

# A Guide to $\text{\LaTeX}$ and Sweave

A casual overview

Matt Lee, mlee8@g.harvard.edu,

Last updated: August 14, 2020

## Contents

<b>1</b>	<b>Overview</b>	<b>2</b>
1.1	What is $\text{\LaTeX}$ ?	2
1.2	What is R Sweave?	2
1.3	Reasons to use/not use $\text{\LaTeX}$	2
<b>2</b>	<b>Getting started</b>	<b>3</b>
2.1	Install $\text{\TeX}$	3
2.2	Install <code>knitr</code> and create a new Sweave file in RStudio	3
<b>3</b>	<b>Structuring a Sweave/<math>\text{\LaTeX}</math> File</b>	<b>6</b>
3.1	The Preamble	6
3.1.1	Style files	8
3.1.2	Other things I often do in the preamble	9
3.2	The Document Environment	9
<b>4</b>	<b>Including R code in your Sweave file</b>	<b>10</b>
<b>5</b>	<b>Typesetting math expressions</b>	<b>13</b>
5.1	Inline mode	13
5.2	Display mode	13
5.3	Common math expressions for PHS 2000 and beyond	15
5.3.1	For regression equations	15
5.3.2	For probability distributions, likelihood functions, etc	18
<b>6</b>	<b>Tables and figures</b>	<b>19</b>
6.1	Tables	19
6.2	Figures	20
<b>7</b>	<b>Lists</b>	<b>21</b>
<b>8</b>	<b>Overleaf</b>	<b>23</b>

# 1 Overview

This document provides some basic guidelines for getting up and running with L<sup>A</sup>T<sub>E</sub>X on your computer, with a specific emphasis of integrating with R Sweave. It is by no means all-inclusive, and the world of typesetting is quite large. Here, we'll illustrate some of the most common scenarios you might encounter when using L<sup>A</sup>T<sub>E</sub>X for your reports, presentations, etc. For another great overview, check out [this guide by Overleaf](#).

**A note:** There is a whole related world of using markdown and/or R markdown syntax to typeset your documents. That material will not be presented here, since its use is excellently detailed by Yihui Xie.<sup>1</sup>

**A disclaimer:** Proficiency in L<sup>A</sup>T<sub>E</sub>X is a lifelong journey, and this document should certainly not be used as a definitive guide to best practices. This is just a semi-structured write up of how I use Sweave and L<sup>A</sup>T<sub>E</sub>X in my own workflow, and should probably be read with a grain of salt.

The raw files for this guide are located at [this GitHub repository](#). I've also included the actual homework template I use for my classes, so feel free to use that as a starting place if desired.

## 1.1 What is L<sup>A</sup>T<sub>E</sub>X?

L<sup>A</sup>T<sub>E</sub>X (pronounced *LAH-* or *LAY-tek*) is a syntax language that is used to typeset documents. It's great for creating reports and presentations that look professional (even "pretty"!), and is particularly useful when typesetting mathematical expressions. L<sup>A</sup>T<sub>E</sub>X is also great for reproducibility, since there is a "script" file that documents how the output file (generally a .pdf) was created and styled.

## 1.2 What is R Sweave?

Sweave is a system that allows us to integrate R code and output with L<sup>A</sup>T<sub>E</sub>X typesetting. This has been particularly helpful for me for all of my homework assignments and working analysis documents, since I don't need to worry about remembering whether I've copied and pasted the most updated results into a word document. All I need to do is update my code (which is written directly into the Sweave .Rnw file) and re-compile it into my final PDF.

## 1.3 Reasons to use/not use L<sup>A</sup>T<sub>E</sub>X

- **pro:** L<sup>A</sup>T<sub>E</sub>X and Sweave are flexible, incredibly powerful, and provide a certain aesthetic that may be appealing depending on your own preferences.
- **pro:** L<sup>A</sup>T<sub>E</sub>X is the preferred method of typesetting used in any math-related field (e.g. biostatistics), so if you will be working in a related space it's probably worth it to learn.
- **con:** L<sup>A</sup>T<sub>E</sub>X can have a steep learning curve and sometimes takes longer relative to other word-processing software like Microsoft Word.

*In practice, my general workflow is to have a set template and style I can re-use for all of my homework assignments. This saves some time and prevents me from having to start from scratch every time. But YMMV and whatever works for you, works.*

---

<sup>1</sup><https://bookdown.org/yihui/rmarkdown/>

## 2 Getting started

### 2.1 Install T<sub>E</sub>X

T<sub>E</sub>X is the programming foundation of L<sup>A</sup>T<sub>E</sub>X. You will probably never have to interact with it directly. All of the commands and text you write in your .tex or .Rnw files get interpreted by T<sub>E</sub>X, which handles the production of the final document.

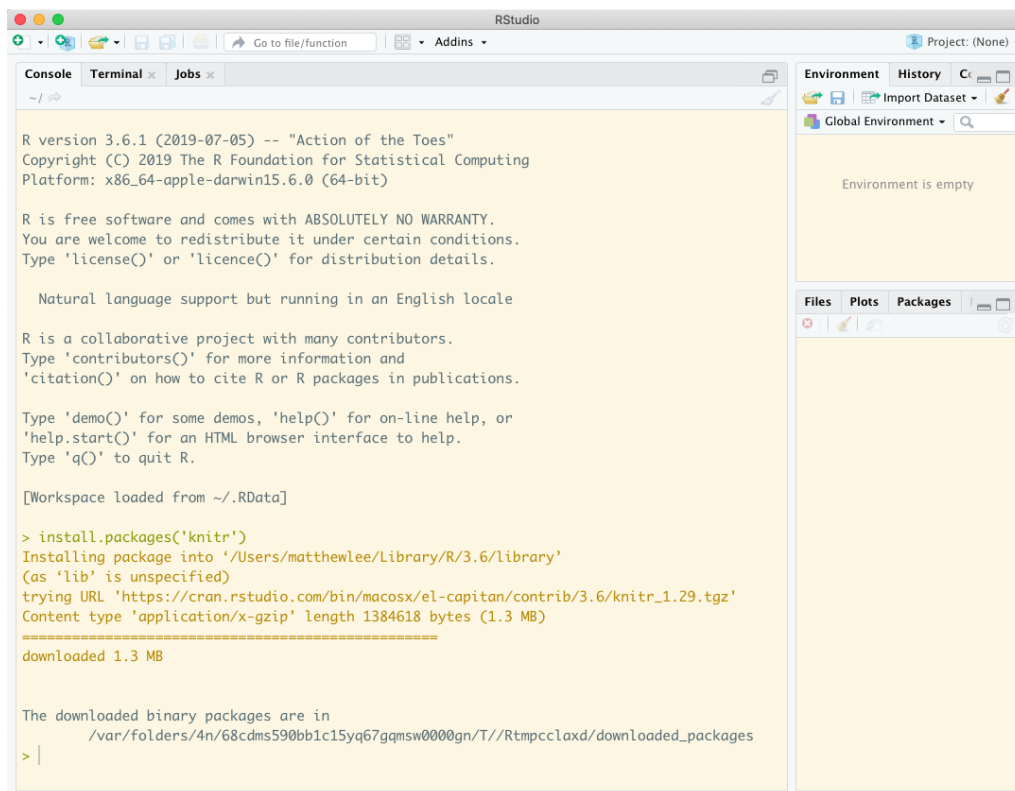
There are many distributions of T<sub>E</sub>X available, and you will need to download a version that works with your operating system (e.g. MacTeX for Mac OS and MiKTeX for Windows). You can look at the different options [here](#) and follow the specific directions for each. This is generally the most comprehensive way of installing T<sub>E</sub>X, and as a results can take quite a long time to download (e.g., MacTeX is roughly 4 Gb).

There is a separate option tailored for R users called [TinyTeX](#) that doesn't include every L<sup>A</sup>T<sub>E</sub>X package, but is significantly smaller in file size (150 Mb on Mac OS or 220 Mb on Windows). This might be useful if you only occasionally use L<sup>A</sup>T<sub>E</sub>X or don't have a ton of storage space on your computer.

### 2.2 Install knitr and create a new Sweave file in RStudio

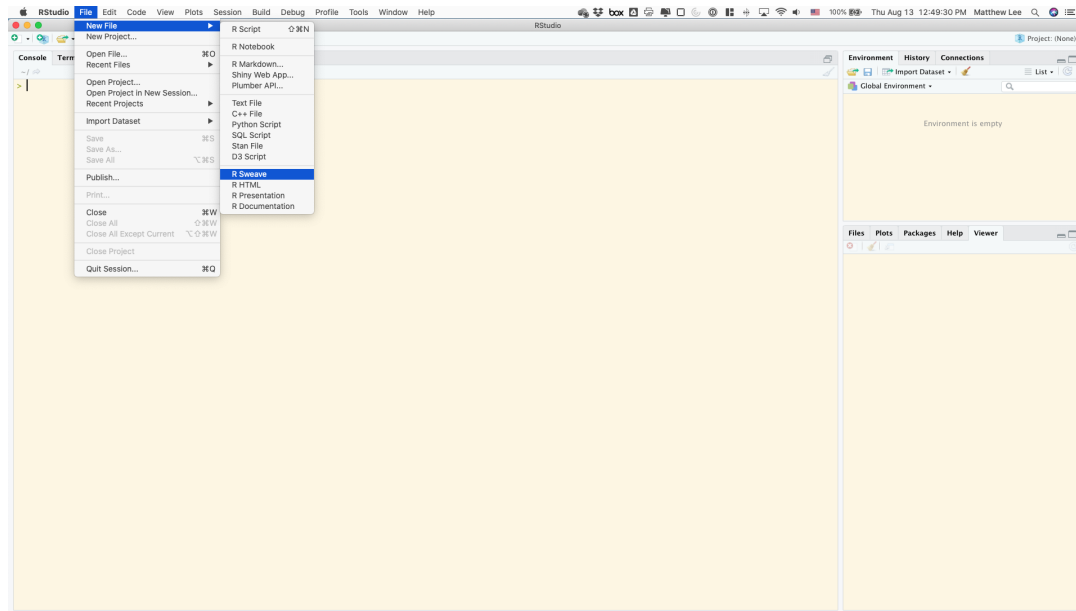
`knitr`<sup>2</sup> is an R package that can be used to “weave” or compile your Sweave .Rnw files. To install:

1. Open RStudio, and type `install.packages('knitr')` into the console window.



<sup>2</sup><https://yihui.org/knitr/>

2. Create your first Sweave document! In Rstudio, go to File → New File → R Sweave



3. Save your Sweave file, using command/ctrl + s or File → Save. You'll notice that when you go to this file in your directory, it has the extension .Rnw. This is the extension for all Sweave files. (.Rmd is for R markdown, .R is for pure R scripts).
4. Your Sweave file should look fairly empty, type in a message in between the `\begin{document}` and `\end{document}` tags, like below. Delete the `\SweaveOpts{concordance=TRUE}` line (this is only used when we knit using Sweave. In this case, we will knit using `knitr`). To tell RStudio that we would like to knit using `knitr`, we also need to include the line `% !Rnw weave = knitr` at the very beginning. Our file should look like:

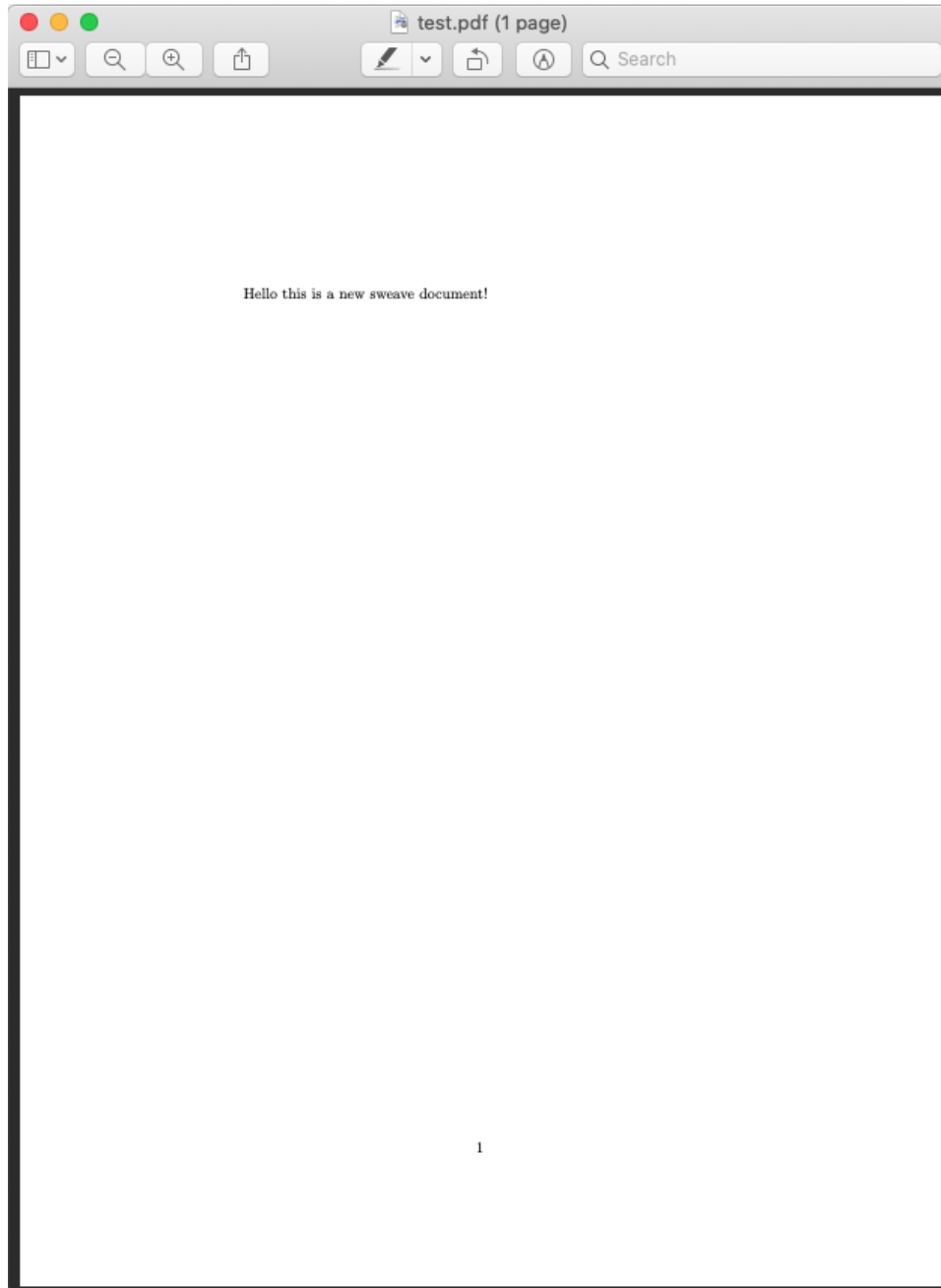
```
% !Rnw weave = knitr
\documentclass{article}

\begin{document}

Hello this is a new sweave document!

\end{document}
```

5. Click “Compile PDF” or hit command/ctrl + shift + k on your keyboard to compile your document and output it into a PDF file. You should get a pop-up that looks something like:



Doesn't look that great, but we will follow this basic procedure as we add to our Sweave file! You can also access some preferences for Sweave documents, like whether to preview the document after each compile, the default weave system (Sweave or knitr), and the default L<sup>A</sup>T<sub>E</sub>X program to use to generate the PDF file, by going to Rstudio → Preferences → Sweave.

### 3 Structuring a Sweave/L<sup>A</sup>T<sub>E</sub>X File

Every Sweave file is basically a L<sup>A</sup>T<sub>E</sub>X file with the added functionality of including R commands. As such, we need to follow L<sup>A</sup>T<sub>E</sub>X rules for structuring our document. Every Sweave file should have:

- A *Preamble*
- A *Document* environment

To comment out text, include the “%” symbol at the front of the comment.

*The general structure of a L<sup>A</sup>T<sub>E</sub>X file:*

```
\documentclass{}
...
\begin{document}
...
\end{document}
```

#### 3.1 The Preamble

The *preamble* portion of a document is where we declare the **document class** as well as any global packages, functions, or options we want to use for our document. Let’s take a look at an example that builds off of new Sweave file we created in the previous section:

```
% !Rnw weave = knitr
\documentclass{article}
\usepackage[]{/Users/matthewlee/style}
\usepackage[utf8]{inputenc}
\usepackage{caption}
\usepackage{geometry}

\geometry{letterpaper, portrait, margin=1in}

\pagestyle{fancy}
\rhead{A guide to \LaTeX}

\title{\vspace{2cm} \textbf{A Guide to} \LaTeX\ \textbf{and Sweave}}

\begin{document}

\maketitle

Hello this is a new sweave document!

\end{document}
```

We see that there's a whole bunch of lines that start with a backslash “\” and end with some text enclosed in curly braces “{}”. These are **functions (also called macros or commands)** in L<sup>A</sup>T<sub>E</sub>X, where the function *name* immediately follows the backslash, and function *arguments* are enclosed in the brackets. **Options** for functions are enclosed by square brackets “[ ]”, and can be omitted if you don't have any options to declare.

*You can specify options before or after the argument, but I like to include them before for consistency.*

Functions come from the L<sup>A</sup>T<sub>E</sub>X packages you've installed on your computer with your specific distribution (e.g. MacTeX, MiKTeX, TinyTeX), or can be defined manually using the `\newcommand{}` function.

The preamble of our Sweave file is anything that comes before the `\begin{document}` command. In the example above, our preamble is:

```
% !Rnw weave = knitr
\documentclass{article}
\usepackage[]{/Users/matthewlee/style}
\usepackage[utf8]{inputenc}
\usepackage{caption}
\usepackage{geometry}

\geometry{letterpaper, portrait, margin=1in}

\pagestyle{fancy}
\rhead{A guide to \LaTeX}

\title{\vspace{2cm} \textbf{A Guide to} \LaTeX\ \textbf{and Sweave}}
```

The **document class** function at the very top, `\documentclass{}` tells T<sub>E</sub>X what standard layout to use for the output. There are a variety of arguments available, but the most common are “article”, “report”, “book”, or “beamer” (for presentations). For homework assignments, I generally use the “article” document class and customize using different packages and options. `\documentclass{}` has several options including font size, paper size and columns. For example, the command below:

```
\documentclass[11pt, letterpaper, twocolumn]{article}
```

Sets the font size to 11pt, the output to letter paper size (8.5in x 11in), and splits the document into two columns. Another way to do this is with the `geometry` package's `\geometry{}` command. In the example above, I've set my formatting to letter paper, portrait orientation, and 1-inch margins with the following line:

```
\geometry{letterpaper, portrait, margin=1in}
```

In order to use this command, however, I need to tell T<sub>E</sub>X to load the package that it belongs to. This is another component of the **preamble**. Before I call `\geometry{}`, I use the `\usepackage{}` command to load the `geometry` package:

```
\usepackage{geometry}
```

### 3.1.1 Style files

If there are many packages you need to load, it is sometimes convenient to include them in a separate “style” file. This is a text file with the extension `.sty` that we call in our main document in order to load all the packages contained in our style file. In our example, I actually do this using the command:

```
\usepackage[]{/Users/matthewlee/style}
```

An excerpt from the actual style file, “style.sty” looks like:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%% TEXT/PAGE FORMATTING PACKAGES
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\usepackage{enumitem}
\usepackage{fancyhdr}
  \setlength{\headheight}{12pt}
\usepackage{float}
\usepackage{parskip}
  \parskip=8pt %% set parskip to 8pt
\usepackage[bottom]{footmisc}
\usepackage{changepage}
\usepackage{rotating}
\usepackage{fancyvrb}
\usepackage{lscapex}
  \allowdisplaybreaks
\usepackage{marginnote}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%% FIGURES
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\usepackage{graphicx}
\usepackage{subcaption}
\usepackage{wrapfig}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%% TABLES
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\usepackage{cellspace, boldline}
  \setlength\cellspacetoplimit{5pt}
  \setlength\cellspacebottomlimit{2pt}
```

*I save my full style.sty file (which contains packages and settings relevant to creating articles and reports) in my home directory and reference it most times I write up an assignment. In cases where I need something specific, I'll create a custom style.sty file and place it in the same folder that my .Rnw file is located*



```

\usepackage{threeparttable, tablefootnote}
\usepackage{booktabs}
\usepackage[table,xcdraw,dvipsnames]{xcolor}
\usepackage{longtable}
\usepackage[export]{adjustbox}
\usepackage{multicol}
\usepackage{multirow}
\usepackage{makecell}
\usepackage{tabularx}

%% Define column with grey background fill
\definecolor{Gray}{gray}{0.85}
\newcolumntype{a}{>{\columncolor{Gray}}C}

```

This just saves some space and makes reading the Sweave files a bit easier. However, you do have to be careful about package **conflicts**, which can occur if you load two packages that aren't compatible with each other. When you try to knit your file in these cases, you'll get an error. I've found that this is most common with packages related to references and bibliography settings.

### 3.1.2 Other things I often do in the preamble

In our example preamble, there are few other commands I use that help set up the layout of the document.

- `\pagestyle{fancy}`: Sets the page style to “fancy” (draws horizontal line separating the header from the main text, and lists the current section on the left side of the header) from the `fancyhdr` package.
- `\rhead{A guide to \LaTeX}`: Specifies that the right side header should include the text: “A guide to L<sup>A</sup>T<sub>E</sub>X”
- `\title{\vspace{2cm} \textbf{A Guide to} \LaTeX\ \textbf{and Sweave}}`: Sets the title to **A Guide to L<sup>A</sup>T<sub>E</sub>X and Sweave**

## 3.2 The Document Environment

The document environment consists of all the text in between `\begin{document}` and `\end{document}`, and contains all of the actual content that you want to print to the output PDF file. In our example, the document environment consists of a title and a single line of content:

```

\begin{document}

\maketitle

Hello this is a new sweave document!

\end{document}

```

Note that the `\maketitle` command can also print out values for “author” and “date” if

you define them in your preamble (i.e. using `\author{}` and `\date{}`). Let’s look at what our example looks like as a PDF. I’ve included some random example text and math just to illustrate how things appear for multiple pages:

A Guide to L<sup>A</sup>T<sub>E</sub>X and Sweave

Hello this is a new sweave document!

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, felis non sodales commodo, lectus velit ultricies augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egetas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

$$x = \frac{1}{n} \sum_{i=1}^{i=n} x_i = \frac{x_1 + x_2 + \dots + x_n}{n}$$

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, felis non sodales commodo, lectus velit ultricies augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egetas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

$$\int_0^{\infty} e^{-ax^2} dx = \frac{1}{2} \sqrt{\int_{-\infty}^{\infty} e^{-ax^2} dx} \int_{-\infty}^{\infty} e^{-ay^2} dy = \frac{1}{2} \sqrt{\frac{\pi}{a}}$$

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, felis non sodales commodo, lectus velit ultricies augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egetas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

$$\sum_{k=0}^{\infty} a_k q^k = \lim_{n \rightarrow \infty} \sum_{k=0}^n a_k q^k = \lim_{n \rightarrow \infty} a_n \frac{1 - q^{n+1}}{1 - q} = \frac{a_n}{1 - q}$$

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, felis non sodales commodo, lectus velit ultricies augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egetas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

1

A guide to L<sup>A</sup>T<sub>E</sub>X

tellus, aliquet aliquam, luctus a, egetas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = \frac{-p \pm \sqrt{p^2 - 4q}}{2}$$

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, felis non sodales commodo, lectus velit ultricies augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egetas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

$$\frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2} + \frac{\partial^2 \Phi}{\partial z^2} = \frac{1}{c^2} \frac{\partial^2 \Phi}{\partial t^2}$$

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, felis non sodales commodo, lectus velit ultricies augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egetas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

2

## 4 Including R code in your Sweave file

To include R code in your document, we need to tell Sweave how to recognize code vs. actual text. For Sweave, this is given by the syntax:

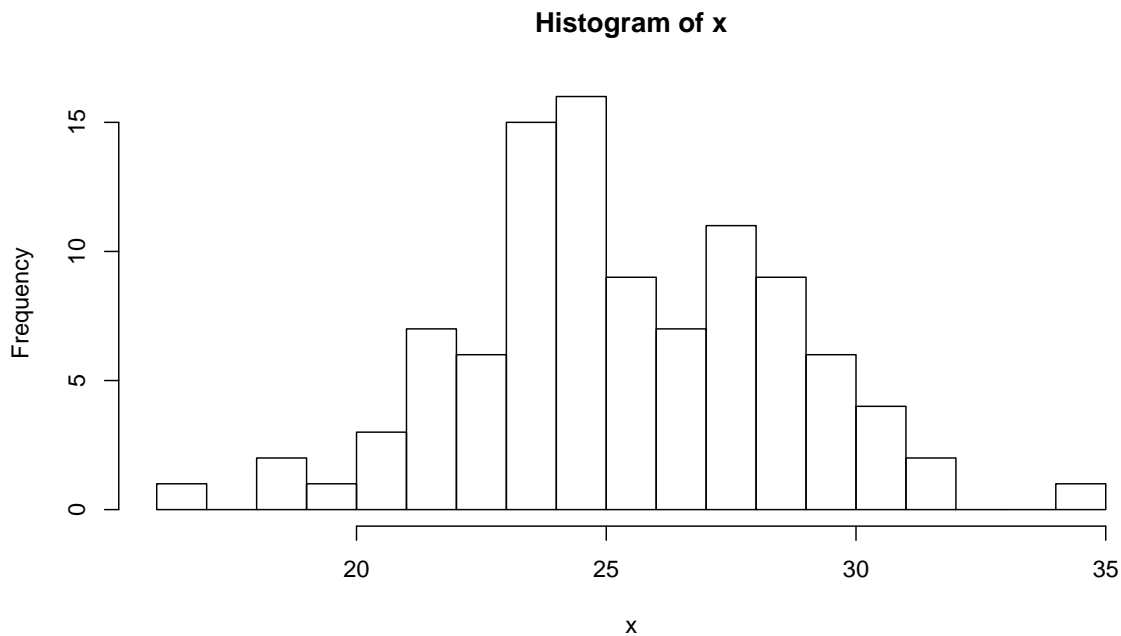
```
<<>>=
# Here is some R code

x <- rnorm(n = 100, mean = 25, sd = 3)
hist(x)

@
```

Where we enclose all of our R code inside the “<<>>=” and “@” symbols. In RStudio, the shortcut to insert these symbols automatically is **command/control + alt + i**. The code chunk above, when evaluated, looks like:

```
x <- rnorm(n = 100, mean = 25, sd = 3)
hist(x, breaks = 20)
```



We can also specify options for our code output by specifying them in between the double left and right inequality signs. For example:

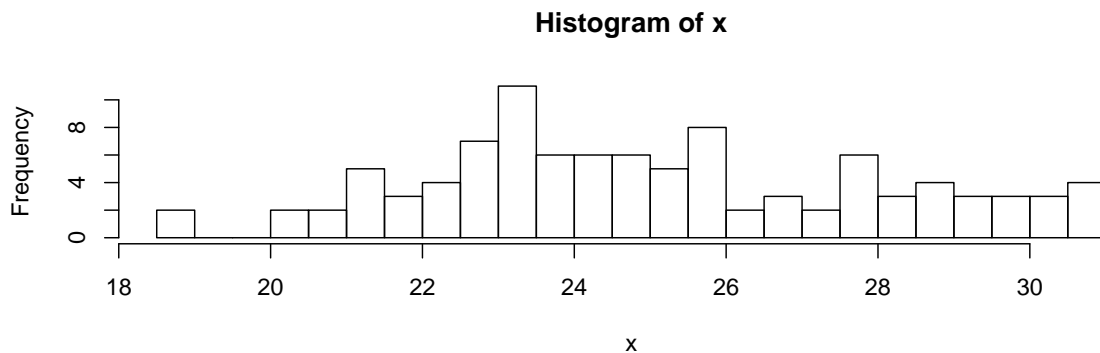
```
<<size='small', fig.height=3, fig.width=8.5, echo=TRUE, cache=TRUE>>=
# Here is some R code

x <- rnorm(n = 100, mean = 25, sd = 3)
hist(x)

@
```

Specifies that we want our code font size to be “small”, figure heights to be 3 inches, figure widths to be 8.5 inches, our code to be “echo”d (i.e. repeated back to us rather than hidden), and our objects to be cached. With these new options, our code chunk looks like:

```
x <- rnorm(n = 100, mean = 25, sd = 3)
hist(x, breaks = 20)
```



If we want to include R output within a line, for example as part of a results paragraph, we can use the command `\Sexpr{}` where we place whatever R command we want to include inside the curly braces. Let's say we want to report the mean value of `x`, rounded to two decimal places. We would have:

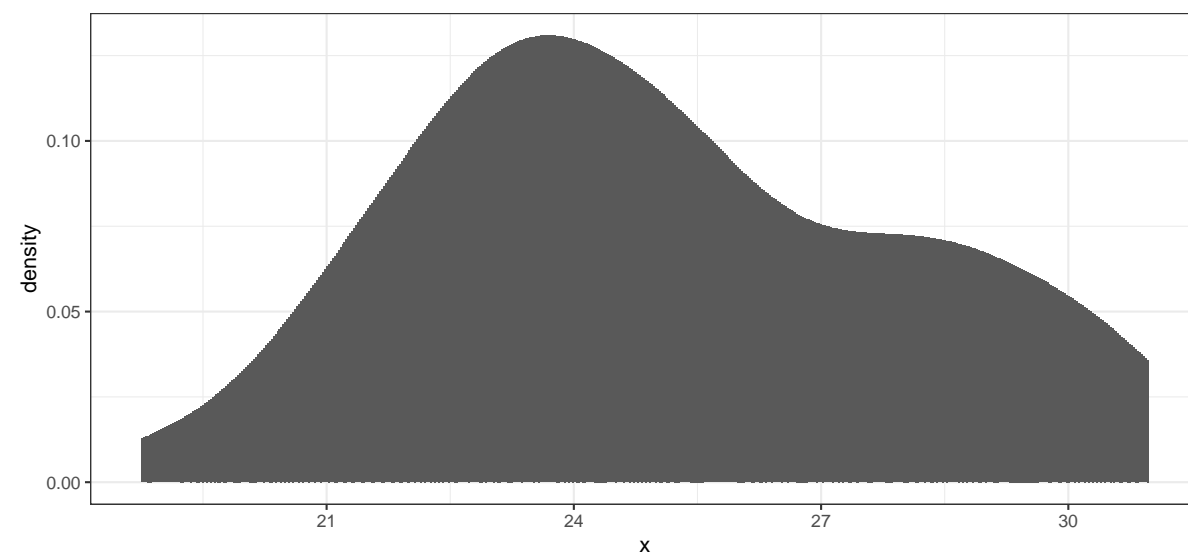
The mean of `x` is `\Sexpr{round(mean(x),2)}`

Show up as:

The mean of `x` is 25.09

We can also use additional R packages with Sweave, just like with R Markdown:

```
library(ggplot2)
x <- as.data.frame(x)
ggplot(x) +
  geom_bar(aes(x = x), stat = "density") +
  theme_bw()
```



*Why **S**`expr`?*  
*Because the **S** programming language was the precursor to what we now know as R.*  
***S**`expr` stands for “S expression”*

## 5 Typesetting math expressions

If you're typesetting anything with math, it's usually prudent to load the following packages either in your preamble or a separate style file:

- `\usepackage{amsmath}`
- `\usepackage{mathtools}`
- `\usepackage{amsfonts}`
- `\usepackage{hyperref}`

### 5.1 Inline mode

There are a number of ways to include math expressions in Sweave/L<sup>A</sup>T<sub>E</sub>X. To include an expression “inline”, i.e. in the line that you are writing in, surround the expression in single dollar signs ( $\dots$ ).

For example, “ $E(Y \mid X) = \beta_0 + \beta_1 X$  is a regression model” will be typset as “ $E(Y \mid X) = \beta_0 + \beta_1 X$  is a regression model”

### 5.2 Display mode

To typeset expressions in “display mode”, i.e. expressions on separate lines, our options become a bit more diverse. The most common options are:

For quick, centered, single expressions: enclose the expression in double dollar signs or backslash-brackets (i.e.  $\dots$  or  $[\dots]$ )<sup>3</sup>.

For example:

```

 $E(Y \mid X) = \beta_0 + \beta_1 X$ 
 $[E(Y \mid X) = \beta_0 + \beta_1 X]$ 

```

will render as:

$$E(Y \mid X) = \beta_0 + \beta_1 X$$

$$E(Y \mid X) = \beta_0 + \beta_1 X$$

For equations that are long and need to be split into multiple lines, or multiple equations that need to be aligned, we can use environments from the `amsmath` package.

- If you want to write a single equation on a single line, you can use the options from above or use the `equation` environment if you want to “tag” the equation with a number. When you tag equations, L<sup>A</sup>T<sub>E</sub>X will keep track of them for you and allow you to reference them if you specify a label using the `\label{}` command.

For example:

*I've never really noticed a big difference between using double-dollar signs or backslash-brackets, but people online seem to prefer the latter due to spacing issues.*

<sup>3</sup><https://tex.stackexchange.com/questions/503/why-is-preferable-to>

```
\begin{equation}
\label{myequation}
E(Y \mid X) = \beta_0 + \beta_1 X
\end{equation}
```

Will get rendered as:

$$E(Y \mid X) = \beta_0 + \beta_1 X \quad (1)$$

And we can reference this automatically, i.e. `\autoref{myequation}` (from the `hyperref` package) appears as [Equation 1](#).

- If we have a single expression that needs to be split across multiple lines, we can use the `split` environment **inside** the `equation` environment. To separate lines, use a double backslash “`\\`” at the end of each line. To align lines, include an ampersand right before the position you want to align by. For example, the command below will align the two lines by the “`=`” signs.

```
\begin{equation}
\begin{split}
\label{myequation2}
E(Y \mid X) &= \beta_0 + \beta_1 X \\
E(Y \mid W) &= \alpha_0 + \alpha_1 W
\end{split}
\end{equation}
```

Will be rendered as:

$$\begin{aligned} E(Y \mid X) &= \beta_0 + \beta_1 X \\ E(Y \mid W) &= \alpha_0 + \alpha_1 W \end{aligned} \quad (2)$$

- If we have a long expression that needs to be split across multiple lines, but we don’t care about aligning them, we can use the `multline` environment, which will align the first line to the left and the second to the right. Again, use a double backslash to separate lines. For example:

```
\begin{multline}
E(Y \mid X, W, L) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 \\
+ \alpha_0 + \alpha_1 W_1 + \alpha_2 W_2 + \alpha_3 W_3 + \alpha_4 W_4 \\
+ \delta_0 + \delta_1 L_1 + \delta_2 L_2 + \delta_3 L_3 + \delta_4 L_4
\end{multline}
```

Will be rendered as:

$$\begin{aligned} E(Y \mid X, W, L) &= \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 \\ &+ \alpha_0 + \alpha_1 W_1 + \alpha_2 W_2 + \alpha_3 W_3 + \alpha_4 W_4 + \delta_0 + \delta_1 L_1 + \delta_2 L_2 + \delta_3 L_3 + \delta_4 L_4 \end{aligned} \quad (3)$$

- If we have multiple equations (and we want to tag both separately), we can use the `align` environment. Again, separate lines with double backslashes and align using ampersands:

```
\begin{align}
E(Y \mid X) &= \beta_0 + \beta_1 X \\
E(Y \mid W) &= \alpha_0 + \alpha_1 W
\end{align}
```

$$E(Y \mid X) = \beta_0 + \beta_1 X \quad (4)$$

$$E(Y \mid W) = \alpha_0 + \alpha_1 W \quad (5)$$

- If we want to have multiple equations, but don't care about alignment, we can use the `gather` environment:

```
\begin{gather}
E(Y \mid X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \\
E(Y \mid W) = \alpha_0 + \alpha_1 W
\end{gather}
```

$$E(Y \mid X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \quad (6)$$

$$E(Y \mid W) = \alpha_0 + \alpha_1 W \quad (7)$$

For any environment where we don't want to tag the equation, we can add the “\*” symbol to the command name. For example:

```
\begin{gather*}
E(Y \mid X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \\
E(Y \mid W) = \alpha_0 + \alpha_1 W
\end{gather*}
```

$$E(Y \mid X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2$$

$$E(Y \mid W) = \alpha_0 + \alpha_1 W$$

## 5.3 Common math expressions for PHS 2000 and beyond

### 5.3.1 For regression equations

Greek letters:

```
\[
\alpha , \beta , \delta , \varepsilon , \gamma , \psi , \Psi , \Sigma , \gamma
\]
```

$$\alpha, \beta, \delta, \varepsilon, \gamma, \psi, \Psi, \Sigma, \gamma$$

To get bold math symbols, use the `\bm{}` command from the `bm` package:

```
\[
\bm{\alpha} , \bm{\beta} , \bm{\delta} , \bm{\varepsilon}
\]
```

$$\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\delta}, \boldsymbol{\varepsilon}$$

**Subscripts, superscripts:** Use “`_`” for subscripts, and “`^`” for superscripts. If you have more than one symbol that should go in the sub or super-script, enclose them in curly braces “`{}`” so L<sup>A</sup>T<sub>E</sub>X recognizes it as a group.

```
\[
\beta_1 , \beta_0 , \beta_{0, 1, 2, 3} , X^2 , X^{2x} , \sum_{i=1}^{N=n}
\]
```

$$\beta_1 , \beta_0 , \beta_{0,1,2,3} , X^2 , X^{2x} , \sum_{i=1}^{N=n}$$

### Other notation related to regression equations

To get the “conditional” symbol, use the `\mid` command:

```
\[
E(Y \mid X)
\]
```

$$E(Y \mid X)$$

To represent a regression equation as a sum of  $p$  linear predictors, use the `\sum` command:

```
\[
E(Y \mid X) = \beta_0 + \sum_{i=1}^p X_i \beta_i
\]
```

$$E(Y \mid X) = \beta_0 + \sum_{i=1}^p X_i \beta_i$$

To represent a regression equation using matrix bold math and text, use the `\textbf{}` command for any regular letter and `\bm{}` command for any bold math symbol:



```
\[
E(Y \mid X) = \textbf{X} \bm{\beta}
\]
```

$$E(Y \mid X) = \mathbf{X}\boldsymbol{\beta}$$

To include other text in a math equation, use the `\text{}` command inside the math environment:

```
\[
\text{logit}(\pi(x)) = \textbf{X} \bm{\beta}
\]
```

$$\text{logit}(\pi(x)) = \mathbf{X}\boldsymbol{\beta}$$

Note that L<sup>A</sup>T<sub>E</sub>X recognizes `\log` and `\exp`, so you don't need to write out `\text{log}`.

```
\begin{gather*}
\log(\pi(x)) = \textbf{X} \bm{\beta} \\\
\pi(x) = \exp(\textbf{X} \bm{\beta})
\end{gather*}
```

$$\begin{aligned}\log(\pi(x)) &= \mathbf{X}\boldsymbol{\beta} \\ \pi(x) &= \exp(\mathbf{X}\boldsymbol{\beta})\end{aligned}$$

Bracket matrices can be written using the `bmatrix` environment, which needs to be placed **inside** a math environment. Separate rows with “`\\`” and columns with “`&`”. We can also subscript or superscript matrices like any other object.

```
\[
\begin{bmatrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{bmatrix}_{(3 \times 3)}
\]
```

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}_{(3 \times 3)}$$

Parenthetical matrices can be written similarly, using the `pmatrix` environment.

```
\[
\begin{pmatrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{pmatrix}
\]
```

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

### 5.3.2 For probability distributions, likelihood functions, etc

For fractions, use the `\frac{}{}` command. The numerator goes in the first set of curly braces, and the denominator goes in the second:

```
\[
\frac{\exp(x)}{1 + \exp(x)}
\]
```

$$\frac{\exp(x)}{1 + \exp(x)}$$

For products (e.g. for the joint likelihood), use the `\prod` command:

```
\[
L(\lambda; \mathbf{x}) = \prod_{i=1}^n \frac{\lambda^{x_i} e^{-\lambda}}{x_i!}
\]
```

$$L(\lambda; \mathbf{x}) = \prod_{i=1}^n \frac{\lambda^{x_i} e^{-\lambda}}{x_i!}$$

For sums (e.g. for the joint *log* likelihood), use the `\sum` command. Calligraphic “l” can be written using the `\ell` command.

```
\[
\ell(\lambda; \mathbf{x}) = -n \lambda - \sum_{i=1}^n \log(x_i!)
+ \log(\lambda) \sum_{i=1}^n x_i
\]
```

$$\ell(\lambda; \mathbf{x}) = -n\lambda - \sum_{i=1}^n \log(x_i!) + \log(\lambda) \sum_{i=1}^n x_i$$

Integrals can be written using the `\int` command:

```
\[
\int_{x=1}^{x=25} f(x) dx
\]
```

$$\int_{x=1}^{x=25} f(x) dx$$

## 6 Tables and figures

### 6.1 Tables

There are many ways to typeset tables. Here, we present one method using the `tabular` environment. To start a table, place a `tabular` environment inside a `table` environment, which allows L<sup>A</sup>T<sub>E</sub>X to keep track of tables in your document. Within the `table` environment, you can also specify a label and caption, as well as horizontal alignment.

To specify where the table appears on the page, pass the *placement specifier* option to `\begin{table}`. For example, `\begin{table}[t]` places the table at the top of the page, `\begin{table}[b]` places the table at the bottom, `\begin{table}[h]` places the table **approximately** where you've written the table, and `\begin{table}[H]` places the table **exactly** where you've written the table.

*In most cases, I use the “H” option for table positioning, since I generally want my tables to appear directly next to the relevant text.*

Within the `table` environment, the actual contents of your table will go in the `tabular` environment. Column alignment is specified in a second set of curly braces directly following the `\begin{tabular}` command, and you must specify “l” (left), “r” (right), or “c” (centered) for each column in your table. In the `tabular` environment, separate rows with `\\` and columns with “&”.

I like to have horizontal borders in my tables by using the `\toprule`, `\midrule`, and `\bottomrule` commands from the `booktabs` package (i.e. `\usepackage{booktabs}`), but there are many alternatives out there you can explore.

```
\begin{table}[H]
\centering % sets horizontal alignment to ``centered''
\label{tab:mytable1}
\caption{My Table}
\begin{tabular}{lcc} % ``lcc'' specifies left for col1, center for cols 2/3
\toprule % toprule for very top of table
left aligned & center & center \\
\midrule % midrule following table headers
1 & 2 & 3 \\
4 & 5 & 6 \\
\bottomrule % bottomrule for very bottom of table
\end{tabular}
\end{table}
```

Table 1: My Table

left aligned	center	center
1	2	3
4	5	6

## 6.2 Figures

To include a figure, use the `\includegraphics{}` command within a `figure` environment. We can use the same positioning options as we did with tables (i.e. `t`, `b`, `h`, or `H`), and include labels and captions as well.

Within the `figure` environment, specify the path to the image using the `\includegraphics` command, which takes in as its argument the file path (not including the file extension) **relative to where your .Rnw file is located**. In the example below, I’ve saved an image in a folder called “img”, which is located in the same folder as my .Rnw file. A helpful option to this command is `width`, which allows you to resize your images – I most often specify this as a multiple of `\textwidth`, which is a L<sup>A</sup>T<sub>E</sub>X variable that returns the width of the text area on a page.

```
\begin{figure}[H]
\centering % sets horizontal alignment to ``centered''
\caption{A figure}
\label{fig:myfig}
\includegraphics[width = \textwidth]{img/puppies-cover-1280x720}
\end{figure}
```

Figure 1: A figure



Scaling to 0.3 times the textwidth:

```
\begin{figure}[H]
\centering % sets horizontal alignment to ``centered''
\caption{A figure, scaled}
\label{fig:myfig}
\includegraphics[width = 0.3\textwidth]{img/puppies-cover-1280x720}
\end{figure}
```

Figure 2: A figure, scaled



## 7 Lists

Lists in L<sup>A</sup>T<sub>E</sub>X can be written without any additional package, but I've found that the `\enumitem` package provides some nice customization (e.g. in changing the formatting of the bullets, symbols, etc.)

To start a **bulleted** list, use the `itemize` environment and start each new item with the `\item` command:

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

- Item 1
- Item 2
- Item 3

To start a **numbered** list, use the `enumerate` environment, again starting each new item with the `\item` command.

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

1. Item 1

2. Item 2
3. Item 3

To change the style of the list (for instance to use alphabetic letters instead of numbers, make specifications to the “label” option of `enumerate`): <sup>4</sup>

```
\begin{enumerate}[label = (\alph*)]
\item Item 1
\item Item 2
\item Item 3
\end{enumerate}
```

- (a) Item 1
- (b) Item 2
- (c) Item 3

```
\begin{enumerate}[label = (\roman*)]
\item Item 1
\item Item 2
\item Item 3
\end{enumerate}
```

- (i) Item 1
- (ii) Item 2
- (iii) Item 3

Last, we can also create nested lists by nesting our `enumerate` or `itemize` environments:

```
\begin{enumerate}[label = \alph*.]
\item Item 1
  \begin{itemize}
    \item Nested
  \end{itemize}
\item Item 2
\item Item 3
\end{enumerate}
```

- a. Item 1
  - Nested
- b. Item 2
- c. Item 3

---

<sup>4</sup>There are many options here, see: <https://www.overleaf.com/learn/latex/lists>

## 8 Overleaf

If you aren't doing anything related to R, and just want to typeset text using L<sup>A</sup>T<sub>E</sub>X, another neat option is to do all of your writing on [Overleaf](#), an online platform that allows you to see your changes in real time, collaborate with others in a Google-docs style editing system, and backup your projects automatically using auto-save and version control features.