Figures

We now turn to discuss tables and graphs. This is perhaps one of the most cumbersome part of LaTeX.

5.1 Tables

The guide on Overleaf is very long (but very detailed and useful): `https://www.overleaf.com/learn/latex/Tables`.

To start, we load the following package in the preamble:

```
\usepackage{graphicx}
```

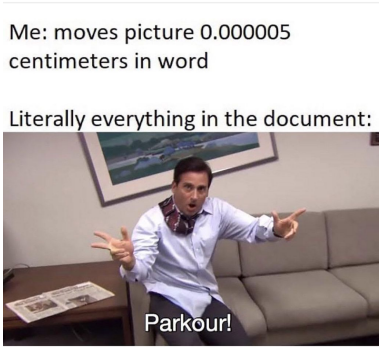
Overleaf has already made this much easier by providing the following when you type `\begin{table}`:

```
\begin{table}[]
    \centering
    \begin{tabular}{c|c}
         &  \\
         &
    \end{tabular}
    \caption{Caption}
    \label{tab:my_label}
\end{table}
```

The `table` is the environment that contains the `tabular`, i.e. the actual table structure itself, and the caption, among others.[7] The square bracket `[]` next to the `\begin{table}` is to define the position of the table environment. It does take some time to understand but these are the following options:

`h`	Place the float approximately at the same point it occurs in the text
`t`	Position at the top of the page
`b`	Position at the bottom of the page
`p`	Put on a special page for floats only
`!`	Override internal parameters LaTeX uses for determining "good" float position
`H`	Places the float at precisely the location in LaTeX– Requires the `float` package

You should use either `h` or `h!` for the majority of the time but occasionally it does feel like one of those memes when using Word Doc:



`\centering` ensures the table and caption are centered.

[7] You can use the caption as a table title and add an additional explanatory footnote.

We now focus on {c|c}. First of all, this default option will create 2 columns (as indicated by a total of 2 letters), separated by a vertical line (as indicated by the |) and the columns will be centered. You can use `l` or `c` or `r` to declare how you want your content to be aligned – left, centered, and right respectively.

Let's compare the following table output against its code:

Date	Meaning	Description
Jan 1	New Year	It's 2022!
Jan 2	Second Day	Ross becomes a loser

Table 1: Important Dates

```
\begin{table}[h]
\centering
\begin{tabular}{l | c c}
    Date & Meaning & Description \\
    \hline
    Jan 1 & New Year & It's 2022! \\
    Jan 2 & Second Day & Ross becomes a loser
\end{tabular}
    \caption{Important Dates}
    \label{tab:dates}
\end{table}
```

We have three columns, as indicated by the three letters, `l`, `c` and `c`. The first column is aligned left while the rest are centered. Each `&` indicates a new column while each \\ creates a new row. If we want to add a horizontal line between rows, we can include `\hline`.

You can create a caption for the table with the command `\caption{Caption}`, where `Caption` should be replaced by your desired caption. You should also label your table such that you can reference it in the main text using `\label`.

5.2 Advanced Tables

Merging columns and rows in LaTeXis *slightly*, i.e. a whole lot, more complicated than in Excel.

Before merging columns, we need to know how many columns we would need in the most disaggregated row, and define the structure of the table normally. In the row where the merging of columns need to occur, we will need to use `\multicolumn{}{}{}`. The three brackets are used for:

- The number of columns that need to be combined
- How the content of the combined column should be aligned, and whether vertical line should be added
- The content

Hence, to create the following table:

Ranking of Universities		
Ranking	University	Location
1	University of Cambridge	UK
2	University of Oxford	UK

Table 2: Caption

we will need:

```
\begin{table}[h]
    \centering
    \begin{tabular}{|c|c|c|}
    \hline
    \hline
     \multicolumn{3}{|c|}{Ranking of Universities} \\
     \hline
     Ranking & University & Location \\
     \hline
     1 & University of Cambridge & UK \\
     2 & University of Oxford & UK
    \end{tabular}
    \caption{Caption}
    \label{tab:my_label}
\end{table}
```

Merging rows requires the `multirow` package. For example, let's say we want to merge the locations together,[8] we will need `\multirow{}{}{}`. The three brackets have a slightly different definition, especially the middle one:

- The number of rows that need to be combined
- The length of the combined combined cell, where you can explicitly define a length using `em` or leave it to be adjusted automatically by typing *
- The content

Ranking of Universities		
Ranking	University	Location
1	University of Cambridge	UK
2	University of Oxford	

Table 3: Caption

The full code for the above table:

```
\begin{table}[h]
    \centering
    \begin{tabular}{|c|c|c|c}
     \hline
    \hline
     \multicolumn{3}{|c|}{Ranking of Universities} \\
     \hline
     Ranking & University & Location \\
     \hline
     1 & University of Cambridge &  \multirow{2}{*}{\centering UK}  \\
     2 & University of Oxford & \\
    \end{tabular}
    \caption{Caption}
    \label{tab:my_label}
\end{table}
```

5.3 Figures

To include a figure in our document, we will need to first upload it to Overleaf. These can be `.png`, or `.jpg` etc. We can now insert it using `\begin{figure}`. Again, Overleaf will automatically supply the majority of the code:

```
\begin{figure}
    \centering
    \includegraphics{}
    \caption{Caption}
    \label{fig:my_label}
\end{figure}
```

We need to edit a couple of things. Firstly, next to `\begin{figure}`, we should define the positioning of the figure using a square bracket, similar to how we position tables.

[8] We can try merging the rankings together, but hey, we are better.

We should also modify the `\includegraphics{}` such that it now reads:

```
\includegraphics[]{}
```

In the square bracket, you can change the size of the figure. This can be done either by declaring the exact width and height of the picture, or declaring the scale of the figure relative to its original size, or setting its size relative to the line width. Some examples are:

```
\includegraphics[width=5cm, height=4cm]{}
\includegraphics[scale=1.2]{}
\includegraphics[width=\textwidth]{}
```

The third one is usually the most flexible, and you can also change it to say `0.9\textwidth` if needed.

In the curly bracket, you will need to type the name of the figure.

6 Bibliography

People tend to use a "BibTeX" to store their bibliography library. You can add any papers you have read into this BibTeX, and Overleaf will only include those you have cited in the document. There are multiple BibTeX software, such as Jabref and Zotero. You should maintain a library of all the papers you have read and found (perhaps) useful, and fill in your BibTeX library using DOI links or entering them manually.

Once you have this library, you can upload the `.bib` file to Overleaf. We will start citing papers by loading the `natbib` package in the preamble. Then, just before the `\end{document}`,[9] we should include the `.bib` file in

```
\bibliography{}
```

and define a bibliography style via

```
\bibliographystyle{}
```

There are many bibliography styles available, such as `chicago` and `apalike`.

Hence, the bottom of your document should look something like:

[9] This is only if you do not have appendix. If you do, these two commands should go before the appendix.

```
\bibliography{latexcourse}
\bibliographystyle{chicago}
\end{document}
```

where `latexcourse` is the `.bib` file name, and should be entered *without* the `.bib` extension, and `chicago` is the citation style for this document.

To cite papers, we will simply need to type

```
\cite{}
```

where the curly bracket contains the bibtex key for the paper. The bibtex key needs to be matching the one in the `.bib` in order for Overleaf to identify the article you are referring to. The citation will then appear with this format `author (year)`, e.g. `Baker (2016)`. It is also possible to do different citation styles, depending on the format of that particular line. Sometimes you might want to cite a paper in a bracket:

```
\citep{}
```

and it will look like `(author, year)`, e.g. `(Baker, 2016)`.

Other citation styles can be found here: `https://www.overleaf.com/learn/latex/Natbib_citation_styles`.

7 Brief Note about Beamer

In LaTeX, there is no PowerPoint. Instead, the LaTeXversion of slides is called "beamer". Most of the things we learned in these two sessions can be applied to beamer. We will first need to select a `theme` and a `color`. This matrix is helpful for you to select the style of your beamer: `https://hartwork.org/beamer-theme-matrix`.

Each slide of the beamer is created using

```
\begin{frame}{}

\end{frame}
```

where the first { } is where the frame title should go. Everything that needs to be printed on the slide will go between the `begin` and `end`.

There are two commands you may wish to learn for beamer. The first is creating blocks, and the second is for creating skip/jump buttons.

7.1 Blocks

Blocks are useful if you want to make your slides more readable, or if you want to highlight a particular statement/result.

```
\begin{block}{}

\end{block}
```

You will just need to put the title of the block in the curly brackets. Depending on the style of your slides, your block might have a different color from the rest of the slides.

7.2 Buttons

We often use buttons to jump between slides. There are different types of buttons:

```
\hyperlink{tag1}{\beamergotobutton{description 1}}
\hyperlink{tag2}{\beamerskipbutton{description 2}}
\hyperlink{tag3}{\beamerreturnbutton{description 3}}
```

where the `tag` part is basically how we used `ref`, and the `description` contains the words written on the button. For example, in the first line of code, we created a button that will go to (with a right arrow) the slide labelled as `tag1`, and the button reads "description 1".