

Introduction to L^AT_EX

Casey Crisman-Cox

7 August 2023

Contents

1	Installation and getting started	3
1.1	Installing on your computer	4
1.2	Installing an editor	4
1.3	First document	4
1.4	Typing special characters	5
1.5	Basic Tweaks	5
1.6	Items, Numbers, and Descriptions	7
1.7	Getting help	9
2	Tables, figures, footnotes, and more	10
2.1	Tables and Figures	10
2.2	Footnotes and endnotes	14
2.3	Abstracts and titles	15
2.4	Sections	17
2.5	Advanced tweaks	19
2.6	Putting it all together	20
2.6.1	SyncTeX	20
3	Writing maths	21
3.1	Basic math	21
3.2	Other fancy equations	24
3.3	Matrices	25
3.4	Theorems and proofs	25
3.5	Fancy fonts and other notations	26
3.6	Your turn	26
4	References using bibTeX	27
4.1	Organizing references with JabRef	27
4.2	Setting up your document	28
4.3	Citing commands	29
4.4	Other fancy characters (accents)	31
5	Making presentations with beamer	32
5.1	Theme and color adjustments	32
5.2	Frames	33
5.3	Appendix slides and references	35

6	Producing simple images with TikZ	36
6.1	Basic Lines	36
6.1.1	Adding Text	39
6.2	Curves	43
6.2.1	Functions	44
6.3	Drawing games	45

Chapter 1

Installation and getting started

Buckle up, true believers! We're setting off on a while ride to teach you how to use \LaTeX to prepare documents. While this may seem like a daunting prospect now, I assure you that with patience and practice you'll be pros before you know it. However, the real question you're likely asking yourself now is: Why in the Sam Hill would I use \LaTeX ? Microsoft Word was good enough for my grandpappy, why ain't it good enough for me? The answer to this question is that \LaTeX provides a few real advantages over office suite software, and once you get used to it, you'll never go back.

List of things that \LaTeX is good for:

- It's the standard bearer for any kind of technical or academic writing. This is especially true for anything involving math.
- It's easy to export statistical (regression) tables in ways that \LaTeX can use.
- It has an easy to use reference management system. It makes citations very easy for your modern academic.
- Files are all text and pdf. This means that there are no compatibility issues across versions or operating systems. It also means that the raw documents are small and easy to load (no crashing when you try to open big documents).

As we'll also discover there are benefits in terms of the kinds of documents it can prepare. It is good for writing:

- Articles
- Slide shows
- Books
- Posters
- Curricula vitae
- Letters

To summarize, this is the beginning of your association with a document preparation system that will change how you do work.

1.1 Installing on your computer

An additional advantage of \LaTeX is that it's easy to install and it's free to do so. It's already on all the Star Lab machines so we wouldn't be doing this. This is for your personal machines.

Linux Use your package manager to install \TeX Live. On a Ubuntu system that would look like this

```
sudo apt-get install texlive-full
```

and you're done.

Mac On a Mac OS X machine you'll want to use \MacTeX . You can download the installer here <https://tug.org/mactex/>. Download it, double click and you should be good to go.

Windows On Windows the standard is \MiKTeX . Download it from <https://miktex.org/download> and run.

1.2 Installing an editor

The next thing you'll want once you have the \LaTeX engine install is an editor for it. On this point there are literally dozens of options. You should feel free to shop around on your own time and try different editors. Your final choice will be a personal decision that reflects you are as an individual. I like \TeX studio. You can download it from <http://www.texstudio.org/>. Just scroll down until you find your operating system and download the correct version. If you use linux you should either download from the website or check your software repository. Again on Ubuntu this looks like

```
sudo apt-get install texstudio
```

easy-peasy.

1.3 First document

Okey-dokey, we've got a \LaTeX engine, we've got an editor, time to actually write something. Before we start, let's create a special folder for this course because we'll have a lot of files that we'll use over the next few days (if you're using a star lab machine, make sure to use your z drive.) Go ahead and open up \TeX studio and type the following into it.

```
\documentclass{article} %tell it what kind of thing we're writing.
%Use article unless you're doing something fancy
\begin{document}
This is my first document using \LaTeX.
\LaTeX\ is fun.
‘‘And now the time as come,’’ the Walrus said, ‘‘to talk of many things.
Of shoe and ships and sealing wax, of cabbages and kings.’’
```

```
This is a new paragraph
\end{document}
```

Now before creating the document you should first save the file, let's call it "Testing.tex". Now you can compile it by clicking on tools > build and view (or just hit F1 if using T_EXstudio). A few things to note here. First, % is how we generate comments. These are remarks that we want to leave for ourselves or others if the source file, but that we don't want to show up in the actual document. You can imagine having part of a document that you're not sure if you want to include. Rather than move it or delete it, you can simply comment it away and reinstate it as needed.

Another thing to note is that the only text to appear in the document is what is between the commands begin and end document. This is no coincidence. Everything above \begin{document} is about setting up the document. In this case it just means telling it we want to write an article, but we'll add things here as we go. Everything below \end{document} is just ignored by the compiler. This another place to hide text you're not sure if you want to include.

Looking back at our example you'll also notice that white space doesn't matter much. Extra spaces and line breaks don't have an effect on the output. This allows you some liberty to structure your text in the source file in a way that is most readable to you.

Another thing to notice is the use of quotation marks. This is something that often get's novice users. The double quote mark " (shift-') produces quotes that only curve one direction (same for single quotes from that key).

Table 1.1: Typing quotes

Command	Result	Note
"Cabbages and Kings"	"Cabbages and Kings"	Double quotes, wrong direction
'Cabbages and Kings'	'Cabbages and Kings'	Single quotes, wrong direction
‘Cabbages and Kings’	‘Cabbages and Kings’	Single quotes, right direction
‘‘Cabbages and Kings’’	“Cabbages and Kings”	Double quotes, right direction

1.4 Typing special characters

Also take note that all L^AT_EX commands start with \. The backslash is an important character for telling L^AT_EX that we are giving it orders. Likewise the curly brackets are special and tell it to capture the text inside to them to be given to the command. Other special characters include %, &, ^, -, ~, #. We discussed % already as a way to include comments. Many of the others will come up later. However, I will say that the way to type most of these characters just attach a backslash to them. the exceptions to this are the caret and the tilde which require an argument (because they are accent marks). We will discuss them later in the course.

Also note that the command \LaTeX writes L^AT_EX in the fancy way. There are lots of commands like this that do special things. We'll pick up a lot more of these as we go.

1.5 Basic Tweaks

Without any tampering the L^AT_EX article uses single spacing, 10 point font, and the margins seem a bit large. So the first thing we're going to do is tweak those to make things seem a

Table 1.2: Typing special characters

Character	Command
%	\%
&	\&
^	\^{}
_	_
~	\~{}
#	\#

little more friendly.

Font size It is easy to use 10, 11, or 12 point font your papers. To use these sizes you adjust our original example as follows

```
\documentclass[11]{article} %11 is 11 pt font, can be 10, 11, or 12.
\begin{document}
This is my first document using \LaTeX.
\LaTeX\ is fun.
‘‘And now the time as come,’’ the Walrus said, ‘‘to talk of many
things. Of shoe and ships and sealing wax, of cabbages and kings
.’’

This is a new paragraph.
\end{document}
```

Margins To adjust the margins we will use our first package. Packages are add-ons to L^AT_EX for increased functionality. For margins we will use the `geometry` package.

```
\documentclass[11]{article} %11 is 11 pt font, can be 10, 11, or 12.
\usepackage[top=1in,bottom=1in,left=1.25in,right=1.25in]{geometry}
\begin{document}
This is my first document using \LaTeX.
\LaTeX\ is fun.
‘‘And now the time as come,’’ the Walrus said, ‘‘to talk of many
things. Of shoe and ships and sealing wax, of cabbages and kings
.’’

This is a new paragraph
\end{document}
```

This allows you to set margins for each side of the paper.

Spacing Often time we like to read papers with double or 1.5 spacing. It just makes it easier for your audience (including yourself) to read. We will use another package for this.

```
\documentclass[11]{article} %11 is 11 pt font, can be 10, 11, or 12.
\usepackage[top=1in,bottom=1in,left=1.25in,right=1.25in]{geometry}
\usepackage{setspace}
```

```

\doublespacing %for 1.5 spacing use \onehalfspacing
\begin{document}
This is my first document using \LaTeX.
\LaTeX\ is fun.
‘‘And now the time as come,’’ the Walrus said, ‘‘to talk of many
  things. Of shoe and ships and sealing wax, of cabbages and kings
  .’’

This is a new paragraph
\end{document}

```

1.6 Items, Numbers, and Descriptions

The next thing we’ll introduce is items, numbering, and description. These are all ways to create lists in your document.

```

\documentclass[11]{article} %11 is 11 pt font, can be 10, 11, or 12.
\usepackage[top=1in,bottom=1in,left=1.25in,right=1.25in]{geometry}
\usepackage{setspace}
\doublespacing %for 1.5 spacing use \onehalfspacing
\begin{document}
This is my first document using \LaTeX.
\LaTeX\ is fun.
‘‘And now the time as come,’’ the Walrus said, ‘‘to talk of many things.
  Of shoe and ships and sealing wax, of cabbages and kings.’’

This is a new paragraph. Here is a list of colors
\begin{itemize}
  \item Red
  \item Blue
  \item Green
\end{itemize}
\end{document}

```

If we wanted to do it with numbers instead:

```

\documentclass[11]{article} %11 is 11 pt font, can be 10, 11, or 12.
\usepackage[top=1in,bottom=1in,left=1.25in,right=1.25in]{geometry}
\usepackage{setspace}
\doublespacing %for 1.5 spacing use \onehalfspacing
\begin{document}
This is my first document using \LaTeX.
\LaTeX\ is fun.
‘‘And now the time as come,’’ the Walrus said, ‘‘to talk of many things.
  Of shoe and ships and sealing wax, of cabbages and kings.’’

This is a new paragraph. Here is a list of colors
\begin{enumerate}
  \item Red

```



```

\item Blue
\item Green
\end{enumerate}
\end{document}

```

We can nest lists and switch them up

```

\documentclass[11]{article} %11 is 11 pt font, can be 10, 11, or 12.
\usepackage[top=1in,bottom=1in,left=1.25in,right=1.25in]{geometry}
\usepackage{setspace}
\doublespacing %for 1.5 spacing use \onehalfspacing
\begin{document}
This is my first document using \LaTeX.
\LaTeX\ is fun.
‘‘And now the time as come,’’ the Walrus said, ‘‘to talk of many things.
Of shoe and ships and sealing wax, of cabbages and kings.’’

```

This is a new paragraph. Here is a list of colors

```

\begin{enumerate}
\item Red
\begin{itemize}
\item Pink
\item Magenta
\item Crimson
\begin{enumerate}
\item Brick
\end{enumerate}
\end{itemize}
\item Blue
\item Green
\end{enumerate}
\end{document}

```

Descriptions are a little different. They take keywords for their headers

```

\documentclass[11]{article} %11 is 11 pt font, can be 10, 11, or 12.
\usepackage[top=1in,bottom=1in,left=1.25in,right=1.25in]{geometry}
\usepackage{setspace}
\doublespacing %for 1.5 spacing use \onehalfspacing
\begin{document}
This is my first document using \LaTeX.
\LaTeX\ is fun.
‘‘And now the time as come,’’ the Walrus said, ‘‘to talk of many things.
Of shoe and ships and sealing wax, of cabbages and kings.’’

```

This is a new paragraph. Here is a list of colors

```

\begin{description}
\item[Reds] Some reds include
\begin{itemize}
\item Pink

```

```

\item Magenta
\item Crimson
  \begin{itemize}
    \item Brick
  \end{itemize}
\end{itemize}
\item[Blue]
\item[Green]
\end{description}
\end{document}

```

1.7 Getting help

The last thing we'll talk about today is about debugging and getting help. Getting help is as easy as using Google to perform searches that look like “how to do [blank] in latex.” Good sources for help also include

- <http://mirror.unl.edu/ctan/info/symbols/comprehensive/symbols-a4.pdf>. This website is a fairly comprehensive list of symbols that can be produced in L^AT_EX and how to produce them.
- <http://detexify.kirelabs.org/classify.html>. An interactive website where you draw the symbol you want to produce and it tries to find it for you.
- <https://en.wikibooks.org/wiki/LaTeX>. A good source with examples on doing ordinary things

If you can't find it on the internet you can ask a Star Lab Fellow for help.

If you're having trouble compiling the best advice I can give you is comment out large parts of the document. Keep doing that until it compiles. Once you have something that works, slowly reintroduce the rest of the document until it fails. This will tell you where the problem is and you can investigate from there.

Chapter 2

Tables, figures, footnotes, and more

Today we're going to take care of most of the things we need to produce papers in L^AT_EX.

2.1 Tables and Figures

Table and figure are important ways to convey information. The regression table is a staple of any empirical paper and figures are often great for conveying theoretical or empirical results. Let's start with our a new document.

```
\documentclass[11]{article} %11 is 11 pt font, can be 10, 11, or 12.
\usepackage[top=1in,bottom=1in,left=1.25in,right=1.25in]{geometry}
\usepackage{setspace}
\onehalfspacing
\begin{document}
This paper examines important scientific questions.
It does so in a super rigorous and presents results so well-identified
    that they can't grab a bite without being spotted.
We address all the important concerns with our results and can say with
    absolute certainty that we have created the last word on this topic.

In this paragraph we want to say just a little more about what a big deal
    the paper is.
This paper advances all of science by about 25 years.
In fact, we have produced 200\% more science than anyone else writing on
    this topic.

\end{document}
```

Now let's say that this paper needed a table. It would look something like this

```
\documentclass[11]{article} %11 is 11 pt font, can be 10, 11, or 12.
\usepackage[top=1in,bottom=1in,left=1.25in,right=1.25in]{geometry}
\usepackage{setspace}
\onehalfspacing
```

```

\begin{document}
This paper examines important scientific questions.
It does so in a super rigorous and presents results so well-identified that
    they can't grab a bite without being spotted.
We address all the important concerns with our results and can say with
    absolute certainty that we have created the last word on this topic.

In this paragraph we want to say just a little more about what a big deal
    the paper is.
This paper advances all of science by about 25 years.
In fact, we have produced 200\% more science than anyone else writing on
    this topic.


\begin{table}[h]
    \centering
    \caption{Table of stuff}
    \label{tab:Stuff}
    \begin{tabular}{rc|l}
        \hline
        Input 1 & Input 2 & Sum \\ \hline
        5      & 10      & 15 \\
        2      & 9       & 11 \\ \hline
    \end{tabular}
\end{table}

Table \ref{tab:Stuff} shows just all the cool science we do.
\end{document}

```

So let's break this down one step at a time. The entire table is between the `\begin{table}` ... `\end{table}` commands. The `[h]` tells it we want the table placed "here." This is just a suggestion as the software will place it where it thinks is best. Other options include `b` for bottom, `t` for top, and `p` for separate page. We use the centering command to center the table. I then gave it a caption and a label. You should always give your tables a caption because that tells the reader what's going on in the table. The label is for you to use to refer to the table. We use the `\ref` command to refer to the table in the text. This means that we never have to manually number or adjust the numbers. \LaTeX does this for us. Quick note is that we can move caption and label to below the tabular commands and this moves the caption to below the table, *but* label has to come after caption.

After this we get to the `tabular` section. This is where the actual table happens. The first thing after we open with `\begin{tabular}` is we have `rc|l` this tells the software two things. First, it tells it there will be three columns (number of letters is the number of columns). The second thing is how to justify the text within each column with `r`, `c`, and `l` meaning right, center, and left, respectively. The pipe, `|` says to put a vertical line between these two columns. We use `\hline` to create horizontal lines after a row, `\\` to end rows, and `&` to separate columns. Everything else is just the text.

Something you may want to do someday is merge cells together. To do that we can just use the multicolumn command.

```
\documentclass[11]{article} %11 is 11 pt font, can be 10, 11, or 12.
\usepackage[top=1in,bottom=1in,left=1.25in,right=1.25in]{geometry}
\usepackage{setspace}
\onehalfspacing
\begin{document}
This paper examines important scientific questions.
It does so in a super rigorous and presents results so well-identified that
    they can't grab a bite without being spotted.
We address all the important concerns with our results and can say with
    absolute certainty that we have created the last word on this topic.

In this paragraph we want to say just a little more about what a big deal
    the paper is.
This paper advances all of science by about 25 years.
In fact, we have produced 200\% more science than anyone else writing on
    this topic.
```

```
\begin{table}[h]
  \centering
  \caption{Table of stuff}
  \label{tab:Stuff}
  \begin{tabular}{rc|l}
    \hline
    Input 1 & Input 2 & Sum \\ \hline
    5        & 10        & 15 \\
    2        & 9         & 11 \\ \hline
    \multicolumn{3}{l}{here's a remark.} \\
  \end{tabular}
\end{table}
```

Table \ref{tab:Stuff} shows just all the cool science we do.

```
\end{document}
```

As you can see multicolumn takes three arguments. The first is how many columns to take up, the second is the justification, and the third is the text.

Figures work a lot like tables. To include them we will add another package: graphicx.

```
\documentclass[11]{article} %11 is 11 pt font, can be 10, 11, or 12.
\usepackage[top=1in,bottom=1in,left=1.25in,right=1.25in]{geometry}
\usepackage{graphicx} %new package
\usepackage{setspace}
\onehalfspacing
\begin{document}
This paper examines important scientific questions.
```

It does so in a super rigorous and presents results so well-identified that they can't grab a bite without being spotted.

We address all the important concerns with our results and can say with absolute certainty that we have created the last word on this topic.

In this paragraph we want to say just a little more about what a big deal the paper is.

This paper advances all of science by about 25 years.

In fact, we have produced 200% more science than anyone else writing on this topic.

```
\begin{table}[h]
  \centering
  \caption{Table of stuff}
  \label{tab:Stuff}
  \begin{tabular}{rc|l}
    \hline
    Input 1 & Input 2 & Sum \\ \hline
    5       & 10      & 15 \\
    2       & 9       & 11 \\ \hline
    \multicolumn{3}{l}{here's a remark.} \\
  \end{tabular}
\end{table}
```

Table \ref{tab:Stuff} shows just all the cool science we do. Figure \ref{fig:plots} graphs some other stuff.

```
\begin{figure}[h]
\includegraphics[width=\columnwidth]{3dPlots.pdf}
\caption{Some plots}
\label{fig:plots}
\end{figure}

\end{document}
```

There are a few things to note here. First, figures start the same as tables with a begin and an end and they use the same placement indicators. To actually include the figure we use the command `\includegraphics[options]{name}`. The options in this case was to tell the software that we want the picture to take the entire width of the page. We can scale that by saying this like `width=0.5\columnwidth` to get it to take half the page. Or we can scale the whole image by a fraction by using `scale=.9` as in `\includegraphics[scale=.9]{name}`. The file of the image goes in the brackets. It needs to be the fixed or relative path to the image. In this case we need the image to be in the same folder we're using. There should be no spaces in the name of the image either. Like with tables we caption and label to identify them.

2.2 Footnotes and endnotes

Frequently, we want to have footnotes to explain things that are important but would interrupt the flow of the text if we included them in the paragraph. For this we use footnotes. This is pretty straightforward, going back to our example.

```
\documentclass[11]{article} %11 is 11 pt font, can be 10, 11, or 12.
\usepackage[top=1in,bottom=1in,left=1.25in,right=1.25in]{geometry}
\usepackage{graphicx} %new package
\usepackage{setspace}
\onehalfspacing
\begin{document}
This paper examines important scientific questions.
It does so in a super rigorous and presents results so well-identified
    that they can't grab a bite without being spotted.
We address all the important concerns with our results and can say with
    absolute certainty that we have created the last word on this topic.

In this paragraph we want to say just a little more about what a big deal
    the paper is.
This paper advances all of science by about 25 years.
In fact, we have produced 200\% more science than anyone else writing on
    this topic.
```

```
\begin{table}[h]
  \centering
  \caption{Table of stuff}
  \label{tab:Stuff}
  \begin{tabular}{rc|l}
    \hline
    Input 1 & Input 2 & Sum \\ \hline
    5       & 10       & 15 \\
    2       & 9        & 11 \\ \hline
    \multicolumn{3}{l}{here's a remark.} \\
  \end{tabular}
\end{table}
```

Table \ref{tab:Stuff} shows just all the cool science we do. Figure \ref{fig:plots} graphs some other stuff.

```
\begin{figure}[h]
\includegraphics[width=\columnwidth]{3dPlots.pdf}
\caption{Some plots}
\label{fig:plots}
\end{figure}
```

```

That figure sure was a pretty.\footnote{The prettiness of the figure was
  rated as 7.9 on the princess beauty scale.}

\end{document}

```

Of course if you really hate your reader (or your journal requires end notes) we can just use `\endnote{}` instead, but this requires you to load the package `endnotes`.

2.3 Abstracts and titles

Every academic paper you write should have an abstract and a title page. So let's get into that.

```

\documentclass[11]{article}
\usepackage[top=1in,bottom=1in,left=1.25in,right=1.25in]{geometry}
\usepackage{graphicx}
\usepackage{setspace}
\onehalfspacing

% % New % %
\title{On the Making of Science}
\author{Bullwinkle J. Moose\footnote{Ph.D. Candidate, Moose Institute for
  Toetapping.}}
\date{\today}
% % % % %

\begin{document}

\maketitle
\newpage
This paper examines important scientific questions.
It does so in a super rigorous and presents results so well-identified that
  they can't grab a bite without being spotted.
We address all the important concerns with our results and can say with
  absolute certainty that we have created the last word on this topic.

In this paragraph we want to say just a little more about what a big deal
  the paper is.
This paper advances all of science by about 25 years.
In fact, we have produced 200\% more science than anyone else writing on
  this topic.


\begin{table}[h]
  \centering
  \caption{Table of stuff}
  \label{tab:Stuff}

```



```

\begin{tabular}{rc|l}
\hline
Input 1 & Input 2 & Sum \\ \hline
5 & 10 & 15 \\
2 & 9 & 11 \\ \hline
\multicolumn{3}{l}{here's a remark.} \\
\end{tabular}
\end{table}

```

Table \ref{tab:Stuff} shows just all the cool science we do. Figure \ref{fig:plots} graphs some other stuff.

```

\begin{figure}[h]
\includegraphics[width=\columnwidth]{3dPlots.pdf}
\caption{Some plots}
\label{fig:plots}
\end{figure}

```

That figure sure was a pretty.\footnote{The prettiness of the figure was rated as 7.9 on the princess beauty scale.}

```
\end{document}
```

That gives us a title page. Note that we define the title, the name of the author, the footnoted information about the author, and the date all in the preamble material. Then to make the title page we just tell it to \maketitle and then to start a new page \newpage.

To add an abstract we just do the following

```

\documentclass[11]{article}
\usepackage[top=1in,bottom=1in,left=1.25in,right=1.25in]{geometry}
\usepackage{graphicx}
\usepackage{setspace}
\onehalfspacing

\title{On the Making of Science}
\author{Bullwinkle J. Moose\footnote{Ph.D. Candidate, Moose Institute for Toetapping.}}
\date{\today}

\begin{document}

\maketitle
\begin{abstract}
This paper does stuff
\end{abstract}
\newpage
This paper examines important scientific questions.
It does so in a super rigorous and presents results so well-identified that
they can't grab a bite without being spotted.

```

We address all the important concerns with our results and can say with absolute certainty that we have created the last word on this topic.

In this paragraph we want to say just a little more about what a big deal the paper is.

This paper advances all of science by about 25 years.

In fact, we have produced 200\% more science than anyone else writing on this topic.

```
\begin{table}[h]
  \centering
  \caption{Table of stuff}
  \label{tab:Stuff}
  \begin{tabular}{rc|l}
    \hline
    Input 1 & Input 2 & Sum \\ \hline
    5        & 10       & 15 \\
    2        & 9        & 11 \\ \hline
    \multicolumn{3}{l}{here's a remark.} \\
  \end{tabular}
\end{table}
```

Table \ref{tab:Stuff} shows just all the cool science we do. Figure \ref{fig:plots} graphs some other stuff.

```
\begin{figure}[h]
\includegraphics[width=\columnwidth]{3dPlots.pdf}
\caption{Some plots}
\label{fig:plots}
\end{figure}
```

That figure sure was a pretty.\footnote{The prettiness of the figure was rated as 7.9 on the princess beauty scale.}

```
\end{document}
```

2.4 Sections

Now ideally your papers are broken into several sections. This is easily accomplished with the commands `\section{title}`, `\subsection{text}`, and `\subsubsection{text}`. Let's give it a try.

```
\documentclass[11]{article}
\usepackage[top=1in,bottom=1in,left=1.25in,right=1.25in]{geometry}
\usepackage{graphicx}
\usepackage{setspace}
```

```

\onehalfspacing

\title{On the Making of Science}
\author{Bullwinkle J. Moose\footnote{Ph.D. Candidate, Moose Institute for
  Toetapping.}}
\date{\today}

\begin{document}

\maketitle
\begin{abstract}
This paper does stuff
\end{abstract}
\newpage
\section{Introduction}
This paper examines important scientific questions.
It does so in a super rigorous and presents results so well-identified that
  they can't grab a bite without being spotted.
We address all the important concerns with our results and can say with
  absolute certainty that we have created the last word on this topic.

In this paragraph we want to say just a little more about what a big deal
  the paper is.
This paper advances all of science by about 25 years.
In fact, we have produced 200\% more science than anyone else writing on
  this topic.

\section{Results}
\begin{table}[h]
\centering
\caption{Table of stuff}
\label{tab:Stuff}
\begin{tabular}{rc|l}
\hline
Input 1 & Input 2 & Sum \\ \hline
5 & 10 & 15 \\
2 & 9 & 11 \\ \hline
\multicolumn{3}{l}{here's a remark.} \\
\end{tabular}
\end{table}

Table \ref{tab:Stuff} shows just all the cool science we do. Figure \ref{
  fig:plots} graphs some other stuff.
\subsection{Interpretation}
\begin{figure}[h]
\includegraphics[width=\columnwidth]{3dPlots.pdf}
\caption{Some plots}

```

```

\label{fig:plots}
\end{figure}

\subsubsection*{More detail}
That figure sure was a pretty.\footnote{The prettiness of the figure was
    rated as 7.9 on the princess beauty scale.}

\end{document}

```

Nothing crazy going on there. Notice that we can use the `*` to get rid of the numbering on sections, subsections, and subsubsections; the choice is a matter of taste.

To create an appendix you just tell the software where the appendix starts by declaring `\appendix`. All sections after this are switched to letters rather than numbers.

2.5 Advanced tweaks

Let's take a brief moment to describe some of the things that you get

Bold, italics, underline To make these happen you use `\textbf{text}`, `\emph{text}`, and `\underline{text}`. The results are **text**, *text*, and text.

Changing colors Load the package `xcolor`. It's as simple as `\textcolor{red}{red}` to get red. Create your own colors by using `\definecolor{name}{model}{color-spec}`. More information at this [link](#)

Changing fonts There are all sorts of ways to include system fonts, but that's a pain. The easy way to shift between fonts is to stay within the default computer modern family and use commands like `\textsf{Sans Font}` and `\texttt{typewriter style}` to change fonts for effect. To change the font of a whole section or a whole document you can use `\fontfamily{code}\selectfont`. To do a whole section wrap the command and the section in braces

Here's some text that is in regular font

```

{
\fontfamily{code}\selectfont
This text is now sans.
}

```

And back to regular

To do from a certain point to the end get rid of the braces.

Here's some text that is in regular font

```

\fontfamily{code}\selectfont
This text is now sans.

```

And still sans

And to do a whole document just place it after `begin document`. To get some fonts you need to load a special package. More info at this [link](#). Notice that changing the fonts at the beginning of the document do not adjust the section headings. There are additional packages and commands for that, which you can look up.

Breaking lines and adding white space Some times you may want to break a line without getting the indent of a new paragraph. Just like with `table` we can use `\\`. This creates a line break and we can add white space by adding an option `\\[12pt]`. This creates a 12pt line break (equivalent to 1 line of text in a 12pt font document). We can also add more white space anywhere using `\hspace{length}` and `\vspace{length}` to create up and down and left and right white space of size `length`.

2.6 Putting it all together

Take five minutes and try and reconstruct the pdf I sent to you. You should know enough by now to be able to do it.

2.6.1 SyncTeX

A neat advantage of using TeXStudio is that it has a feature called SyncTeX. What this does is that it allows you to right click on the pdf preview and select “Go to Source.” This takes you back to the section of the source file that deals with that particular part of the pdf. This is very useful for editing.

Chapter 3

Writing maths

The next thing we're going to do is include math in our documents. Open a new, blank document, create the necessary header so that it compiles, and include this in the preamble `\usepackage{amsmath, amsfonts, amsthm}`. Now you can follow along by typing the commands and compiling as we go.

3.1 Basic math

The most basic math is in-line math. We just use the dollar sign to enter math mode. Note that either `$ $` or `\(\)` can be used to enter math mode I like the dollar sign because it's one key stroke. So `$2+2=4$` produces $2 + 2 = 4$. So far so good. There are lots of special math operators that only get used in this mode. So we can write `$x = \log(\exp(x))$` to get $x = \log(\exp(x))$. Some common functions are listed below. You can only use these functions when in math mode. I'm sure I forgot some functions you'll want to use, but google is your friend here. While we're on the subject of operators we can declare new operators in our preambles by using the function `\DeclareMathOperator{\var}{var}`. This gives us a new command `\var` that produces a plain text version of the variance operator. Compare

$$\begin{aligned}\mathrm{var}(x) &= \sigma^2 \\ \var(x) &= \sigma^2,\end{aligned}$$

where the top one is our new operator. We can also produce plain text words in math mode by using `\text{text}`.

Now writing everything inline is just a bad idea. It gets cramped and hard to read, so unless you hate your readers you'll want to mix it up with display math. The switch to display math is accomplished using either `$$ $$` or `\[\]`; \LaTeX connoisseurs seem to have a preference here for the later, but I can't tell the difference. Switching to display you go from $f(x) = \exp(x)$ to

$$f(x) = \exp(x).$$

This may not seem like a big deal, but you'll find cases where you'll want to do this for style, ease of reading, and to emphasize the important steps/results in the process.

To add subscripts and superscripts to variables we use the underscore and the carrot. For example to write the sum from $i = 1$ to N x_i^2 we have `\sum_{i=1}^N x_i^2`. Notice that we use the subscript and the superscript on the summation operator to set the bounds

Table 3.1: Common math commands

Command	Outcome
<code>\exp</code>	\exp
<code>\log</code>	\log
<code>\ln</code>	\ln
<code>\sqrt{x}</code>	\sqrt{x}
<code>\sin</code>	\sin
<code>\tan</code>	\tan
<code>\cos</code>	\cos
<code>\sum</code>	\sum
<code>\prod</code>	\prod
<code>\min</code>	\min
<code>\max</code>	\max
<code>\int</code>	\int
<code>\partial</code>	∂
<code>\lim</code>	\lim
<code>\vee</code>	\vee
<code>\wedge</code>	\wedge
<code>\cup</code>	\cup
<code>\cap</code>	\cap
<code>\rightarrow</code>	\rightarrow
<code>\cdot</code>	\cdot
<code>\times</code>	\times
<code>\implies</code>	\implies
<code>\in</code>	\in
<code>\setminus</code>	\setminus
<code>\geq</code>	\geq
<code>\leq</code>	\leq
<code>\neq</code>	\neq
<code>\infty</code>	∞
<code>\emptyset</code>	\emptyset

on the operation. The inline output is $\sum_{i=1}^N x_i^2$ and the display output is

$$\sum_{i=1}^N x_i^2.$$

Notice that with the bounds on the operator, there is a cosmetic difference between the display and the inline presentations. The other thing to note is the use of `{i=1}` to tell the software that the whole segment is the subscript. Omitting the braces is a common mistake with obvious results

$$\sum_i = 1^N x_i^2.$$

Can you tell me why this looks like it does?

Note that we can create a new operator with limits. For example if want wanted to write

$$\theta^* = \arg \max_{\theta} -\theta^2,$$

but we didn't like the space between arg and max. We can declare our own argmax operator with limits as `\DeclareMathOperator*\argmax{\operatornamewithlimits{argmax}}` and now we have

$$\theta^* = \operatornamewithlimits{argmax}_{\theta} -\theta^2.$$

The star tells L^AT_EX to be smart and adjust where it places the subscript/superscript depending on whether we're in display or inline mode.

Fractions easy to implement here we just use `\frac{num}{den}` to have the top and bottom parts. Inline this gives us $\frac{1}{2}$ and display

$$\frac{x}{y}.$$

While we're looking at fractions, let's consider the following issue.

```
\[
\frac{1}{N} (\sum_{i=1}^N \log(f(\theta|y_i))) + (1-\frac{1}{p})X
\]
```

produces

$$\frac{1}{N} \left(\sum_{i=1}^N \log(f(\theta|y_i)) \right) + \left(1 - \frac{1}{p} \right) X,$$

which is garbage. Look at those terrible parentheses. We can make them self adjusting with the following tweak.

```
\[
\frac{1}{N} \left( \sum_{i=1}^N \log(f(\theta|y_i)) \right) + \left( 1 - \frac{1}{p} \right) X
\]
```

to get

$$\frac{1}{N} \left(\sum_{i=1}^N \log(f(\theta|y_i)) \right) + \left(1 - \frac{1}{p} \right) X.$$

This also works for brackets and curly braces. The only thing different is because braces are special they are written `\left\{ \right\}`.

$$\frac{1}{N} \left[\sum_{i=1}^N \log(f(\theta|y_i)) \right] + \left\{ 1 - \frac{1}{p} \right\} X.$$

Binomials (the choose function) works the same way as fractions just replace `frac` with `binom` to get

$$\binom{1}{4}.$$

Greek letters are also important. They are exactly what you might think they would be where it's just backslash and the name of the letter. This means that θ is `\theta` and capitals like Θ is with a capital `\Theta`.

3.2 Other fancy equations

Numbered equations are also an important tool to draw attention to the important results. To get equations with number we have the equation environment.

```
\begin{equation}\label{eq:exp}  
F(t) = 1- \exp(-\lambda t)  
\end{equation}
```

which produces

$$F(t) = 1 - \exp(-\lambda t) \quad (3.1)$$

Note the use of the label command works exactly like it did with figures and tables so we can refer to Equation 3.1 by using `\ref{eq:exp}`.

Being able to line up multiple equations creates a nice clean visual effect. The align environment provides a system for doing that. This code

```
\begin{align*}  
F(t) &= 1- \exp(-\lambda t)\\  
F(0) &= 1-\exp(0)\\  
      &= 0.  
\end{align*}
```

produces

$$\begin{aligned} F(t) &= 1 - \exp(-\lambda t) \\ F(0) &= 1 - \exp(0) \\ &= 0. \end{aligned}$$

Notice that this combines a few things we've done so far. The * removes numbers, just like it did with sections, while the & is what lines things up and the \\ breaks the line, just like with tables. If we remove the * we get equation numbers for each line that can be assigned labels.

Piecewise functions are also important and will come up in your futures. The cases environment inside math mode helps us here.

```
\[ %math mode on  
f(t) =  
\begin{cases}  
  1-\exp(-\lambda t) & \text{if } t>0\\  
  0 & \text{otherwise.}  
\end{cases}  
\[ %math mode off
```

This code produces

$$f(t) = \begin{cases} 1 - \exp(-\lambda t) & \text{if } t > 0 \\ 0 & \text{otherwise.} \end{cases}$$

Notice the use of `\text{}` to produce normal looking text inside math. Also notice that I included the space after “if” because math mode otherwise runs things together.

3.3 Matrices

At some point you may find yourself with the unenviable task of writing a matrix. Here's how you would go about that.

```
\[ %math mode on
A =
\begin{bmatrix*}
a_{11} & a_{12} \\
a_{21} & a_{22} \\
\end{bmatrix*}
\] %math mode off
```

And this gives you

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

The b in bmatrix refers to the brackets. You can also use parentheses. For instance

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \dots & a_{m,n} \end{pmatrix}$$

is made by

```
\[
A =
\begin{pmatrix*}
a_{1,1} & a_{1,2} & \ldots & a_{1,n} \\
a_{2,1} & a_{2,2} & \ldots & a_{2,n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m,1} & a_{m,2} & \ldots & a_{m,n}
\end{pmatrix*}
\]
```

3.4 Theorems and proofs

The theorem and proof environments are the basis for most formal theoretical analysis. You setup these environments by declaring theorems in your preamble. For example,

```
\newtheorem{thm}{Theorem}
\newtheorem{claim}{Claim}
\newtheorem{proposition}{Proposition}
\newtheorem{lemma}{Lemma}
```

defines four environments we can use. Let's go ahead and write a theorem with the following

```
\begin{thm}\label{th:subsets}
If  $A$  is a subset of  $B$  and  $B$  is a subset of  $C$  then  $A$  is a subset
of  $C$ .
\end{thm}
```

Theorem 1. *If A is a subset of B and B is a subset of C then A is a subset of C .*

Now we have the statement of Theorem 1. For the proof we use the proof environment

```
\begin{proof}
Left as an exercise to the reader.
\end{proof}
```

Proof. Left as an exercise to the reader. □

3.5 Fancy fonts and other notations

Table 3.2 shows some additional things you may want to use

Table 3.2: Other math		
Command	Use	Examples
<code>\mathbb{ }</code>	Blackboard math	\mathbb{R} is used to denote real numbers
<code>\mathbf{ }</code>	Boldface math	\mathbf{X} bold is sometimes used for matrices
<code>\mathfrak{ }</code>	Fraktur math	\mathfrak{R} is also sometimes used to denote real numbers
<code>\hat{ }</code>	Add a hat to variable	$\hat{\theta}$ is an estimate of θ
<code>\tilde{ }</code>	Add a tilde to variable	$\tilde{\theta}$ a sometimes used to denote a specific value of θ
<code>\bar{ }</code>	Add a bar to variable	\bar{x} often used to denote the mean of x

3.6 Your turn

Try and reproduce mathExample.pdf.

Chapter 4

References using bibTeX

Everything we write has references to other work. To avoid plagiarism, we cite this work. L^AT_EX has a couple different citation packages, we will focus on the most common one called bibTeX.¹

4.1 Organizing references with JabRef

BibTeX requires the use of a `.bib` file where you store your references. The references have to be stored in a certain way, but we will use a particular program to store, organize, and edit our reference database. This program is called JabRef. Download and install from this [link](#).

Once you have that installed, open it up and create a new database. You can add an entry using the green plus button. It will ask you what the kind of thing you want to cite and then give you a form to fill in. Let's try it out. Go to this [link](#), and create a reference for it as an article. Fill the form and create your own bibTeX key. This is just a short tag that you will use when you want to cite the article. I usually use author year. For instance, I would call this `FearonLaitin2003`.

Often however, we don't even need to do the form. Nearly every journal repository (sage, jstor, etc.) has a button that will produce the bibTeX source code. For example, check out this [link](#). If we look at the right column we see a section labeled "Services" and about half way down that list we find a link that says "Download to citation manager." Click that link and the first choice on the right hand list is bibTeX, click on it. You can now copy all of this text, switch over to JabRef, add another article, but this time click on the source tab of the form and paste the text from the link. That's all you need to do, but you may want to choose a different key for the article.

Google Scholar also has support for bibTeX. For example, check out this [link](#). Click the cite button on the first link, this will open up a window with some citations. We want the bibTeX source, so click on that link and like before we can just copy and paste that into a new JabRef entry.

At this point you should have three references in your database. Save it to the folder with the document we created back in Chapter 2, because we will now add references to this document. I'm calling my file `texCourse.bib`.

¹The other most common is a very flexible package called bibL^AT_EX. We won't talk about this, but you should be aware of it if you need something adjustable.

We'll add one line to our preamble and two lines to the end. Those ending lines set the style of our bibliography (in this case the style of the `apsr`), and then we direct it to our `bib` file.

4.2 Setting up your document

```
\documentclass[11]{article}
\usepackage[top=1in,bottom=1in,left=1.25in,right=1.25in]{geometry}
\usepackage{graphicx}
\usepackage{setspace}
\onehalfspacing
\title{On the Making of Science}
\author{Bullwinkle J. Moose\footnote{Ph.D. Candidate, Moose Institute for
Toetapping.}}
\date{\today}

\usepackage{natbib} %NEW BIBTEX STUFF

\begin{document}
\maketitle
\begin{abstract}
This paper does stuff
\end{abstract}
\newpage
\section{Introduction}
This paper examines important scientific questions.
It does so in a super rigorous and presents results so well-identified that
they can't grab a bite without being spotted.
We address all the important concerns with our results and can say with
absolute certainty that we have created the last word on this topic.
In this paragraph we want to say just a little more about what a big deal
the paper is.
This paper advances all of science by about 25 years.
In fact, we have produced 200\% more science than anyone else writing on
this topic.
\section{Results}
\begin{table}[h]
\centering
\caption{Table of stuff}
\label{tab:Stuff}
\begin{tabular}{rc|l}
\hline
Input 1 & Input 2 & Sum \\ \hline
5 & 10 & 15 \\
2 & 9 & 11 \\ \hline
\multicolumn{3}{l}{here's a remark.} \\
\end{tabular}
\end{table}
```

Table 4.1: Cite Commands

Command	Output
<code>\cite{FearonLaitin2003}</code> or <code>\citet{FearonLaitin2003}</code>	Fearon and Laitin (2003)
<code>\citep{FearonLaitin2003}</code>	(Fearon and Laitin, 2003)
<code>\citeyearpar{FearonLaitin2003}</code>	(2003)
<code>\cite[29]{FearonLaitin2003}</code>	Fearon and Laitin (2003, 29)
<code>\citep[see][29]{FearonLaitin2003}</code>	(see Fearon and Laitin 2003, 29)

```

\end{table}
Table \ref{tab:Stuff} shows just all the cool science we do.
Figure \ref{fig:plots} graphs some other stuff.
\subsection{Interpretation}
\begin{figure}[h]
\includegraphics[width=\columnwidth]{3dPlots.pdf}
\caption{Some plots}
\label{fig:plots}
\end{figure}
\subsubsection*{More detail}
That figure sure was a pretty.\footnote{The prettiness of the figure was
rated as 7.9 on the princess beauty scale.}

% % % BIBTEX STUFF
\bibliographystyle{apsr}
\bibliography{texCourse}
% % %
\end{document}

```

4.3 Citing commands

We’re now ready to actually do some citing. Table 4.1 shows a bunch of different ways to cite works.

The use of brackets adds notes to the citation. One set of brackets adds text to the end of the cite. Two sets of brackets sandwiches the citation between the two bits of text. These are the most common cite commands, others can be found [here](#) if you need more flexibility.

Let’s add a lit review to our example.

```

\documentclass[11]{article}
\usepackage[top=1in,bottom=1in,left=1.25in,right=1.25in]{geometry}
\usepackage{graphicx}
\usepackage{setspace}
\onehalfspacing
\title{On the Making of Science}
\author{Bullwinkle J. Moose\footnote{Ph.D. Candidate, Moose Institute for
Toetapping.}}
\date{\today}

```

```

\usepackage{natbib}

\begin{document}
\maketitle
\begin{abstract}
This paper does stuff
\end{abstract}
\newpage
\section{Introduction}
This paper examines important scientific questions.
It does so in a super rigorous and presents results so well-identified that
they can't grab a bite without being spotted.
We address all the important concerns with our results and can say with
absolute certainty that we have created the last word on this topic.
In this paragraph we want to say just a little more about what a big deal
the paper is.
This paper advances all of science by about 25 years.
In fact, we have produced 200\% more science than anyone else writing on
this topic.

% % % NEW SECTION % % %
\section{Literature}
\cite{Walter2009b, FearonLaitin2003} both show that something effects civil
wars.
However, others have also shown things \citep[e.g.][]{CollierAll2004}.
In particular, Walter's \citeyearpar{Walter2009b} findings are very
interesting, and her Table 1 is really awesome \citep[78]{Walter2009b}.

\section{Results}
\begin{table}[h]
\centering
\caption{Table of stuff}
\label{tab:Stuff}
\begin{tabular}{rc|l}
\hline
Input 1 & Input 2 & Sum \\ \hline
5 & 10 & 15 \\
2 & 9 & 11 \\ \hline
\multicolumn{3}{l}{here's a remark.} \\
\end{tabular}
\end{table}
Table \ref{tab:Stuff} shows just all the cool science we do.
Figure \ref{fig:plots} graphs some other stuff.
\subsection{Interpretation}
\begin{figure}[h]
\includegraphics[width=\columnwidth]{3dPlots.pdf}
\caption{Some plots}

```

```

\label{fig:plots}
\end{figure}
\subsubsection*{More detail}
That figure sure was a pretty.\footnote{The prettiness of the figure was
rated as 7.9 on the princess beauty scale.}

\bibliographystyle{apsr}
\bibliography{texCourse}
\end{document}

```

4.4 Other fancy characters (accents)

Notice that when we copied the citation for Collier, et al. there were special characters in Møas Söderbom's name. The copy-paste routine did not help us with that and these character's didn't appear in our document. We will need to manually fix that in our bibT_EX file, but before we can do that, we need to know how to write these characters. For more characters check [this table](#). Go ahead and edit the Collier reference over in JabRef, save

Table 4.2: Special letters

Command	Output
<code>\`{o}</code>	ò
<code>\'{o}</code>	ó
<code>\^{o}</code>	ô
<code>\"{o}</code>	ö
<code>\~{o}</code>	õ
<code>\c{c}</code>	ç
<code>\r{a}</code>	å

the file and recompile the example. Make sure that it works.

Chapter 5

Making presentations with beamer

L^AT_EX is great for making papers, but it's great for other things like CVs, letters, and presentations. It is presentation to which we now turn our attention. There are many ways to create slideshows in L^AT_EX, and there are many different ways to customize them. We will focus on making basic beamer presentations and doing some mild customizations.

Let's start by opening a new file and adding the following

```
\documentclass{beamer}
\author{Rocket J. Squirrel}
\institute{Moose Institute for Toetapping}
\title{Flying Squirrels can't Actually Fly}
\date{\today}

\begin{document}
\frame{\maketitle}
\end{document}
```

Not much as changed from when we were making papers. The first change is we switched from article to beamer in the document class. Second, add the institute line to our preamble. Finally, we have our maketitle command wrapped inside something called a frame. A frame is just what beamer calls a slide. When we compile this we get a single slide, which is the title slide.

5.1 Theme and color adjustments

There are two important elements to adjust if you want to tweak the way your slides work. They are `\usebeamertheme` to adjust layout and extras and `\usecolortheme` to set the color palette. This [link](#) shows the various built-in combinations of these two options. You can look through those at your leisure and find combinations that appeal to you. For now, we will use the Frankfurt theme and seahorse colors. This looks like this

```
\documentclass{beamer}
\author{Rocket J. Squirrel}
\institute{Moose Institute for Toetapping}
\title{Flying Squirrels can't Actually Fly}
\date{\today}
```

```

\usetheme{Frankfurt}
\usecolortheme{seahorse}

\begin{document}
\frame{\maketitle}
\end{document}

```

5.2 Frames

With a setup that is easy on the eye, we will now add some slides

```

\documentclass{beamer}
\author{Rocket J. Squirrel}
\institute{Moose Institute for Toetapping}
\title{Flying Squirrels can't Actually Fly}
\date{\today}
\usetheme{Frankfurt}
\usecolortheme{seahorse}

\begin{document}
\frame{\maketitle}

%Navigation choices
\section{Introduction}
\stepcounter{subsection}

%slide 1
\begin{frame}{Introduction}
This is a great presentation.

We will show many things.
\end{frame}

%slide 2
\begin{frame}{Introduction (con't)}
How good are we?
\begin{itemize}
\item We're not just good
\item we're great
\end{itemize}
\end{frame}

\end{document}

```

This has created two slides, both of which are part of our introduction section. The call to `stepcounter` is to tell the software how to pick separate the slides for the navigation bar (the dots in this theme).

Tables, figures, math, lists, colors, and just about everything else works exactly the

same here, so we won't redo them. Something we haven't talked about yet that you may want to include on your slides are columns. Columns are just another environment, our new slide will have two columns.

```
\documentclass{beamer}
\author{Rocket J. Squirrel}
\institute{Moose Institute for Toetapping}
\title{Flying Squirrels can't Actually Fly}
\date{\today}
\usetheme{Frankfurt}
\usecolortheme{seahorse}

\begin{document}
\frame{\maketitle}

%Navigation choices
\section{Introduction}
\stepcounter{subsection}

%slide 1
\begin{frame}{Introduction}
This is a great presentation.

We will show many things.
\end{frame}

%slide 2
\begin{frame}{Introduction (con't)}
How good are we?
\begin{itemize}
\item We're not just good
\item we're great
\end{itemize}
\end{frame}

\section{Results}
\stepcounter{subsection}
\begin{frame}{Main Results}
\begin{columns}
\begin{column}{.4\textwidth}
The main result shows that something happened\\[12pt]
The second result is unclear
\end{column}
\begin{column}{.6\textwidth}
\begin{tabular}{rcc}
& Model 1 & Model 2 \\ \hline
 $X_1$  & 2.99 & 5.88
\end{tabular}
\end{column}
\end{columns}
\end{frame}
```

```

$X_2$&    -4.50 & 0.23\\ \hline
\end{tabular}
\end{column}
\end{columns}
\end{frame}

\end{document}

```

What we did here was open the columns environment and then begin and end one column at a time. For each column we also specified how much of the page it will take.

5.3 Appendix slides and references

Often times we want to include additional slides that don't show up in the main we can do this by adding the `\appendix` command. All sections after this declaration are separated from the main slides.

Adding citations and a bibliography is exactly the same as before, but with one exception. We just need to add the following line to our preamble to make everything work in beamer.

```
\renewcommand{\bibsection}{\subsubsection*{\bibname}}
```

After that we're good to go.

Chapter 6

Producing simple images with TikZ

This chapter deals with a new producing TikZ images. To create these images we begin with a new command

```
\usepackage{tikz}
```

This opens up a whole world of drawing pictures and diagrams.

6.1 Basic Lines

While not strictly necessary we will treat our first couple TikZ images as figures. This means that our images will take the following form.

```
\begin{figure}
  \centering
  \begin{tikzpicture}
    \draw (0, 0) -- (1,2);
  \end{tikzpicture}
  \caption{My first TikZ line}
  \label{fig:tikz1}
\end{figure}
```



Figure 6.1: My first TikZ line

What this produces is a simple line from the point (0,0) to the point (1,2). What tells us it's a line is the `--`. Also, note that TikZ requires that we end commands with a semi-colon. The use of points tells us that there is some kind of grid system in place. We can tell TikZ to print that grid for us. Often times, this makes the drawing stage easier, and we will turn off the grid upon completion.

```
\begin{figure}
  \centering
```

```

\begin{tikzpicture} %start a picture
  \draw[help lines] (0,0) grid (3,3);
  \draw (0, 0) -- (1,2)--(0,3);
\end{tikzpicture}
\caption{My first TikZ line with lines}
\label{fig:tikz1GRID}
\end{figure}

```

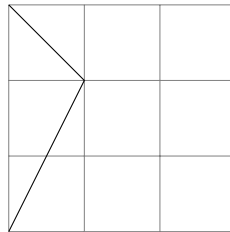


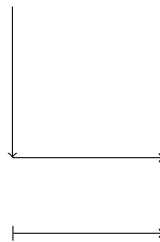
Figure 6.2: My first TikZ line with lines

We can turn lines into arrows by just adding heads, and we can add flat ends just as easily.

```

\begin{figure}
\centering
\begin{tikzpicture}
\draw [->] (0,0) -- (2,0);
\draw [<-] (0,0) -- (0,2);
\draw [|->] (0,-1) -- (2,-1);
\end{tikzpicture}
\end{figure}

```

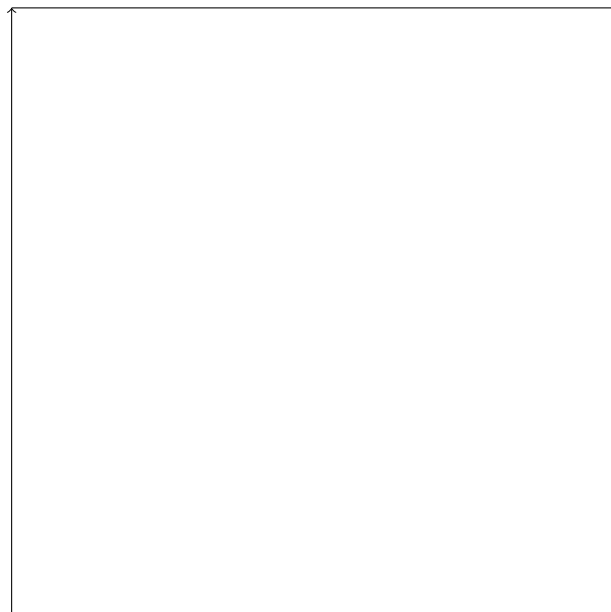
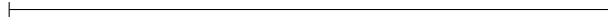


We can make these bigger and smaller using the scale option.

```

\begin{figure}
\centering
\begin{tikzpicture}[scale=4]
\draw [->] (0,0) -- (2,0);
\draw [<-] (0,0) -- (0,-2);
\draw [|->] (0,1) -- (2,1);
\end{tikzpicture}
\end{figure}

```



This option scales the image uniformly. If we scale in particular dimensions by changing that option to `[xscale=3, yscale=4]`.

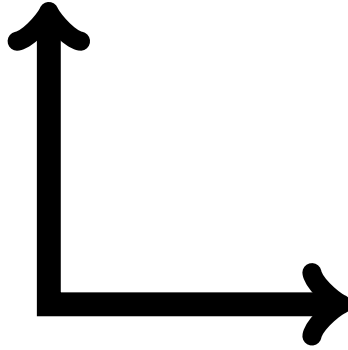
The thickness of lines can also be changed.

```
\begin{figure}
\centering
\begin{tikzpicture}[scale=4]
\draw [ultra thick] [->] (0,0) -- (2,0);
\end{tikzpicture}
\end{figure}
```



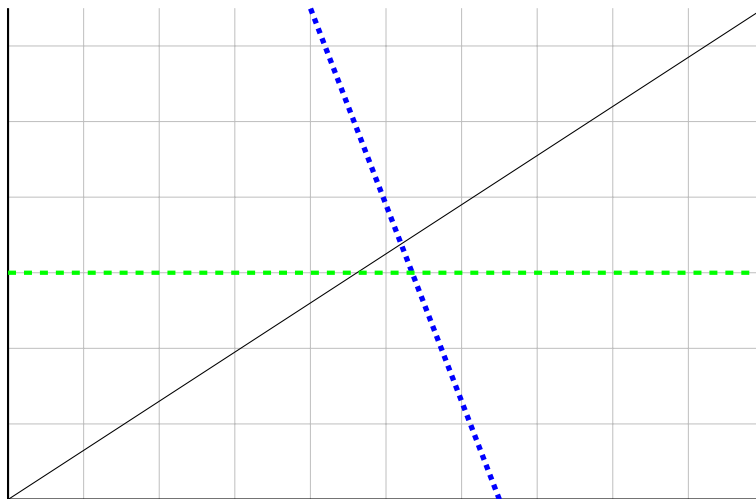
Thickness options are ultra thin, very thin, thin, semithick, thick, very thick, and ultra thick. Although you can also specify a particular thickness using the option `line width`.

```
\begin{figure}
\centering
\begin{tikzpicture}
\draw [line width = 9pt] [<->] (0,4)-- (0,0) -- (4,0);
\end{tikzpicture}
\end{figure}
```



You can specify dashed, dotted, and colored lines, as well.

```
\begin{figure}
\centering
\begin{tikzpicture}
  \draw[help lines, lightgray] (0,0) grid (10, 6.5);
  \draw[thick] (0, 6.5) -- (0,0) -- (10, 0); %axes
  \draw[thin] (0,0) -- (10,6.5);
  \draw[dotted, blue, line width=2pt] (6.5,0)--(4,6.5);
  \draw[dashed, green, ultra thick] (0, 3) -- (10,3);
\end{tikzpicture}
\end{figure}
```



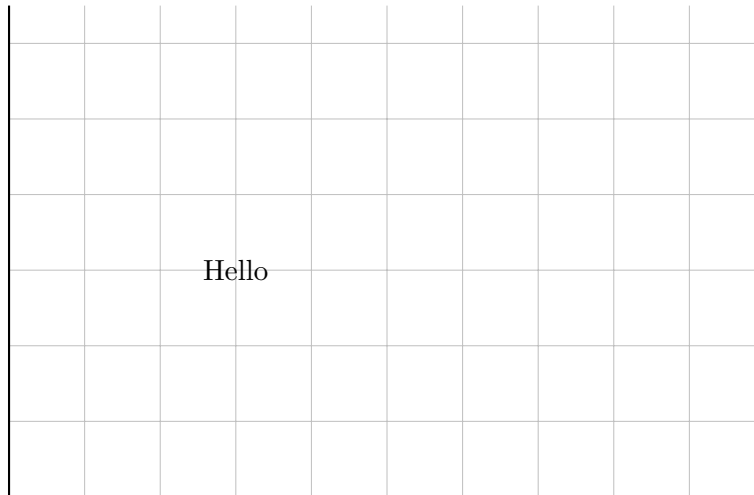
6.1.1 Adding Text

We can easily add text to our pictures as well

```
\begin{figure}
\centering
\begin{tikzpicture}
  \draw[help lines, lightgray] (0,0) grid (10, 6.5);
  \draw[thick] (0, 6.5) -- (0,0) -- (10, 0); %axes
  \node at (3,3) {Hello};
\end{tikzpicture}
\end{figure}
```

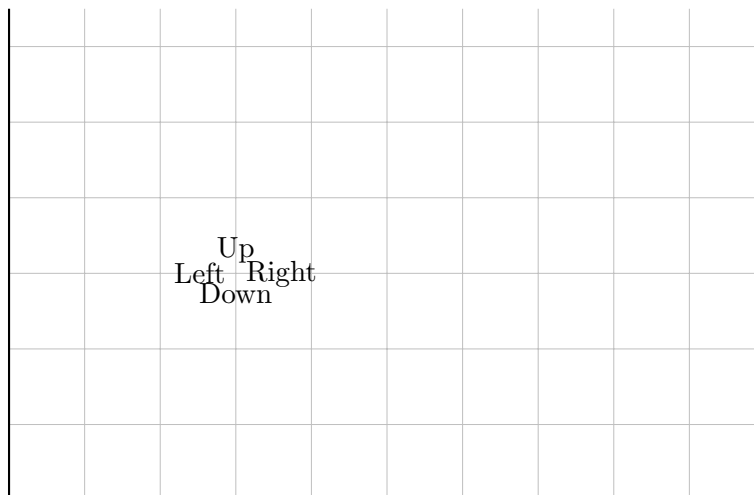


```
\end{figure}
```



By default this text is roughly centered, but we can adjust that

```
\begin{figure}
\centering
\begin{tikzpicture}
  \draw[help lines, lightgray] (0,0) grid (10, 6.5);
  \draw[thick] (0, 6.5) -- (0,0) -- (10, 0); %axes
  \node [below] at (3,3) {Down};
  \node [above] at (3,3) {Up};
  \node [left] at (3,3) {Left};
  \node [right] at (3,3) {Right};
\end{tikzpicture}
\end{figure}
```

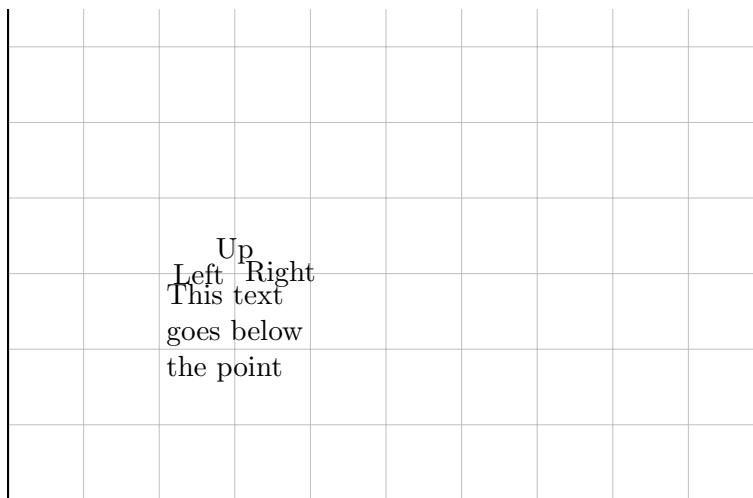


Note that if your text requires multiple lines we use the standard `\\` to break lines, but you need to tell TikZ how to align the text.

```

\begin{figure}
\centering
\begin{tikzpicture}
  \draw[help lines, lightgray] (0,0) grid (10, 6.5);
  \draw[thick] (0, 6.5) -- (0,0) -- (10, 0); %axes
  \node [below, align=left] at (3,3) {This text \\\ goes below\\the point
};
  \node [above] at (3,3) {Up};
  \node [left] at (3,3) {Left};
  \node [right] at (3,3) {Right};
\end{tikzpicture}
\end{figure}

```

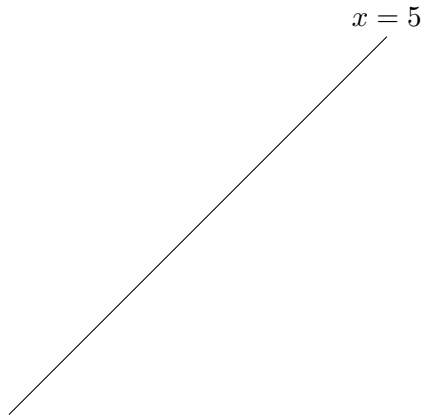


You can also add labels to start or end of lines.

```

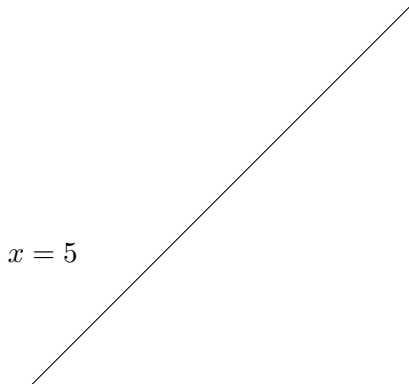
\begin{figure}
\centering
\begin{tikzpicture}
  \draw[thin] (0,0) -- (5,5) node[above]{$x=5$};
\end{tikzpicture}
\end{figure}

```



We can also put text in the middle of a line, note that `pos` tells TizK how close to put it to the end points on a $[0, 1]$ scale.

```
\begin{figure}
\centering
\begin{tikzpicture}
\draw[thin] (0,0) -- node[pos=.35, left=1cm]{$x=5$} (5,5);
\end{tikzpicture}
\end{figure}
```



Note that the direction (north, east, south, or west) says that the point $(5, 5)$ is south of the text.

We can put shapes around text as well.

```
\begin{figure}
\centering
\begin{tikzpicture}
\node[circle, draw=black] at (0,0) {$A$};
\end{tikzpicture}
\end{figure}
```

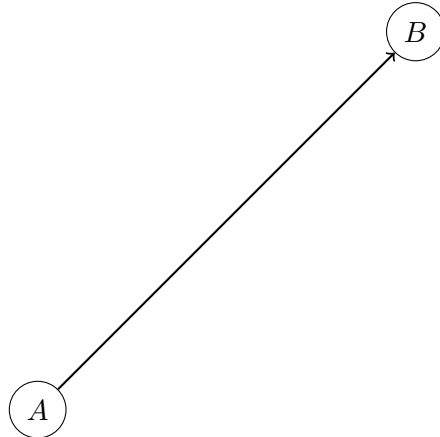


Nodes can also be labeled for internal reference.

```

\begin{figure}
\centering
\begin{tikzpicture}
  \node[circle, draw=black] (node1) at (0,0) {$A$};
  \node[circle, draw=black] (node2) at (5,5) {$B$};
  \draw[thick, ->] (node1) -- (node2) ;
\end{tikzpicture}
\end{figure}

```



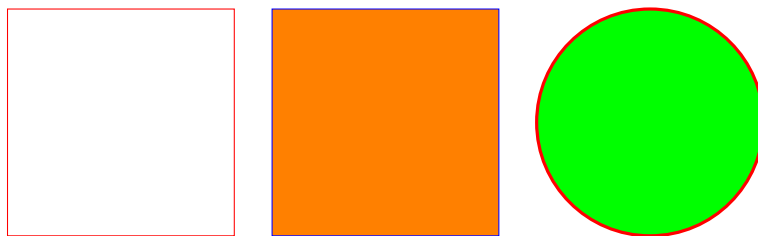
6.2 Curves

We're not really limited to just lines, we also can do shapes and curves. The main shapes are rectangles and circles.

```

\begin{figure}
\centering
\begin{tikzpicture}
  \draw[red] (0,0) -- (0,3)--(3,3)--(3,0)--(0,0); %hard way
  \draw[blue, fill=orange] (3.5, 0) rectangle (6.5, 3); %easy, and filled
  \draw[red, fill=green, very thick] (8.5,1.5) circle [radius=1.5];
\end{tikzpicture}
\end{figure}

```



To plot a curve through some points you can draw a plot. Smooth keeps things smooth, and the tension also helps with that. I don't really have a good grasp on how tension works, but I do know that it gets really wacky if you set it too big.

```

\begin{figure}
\centering
\begin{tikzpicture}
  \draw[thick] (0, 6.5) -- (0,0) -- (10, 0); %axes
  \draw[thin] plot [smooth, tension=1] coordinates { (0,1.5) (3.5,2.75)
    (5,4)};
\end{tikzpicture}
\end{figure}

```



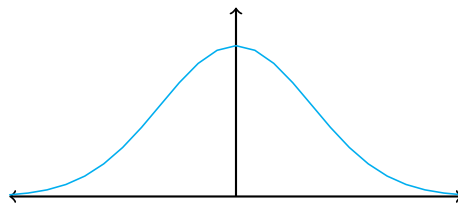
6.2.1 Functions

Alternatively, if you actually know the function you want to plot, you can program it in directly.

```

%Normal PDF
\begin{figure}
\centering
\begin{tikzpicture}[yscale=5]
  \draw[thick, <->] (-3,0) -- (3,0); %x-axis
  \draw[thick, ->] (0,0)--(0, .5); %y-axis
  \draw[domain=-3:3, cyan, semithick] plot (\x, {1/sqrt(2*pi) * exp(-(\x)
    ^2 / 2)});
\end{tikzpicture}
\end{figure}

```

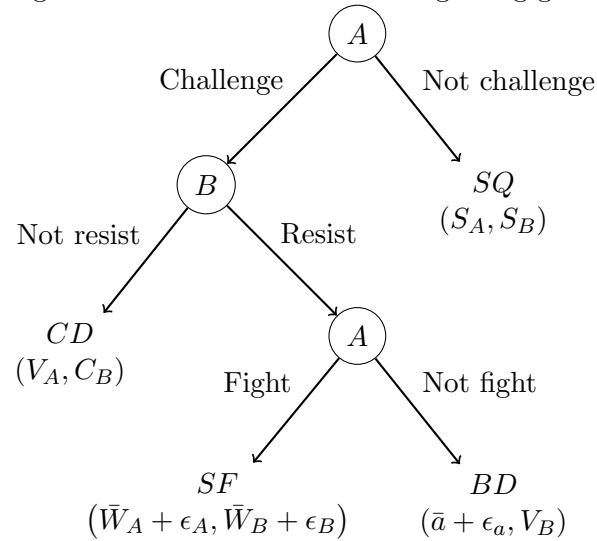


This creates a very easy way to graph functions that are important to your work.

6.3 Drawing games

The last thing we'll consider here is an application of TikZ to draw extensive form games. We're skipping normal form games because those are just rectangles and we all ready did that. You should know enough to produce the following image.

Figure 6.3: The canonical crisis-signaling game



Try it and see how you do.