

# Learning L<sup>A</sup>T<sub>E</sub>X

Patrick Lam

# setting up

## setting up

1. download a TeX distribution (MiKTeX, MacTeX, etc.)
2. download an editor (Texmaker, WinEDT, XEmacs, etc.)
3. start a .tex file in editor
4. work only in the .tex file

Let  $\langle \rangle$  denote things you fill in, but without the  $\langle \rangle$ .

Let `<>` denote things you fill in, but without the `<>`.

For example, for `<myname>`, I would write `Patrick`.

Example code is in red

# commands

# commands

You have to tell  $\text{\LaTeX}$  to do everything with commands, which always begin with  $\backslash$ :



# commands

You have to tell  $\text{\LaTeX}$  to do everything with commands, which always begin with  $\backslash$ :

```
 $\backslash$ <command>
```

```
 $\backslash$ <command>{<something>}
```

```
 $\backslash$ <command>[<options>]{<something>}
```

the bare minimum

## the bare minimum

```
\documentclass[<options>]{<class>}
```

```
\begin{document}
```

```
My text here!
```

```
\end{document}
```

## the bare minimum

```
\documentclass[<options>]{<class>}
```

```
\begin{document}
```

```
My text here!
```

```
\end{document}
```

```
article, beamer ∈ <class>
```

## the bare minimum

```
\documentclass[<options>]{<class>}
```

```
\begin{document}
```

```
My text here!
```

```
\end{document}
```

article, beamer  $\in$  <class>

translation: some class types include article and beamer

## the bare minimum

```
\documentclass[<options>]{<class>}
```

```
\begin{document}
```

```
My text here!
```

```
\end{document}
```

article, beamer  $\in$  <class>

translation: some class types include article and beamer

For almost everything, we will be using article.

.tex -> something readable

On your .tex file, compile using LaTeX or PDFLaTeX (usually buttons or commands on your editor).

.tex -> something readable

On your .tex file, compile using LaTeX or PDFLaTeX (usually buttons or commands on your editor).

- ▶ .tex file -> LaTeX -> .dvi file



.tex -> something readable

On your .tex file, compile using LaTeX or PDFLaTeX (usually buttons or commands on your editor).

- ▶ .tex file -> LaTeX -> .dvi file
  - ▶ you can convert .dvi to .pdf with dvi2pdf or dvi2ps and ps2pdf

.tex -> something readable

On your .tex file, compile using LaTeX or PDFLaTeX (usually buttons or commands on your editor).

- ▶ .tex file -> LaTeX -> .dvi file
  - ▶ you can convert .dvi to .pdf with dvi2pdf or dvi2ps and ps2pdf
- ▶ .tex file -> PDFLaTeX -> .pdf file

`.tex` -> something readable

On your `.tex` file, compile using LaTeX or PDFLaTeX (usually buttons or commands on your editor).

- ▶ `.tex` file -> LaTeX -> `.dvi` file
  - ▶ you can convert `.dvi` to `.pdf` with `dvi2pdf` or `dvi2ps` and `ps2pdf`
- ▶ `.tex` file -> PDFLaTeX -> `.pdf` file

Output files will be in the same directory as your `.tex` file.

Try compiling now!

Try compiling now!

If you get an error message, something is wrong in your code.

Try compiling now!

If you get an error message, something is wrong in your code.

Compile often to catch errors before they pile up!

Let's get started with more complicated stuff!

title, author, date

Let's give our article a title, author, and date.



title, author, date

Let's give our article a title, author, and date.

```
\documentclass{article}
```

```
\title{This is my title}
```

```
\author{Patrick Lam}
```

```
\date{}
```

```
\begin{document}
```

```
My text here!
```

```
\end{document}
```

# packages

Tell  $\text{\LaTeX}$  to use some packages before beginning the document:

# packages

Tell  $\text{\LaTeX}$  to use some packages before beginning the document:

```
\documentclass{article}
```

```
\usepackage{<package>}
```

```
\usepackage{<package>}
```

```
\title{This is my title}
```

```
\author{Patrick Lam}
```

```
\date{}
```

```
\begin{document}
```

```
My text here!
```

```
\end{document}
```

Everything else from now on goes after `\begin{document}`.

Everything else from now on goes after `\begin{document}`.

```
\begin{document}
```

Everything I write should go here.

```
\end{document}
```

OK, let's try compiling what we have.

OK, let's try compiling what we have.

Where is our title, abstract and date?

We have to tell  $\text{\LaTeX}$  to put it in our document with `\maketitle` or `\titlepage`:



We have to tell  $\text{\LaTeX}$  to put it in our document with `\maketitle` or `\titlepage`:

```
\begin{document}  
\maketitle
```

My text here!

```
\end{document}
```

We have to tell  $\text{\LaTeX}$  to put it in our document with `\maketitle` or `\titlepage`:

```
\begin{document}  
\maketitle
```

My text here!

```
\end{document}
```

Try compiling again!

# lines

$\text{\LaTeX}$  is smart with line spacing.

## lines

$\text{\LaTeX}$  is smart with line spacing.

To start a new paragraph, skip a line in your .tex file:

# lines

L<sup>A</sup>T<sub>E</sub>X is smart with line spacing.

To start a new paragraph, skip a line in your .tex file:

Paragraph 1

Paragraph 2

Paragraph 3

To end a line and start a new line, use `\\`:

To end a line and start a new line, use \\:

```
This is my line. \\
This is my new line.
```

To put space between lines, use multiple `\\`, `\bigskip`,  
`\medskip`, or `\smallskip`:



To put space between lines, use multiple `\\`, `\bigskip`, `\medskip`, or `\smallskip`:

```
This is my line. \\ \\ \\
This is my new line. \\
\bigskip
This is another line.
```

To double space, use the `setspace` package and the `\doublespacing` command:

To double space, use the setspace package and the `\doublespacing` command:

```
\documentclass{article}
```

```
\usepackage{setspace}
```

```
\begin{document}
```

```
\doublespacing
```

```
This document is now doublespaced.\\  
See.
```

```
\end{document}
```

- ▶ by default,  $\text{\LaTeX}$  uses large margins and single spacing as the optimal format

- ▶ by default,  $\text{\LaTeX}$  uses large margins and single spacing as the optimal format
- ▶ to get regular margins, use the `fullpage` package

# sectioning

$\text{\LaTeX}$  can divide out document into sections with titles using `\section` (with numbers) or `\section*` (without numbers).

# sectioning

$\text{\LaTeX}$  can divide out document into sections with titles using `\section` (with numbers) or `\section*` (without numbers).

```
\section{My First Section Title}  
Text here
```

```
\section{My Second Section Title}  
More text here.
```

# sectioning

L<sup>A</sup>T<sub>E</sub>X can divide out document into sections with titles using `\section` (with numbers) or `\section*` (without numbers).

```
\section{My First Section Title}  
Text here
```

```
\section{My Second Section Title}  
More text here.
```

`\subsection` and `\subsubsection` also available.



# sectioning

L<sup>A</sup>T<sub>E</sub>X can divide out document into sections with titles using `\section` (with numbers) or `\section*` (without numbers).

```
\section{My First Section Title}  
Text here
```

```
\section{My Second Section Title}  
More text here.
```

`\subsection` and `\subsubsection` also available. Try and compile!

# environments

# environments

Think of environments as creating space in your document for certain activities.

# environments

Think of environments as creating space in your document for certain activities. Environments must always begin and end.

```
\begin{<environment>}
```

Stuff here!

```
\end{<environment>}
```

# environments

Think of environments as creating space in your document for certain activities. Environments must always begin and end.

```
\begin{<environment>}
```

Stuff here!

```
\end{<environment>}
```

document, verbatim, equation, eqnarray, table, tabular,  
figure, center, itemize, enumerate  $\in$  <environment>

# lists

Let's create a list environment with `itemize` or `enumerate`:

# lists

Let's create a list environment with `itemize` or `enumerate`:

My favorite drinks:

```
\begin{itemize}
\item Barq's Root Beer
\item Dr. Pepper
\item Orange Soda
\end{itemize}
```

# lists

Let's create a list environment with `itemize` or `enumerate`:

My favorite drinks:

```
\begin{itemize}
\item Barq's Root Beer
\item Dr. Pepper
\item Orange Soda
\end{itemize}
```

What's the difference between the two?



## math mode

One of the major advantages of  $\text{\LaTeX}$  is in typing math.

## math mode

One of the major advantages of  $\text{\LaTeX}$  is in typing math.

There are many ways to go to math mode. I like:

# math mode

One of the major advantages of  $\text{\LaTeX}$  is in typing math.

There are many ways to go to math mode. I like:

- ▶  $\$$  for inline math

# math mode

One of the major advantages of  $\text{\LaTeX}$  is in typing math.

There are many ways to go to math mode. I like:

- ▶  $\$$  for inline math
- ▶ `eqnarray` environment for centered math and equations

# math mode

One of the major advantages of  $\text{\LaTeX}$  is in typing math.

There are many ways to go to math mode. I like:

- ▶ `$` for inline math
- ▶ `eqnarray` environment for centered math and equations

Others include:

- ▶ `$$`
- ▶ `equation` environment for one equation only
- ▶ `displaymath` environment

For math within a line such as  $\alpha$  or  $5 > 4$ , enclose the math in dollar signs:

For math within a line such as  $\alpha$  or  $5 > 4$ , enclose the math in dollar signs:

For math within a line such as  $\alpha$  or  $5 > 4$ , enclose the math in dollar signs:

For equations or centered math, use the `eqnarray` environment.



For equations or centered math, use the `eqnarray` environment.  
You can align using `&`.

For equations or centered math, use the `eqnarray` environment.  
You can align using `&`.

$$5 + 4 = 9 \qquad (1)$$

$$3 + 2 = 5 \qquad (2)$$

For equations or centered math, use the `eqnarray` environment.  
You can align using `&`.

$$5 + 4 = 9 \tag{1}$$

$$3 + 2 = 5 \tag{2}$$

```
\begin{eqnarray}
5+4 &=& 9 \\
3+2 &=& 5
\end{eqnarray}
```

other math stuff

## other math stuff

- ▶ Greek symbols are intuitive: `\delta` `\Delta`

## other math stuff

- ▶ Greek symbols are intuitive: `\delta` `\Delta`
- ▶ scripting is easy:  $x_{ij} + x^2$  is `x_{ij}+x^2`

## other math stuff

- ▶ Greek symbols are intuitive: `\delta` `\Delta`
- ▶ scripting is easy:  $x_{ij} + x^2$  is `x_{ij}+x^2`
- ▶ bolding math is with `\mathbf{<text>}` for non-Greek symbols and `\bm{<symbol>}` in the `bm` package for Greek symbols

## other math stuff

- ▶ Greek symbols are intuitive: `\delta` `\Delta`
- ▶ scripting is easy:  $x_{ij} + x^2$  is `x_{ij}+x^2`
- ▶ bolding math is with `\mathbf{<text>}` for non-Greek symbols and `\bm{<symbol>}` in the `bm` package for Greek symbols
- ▶ fractions are `\frac{<numerator>}{<denominator>}`



## other math stuff

- ▶ Greek symbols are intuitive: `\delta` `\Delta`
- ▶ scripting is easy:  $x_{ij} + x^2$  is `x_{ij}+x^2`
- ▶ bolding math is with `\mathbf{<text>}` for non-Greek symbols and `\bm{<symbol>}` in the `bm` package for Greek symbols
- ▶ fractions are `\frac{<numerator>}{<denominator>}`
- ▶ square roots are `\sqrt{<number>}`

## other math stuff

- ▶ Greek symbols are intuitive: `\delta` `\Delta`
- ▶ scripting is easy:  $x_{ij} + x^2$  is `x_{ij}+x^2`
- ▶ bolding math is with `\mathbf{<text>}` for non-Greek symbols and `\bm{<symbol>}` in the `bm` package for Greek symbols
- ▶ fractions are `\frac{<numerator>}{<denominator>}`
- ▶ square roots are `\sqrt{<number>}`
- ▶ many many other things you can do

## other math stuff

- ▶ Greek symbols are intuitive: `\delta` `\Delta`
- ▶ scripting is easy:  $x_{ij} + x^2$  is `x_{ij}+x^2`
- ▶ bolding math is with `\mathbf{<text>}` for non-Greek symbols and `\bm{<symbol>}` in the `bm` package for Greek symbols
- ▶ fractions are `\frac{<numerator>}{<denominator>}`
- ▶ square roots are `\sqrt{<number>}`
- ▶ many many other things you can do
- ▶ use google to search!

Try typing the following equations:

$$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

# tables

Tables are created using the `tabular` environment.

# tables

Tables are created using the `tabular` environment.

```
\begin{tabular}{l|c|r}  
year & country & leader \\  
\hline  
2009 & US & Obama \\  
2009 & UK & Brown \\  
\hline  
\end{tabular}
```

# tables

Tables are created using the tabular environment.

```
\begin{tabular}{l|c|r}  
year & country & leader \\  
\hline  
2009 & US & Obama \\  
2009 & UK & Brown \\  
\hline  
\end{tabular}
```

year	country	leader
2009	US	Obama
2009	UK	Brown

# tables

Tables are created using the tabular environment.

```
\begin{tabular}{l|c|r}  
year & country & leader \\  
\hline  
2009 & US & Obama \\  
2009 & UK & Brown \\  
\hline  
\end{tabular}
```

year	country	leader
2009	US	Obama
2009	UK	Brown

- ▶ {} arguments after \begin{tabular} specify number of columns and alignment



# tables

Tables are created using the `tabular` environment.

```
\begin{tabular}{l|c|r}  
year & country & leader \\  
\hline  
2009 & US & Obama \\  
2009 & UK & Brown \\  
\hline  
\end{tabular}
```

year	country	leader
2009	US	Obama
2009	UK	Brown

- ▶ `{}` arguments after `\begin{tabular}` specify number of columns and alignment
  - ▶ use `|` for vertical dividers.

# tables

Tables are created using the tabular environment.

```
\begin{tabular}{l|c|r}  
year & country & leader \\  
\hline  
2009 & US & Obama \\  
2009 & UK & Brown \\  
\hline  
\end{tabular}
```

year	country	leader
2009	US	Obama
2009	UK	Brown

- ▶ `{}` arguments after `\begin{tabular}` specify number of columns and alignment
  - ▶ use `|` for vertical dividers.
- ▶ `&` is used to specify column breaks

# tables

Tables are created using the `tabular` environment.

```
\begin{tabular}{l|c|r}  
year & country & leader \\  
\hline  
2009 & US & Obama \\  
2009 & UK & Brown \\  
\hline  
\end{tabular}
```

year	country	leader
2009	US	Obama
2009	UK	Brown

- ▶ `{}` arguments after `\begin{tabular}` specify number of columns and alignment
  - ▶ use `|` for vertical dividers.
- ▶ `&` is used to specify column breaks
- ▶ use `\hline` for horizontal lines

Enclose the table in a `table` environment for captions and placement options (`[!http]`).

Enclose the table in a table environment for captions and placement options ([!htp]).

```
\begin{table}[!htp]
\caption{table caption}
\begin{tabular}{l|c|r}
year & country & leader \\
\hline
2009 & US & Obama \\
2009 & UK & Brown \\
\hline
\end{tabular}
\end{table}
```

Enclose the table in a table environment for captions and placement options ([!htp]).

```
\begin{table}[!htp]
\caption{table caption}
\begin{tabular}{l|c|r}
year & country & leader \\
\hline
2009 & US & Obama \\
2009 & UK & Brown \\
\hline
\end{tabular}
\end{table}
```

Table: table caption

year	country	leader
2009	US	Obama
2009	UK	Brown

Enclose the table in a table environment for captions and placement options ([!htp]).

```
\begin{table}[!htp]
\caption{table caption}
\begin{tabular}{l|c|r}
year & country & leader \\
\hline
2009 & US & Obama \\
2009 & UK & Brown \\
\hline
\end{tabular}
\end{table}
```

Table: table caption

year	country	leader
2009	US	Obama
2009	UK	Brown

Try creating your own tables!

# graphics

Include graphics (plots, pictures, etc.) with the `graphicx` package and `\includegraphics`.



# graphics

Include graphics (plots, pictures, etc.) with the `graphicx` package and `\includegraphics`.

```
\includegraphics[<options>]{myfilename.jpg}
```

# graphics

Include graphics (plots, pictures, etc.) with the `graphicx` package and `\includegraphics`.

```
\includegraphics[<options>]{myfilename.jpg}
```

- ▶ adjust the size in options (i.e. `[scale = .5]` or `[width = 2in, height=2in]`)

# graphics

Include graphics (plots, pictures, etc.) with the `graphicx` package and `\includegraphics`.

```
\includegraphics[<options>]{myfilename.jpg}
```

- ▶ adjust the size in options (i.e. `[scale = .5]` or `[width = 2in, height=2in]`)
- ▶ for compiling with PDFLaTeX, file must be `.jpg`, `.pdf`, or `.png`

# graphics

Include graphics (plots, pictures, etc.) with the `graphicx` package and `\includegraphics`.

```
\includegraphics[<options>]{myfilename.jpg}
```

- ▶ adjust the size in options (i.e. `[scale = .5]` or `[width = 2in, height=2in]`)
- ▶ for compiling with PDFLaTeX, file must be `.jpg`, `.pdf`, or `.png`
- ▶ for compiling with LaTeX, file must be `.ps` or `.eps`

# graphics

Include graphics (plots, pictures, etc.) with the `graphicx` package and `\includegraphics`.

```
\includegraphics[<options>]{myfilename.jpg}
```

- ▶ adjust the size in options (i.e. `[scale = .5]` or `[width = 2in, height=2in]`)
- ▶ for compiling with PDFLaTeX, file must be `.jpg`, `.pdf`, or `.png`
- ▶ for compiling with LaTeX, file must be `.ps` or `.eps`
- ▶ embed graphic in a `figure` environment for captions and placement (similar to `table`)

# floats

The `table` and `figure` environments are known as *floats*, which are objects that cannot be broken up into multiple pages.

# floats

The `table` and `figure` environments are known as *floats*, which are objects that cannot be broken up into multiple pages.

Floats have three qualities that you should know:

# floats

The `table` and `figure` environments are known as *floats*, which are objects that cannot be broken up into multiple pages.

Floats have three qualities that you should know:

- ▶ placement
- ▶ caption
- ▶ labels and cross-referencing



placement:

placement:

- ▶ h: here
- ▶ t: top of page
- ▶ b: bottom of page
- ▶ p: special page for floats
- ▶ !: override L<sup>A</sup>T<sub>E</sub>X defaults

placement:

- ▶ h: here
- ▶ t: top of page
- ▶ b: bottom of page
- ▶ p: special page for floats
- ▶ !: override  $\text{\LaTeX}$  defaults

$\text{\LaTeX}$  will try to accomodate in order.

placement:

- ▶ h: here
- ▶ t: top of page
- ▶ b: bottom of page
- ▶ p: special page for floats
- ▶ !: override L<sup>A</sup>T<sub>E</sub>X defaults

L<sup>A</sup>T<sub>E</sub>X will try to accomodate in order.

```
\begin{figure}[!htp]  
\includegraphics{myfilename.jpg}  
\end{figure}
```

caption:

caption:

- ▶ use the `caption` command

caption:

- ▶ use the `caption` command
- ▶ can be placed below or above

caption:

- ▶ use the `caption` command
- ▶ can be placed below or above

```
\begin{figure}[!htp]  
\includegraphics{myfilename.jpg}  
\caption{my caption here}  
\end{figure}
```



labels and cross-referencing:

labels and cross-referencing:

- ▶  $\text{\LaTeX}$  will keep an internal numbering of the floats (separate counts for tables and figures)

labels and cross-referencing:

- ▶  $\text{\LaTeX}$  will keep an internal numbering of the floats (separate counts for tables and figures)
- ▶ if you have captions,  $\text{\LaTeX}$  will number your floats for you (i.e. Figure 1: my caption)

labels and cross-referencing:

- ▶  $\text{\LaTeX}$  will keep an internal numbering of the floats (separate counts for tables and figures)
- ▶ if you have captions,  $\text{\LaTeX}$  will number your floats for you (i.e. Figure 1: my caption)
- ▶ use the `\label` and `\ref` commands to refer to the float number in your text

```
\begin{figure}[!htp]
\includegraphics{myfilename.jpg}
\caption{my caption here}
\label{<key>}
\end{figure}
```

In Figure \ref{<key>}, we can see that...

```
\begin{figure}[!htp]  
\includegraphics{myfilename.jpg}  
\caption{my caption here}  
\label{<key>  
\end{figure}
```

In Figure \ref{<key>}, we can see that...

- ▶ \label must be below \caption

```
\begin{figure}[!htp]
\includegraphics{myfilename.jpg}
\caption{my caption here}
\label{<key>}
\end{figure}
```

In Figure \ref{<key>}, we can see that...

- ▶ \label must be below \caption
- ▶ if figure is Figure 1, then text will show "In Figure 1, we can see that..."

```
\begin{figure}[!htp]
\includegraphics{myfilename.jpg}
\caption{my caption here}
\label{<key>}
\end{figure}
```

In Figure \ref{<key>}, we can see that...

- ▶ \label must be below \caption
- ▶ if figure is Figure 1, then text will show "In Figure 1, we can see that..."
- ▶ <key> is any word, phrase, alphanumeric indicator you want



```
\begin{figure}[!htp]
\includegraphics{myfilename.jpg}
\caption{my caption here}
\label{<key>}
\end{figure}
```

In Figure \ref{<key>}, we can see that...

- ▶ \label must be below \caption
- ▶ if figure is Figure 1, then text will show "In Figure 1, we can see that..."
- ▶ <key> is any word, phrase, alphanumeric indicator you want
- ▶ may have to compile more than once to get the references to show up correctly

# verbatim

One particular environment useful for pasting R code is the `verbatim` environment.

# verbatim

One particular environment useful for pasting R code is the `verbatim` environment.

```
\begin{verbatim}
```

```
LaTeX will copy everything in the verbatim  
environment exactly, rather than interpret it as code.  
For example, \begin{itemize}  
here is typed out exactly rather than beginning a list.  
\end{ verbatim}
```

# fonts

# fonts

- ▶ define the font size for the document in as an option in `\documentclass`:

# fonts

- ▶ define the font size for the document in as an option in  
`\documentclass: \documentclass[12pt]{article}`

# fonts

- ▶ define the font size for the document in as an option in `\documentclass: \documentclass[12pt]{article}`
- ▶ change font size in text using `\tiny`, `\scriptsize`, `\footnotesize`, `\small`, `\normalsize`, `\large`, `\Large`, `\LARGE`, `\huge`, `\Huge`

# fonts

- ▶ define the font size for the document in as an option in `\documentclass: \documentclass[12pt]{article}`
- ▶ change font size in text using `\tiny`, `\scriptsize`, `\footnotesize`, `\small`, `\normalsize`, `\large`, `\Large`, `\LARGE`, `\huge`, `\Huge`
- ▶ **colored fonts** using color package



# fonts

- ▶ define the font size for the document in as an option in `\documentclass`: `\documentclass[12pt]{article}`
- ▶ change font size in text using `\tiny`, `\scriptsize`, `\footnotesize`, `\small`, `\normalsize`, `\large`, `\Large`, `\LARGE`, `\huge`, `\Huge`
- ▶ **colored fonts** using color package
- ▶ fonts in math mode are different

# footnotes

use the `\footnote` command<sup>1</sup>

---

<sup>1</sup>like this

# footnotes

use the `\footnote` command<sup>1</sup>

use the `{\tt $\backslash$footnote}`  
`command\footnote{like this}`

---

<sup>1</sup>like this

# bibliography

Compiling a bibliography is simple in  $\text{\LaTeX}$ .

# bibliography

Compiling a bibliography is simple in  $\text{\LaTeX}$ .

- ▶ need to use the natbib package

# bibliography

Compiling a bibliography is simple in  $\text{\LaTeX}$ .

- ▶ need to use the `natbib` package
- ▶ need a separate file with `.bib` extension in the same directory

# bibliography

Compiling a bibliography is simple in  $\text{\LaTeX}$ .

- ▶ need to use the natbib package
- ▶ need a separate file with .bib extension in the same directory
- ▶ in your .tex file, need the line `\bibliography{<name of  
bib file w/o .bib extension>}`

# bibliography

Compiling a bibliography is simple in  $\text{\LaTeX}$ .

- ▶ need to use the natbib package
- ▶ need a separate file with .bib extension in the same directory
- ▶ in your .tex file, need the line `\bibliography{<name of bib file w/o .bib extension>}`
- ▶ also in your .tex file, need the line `\bibliographystyle{<name of a style>}`



# bibliography

Compiling a bibliography is simple in  $\text{\LaTeX}$ .

- ▶ need to use the natbib package
- ▶ need a separate file with .bib extension in the same directory
- ▶ in your .tex file, need the line `\bibliography{<name of bib file w/o .bib extension>}`
- ▶ also in your .tex file, need the line `\bibliographystyle{<name of a style>}`
  - ▶ styles may be downloaded, including an apsr style

in .bib file, for each reference, need something like

```
@Article{<key>,  
author = {},  
title = {},  
journal = {},  
year = {},  
}
```

in .bib file, for each reference, need something like

```
@Article{<key>,  
author = {},  
title = {},  
journal = {},  
year = {},  
}
```

- ▶ <key> is a unique identifier for each reference that you will refer to in your .tex file

in .bib file, for each reference, need something like

```
@Article{<key>,  
author = {},  
title = {},  
journal = {},  
year = {},  
}
```

- ▶ <key> is a unique identifier for each reference that you will refer to in your .tex file
- ▶ your editor should have these fields preset somewhere

in .bib file, for each reference, need something like

```
@Article{<key>,  
author = {},  
title = {},  
journal = {},  
year = {},  
}
```

- ▶ <key> is a unique identifier for each reference that you will refer to in your .tex file
- ▶ your editor should have these fields preset somewhere

in .tex file, use `\citep{<key>}`, `\citet{<key>}`, or `\citen{<key>}` where you want to reference a certain work

when compiling

when compiling

1. LaTeX or PDFLatex once

when compiling

1. LaTeX or PDFLatex once
2. BibTeX once (should be a button or compiling command)



when compiling

1. LaTeX or PDFLatex once
2. BibTeX once (should be a button or compiling command)
3. LaTeX or PDFLatex once

when compiling

1. LaTeX or PDFLatex once
2. BibTeX once (should be a button or compiling command)
3. LaTeX or PDFLatex once
4. LaTeX or PDFLatex once more

# presentations

Powerpoint-like presentations can be done using a document class called beamer (may need to be downloaded).

# presentations

Powerpoint-like presentations can be done using a document class called beamer (may need to be downloaded).

- ▶ can choose a color theme (default theme used here)

# presentations

Powerpoint-like presentations can be done using a document class called `beamer` (may need to be downloaded).

- ▶ can choose a color theme (default theme used here)
- ▶ same basic code

# presentations

Powerpoint-like presentations can be done using a document class called `beamer` (may need to be downloaded).

- ▶ can choose a color theme (default theme used here)
- ▶ same basic code
- ▶ each slide is a `frame` environment

# presentations

Powerpoint-like presentations can be done using a document class called `beamer` (may need to be downloaded).

- ▶ can choose a color theme (default theme used here)
- ▶ same basic code
- ▶ each slide is a `frame` environment
- ▶ use `\pause` between clicks

**anything** can be done using  $\text{\LaTeX}$  with the right package



you will learn much much much more along the way

google is your friend

## exercises

Include the following into one .pdf document:

- ▶ the following equations:

$$\begin{aligned} p(\theta|\mathbf{y}) &\propto p(\mathbf{y}|\theta)p(\theta) \\ &= \exp \left[ -\frac{1}{2\sigma^2\tau_0^2} \left( \theta^2 (\sigma^2 + \tau_0^2 n) - 2\theta (\mu_0\sigma^2 + \tau_0^2 n\bar{y}) \right) \right] \end{aligned}$$

- ▶ a  $5 \times 3$  table of 5 Harvard political scientists (or other academics) with their names, field or subfield, and office room number
- ▶ a picture of your favorite celebrity with an appropriate caption
- ▶ a list of your 3 favorite things about Harvard so far