Workshop: A Simple Introduction to LaTeX

Michael P. Fix
Susanne Schorpp
Georgia State University

29 March 2019

Outline

What is LaTeX?

- LaTeX is an alternative to WYSIWYG word processors for creating documents.
- LaTeX allows you to focus on writing your document without spending hours formatting.
- You simply provide a few logical commands to define the structure of your document, and let LaTeX design and TeX typeset your work.

Why You Should Use LaTeX

- Your documents look professionally printed.
- While it requires a bit of time investment upfront to learn, it will save you many wasted hours later.
- You can typeset elegant mathematical formulae with simple commands.
- Everything is free!

Basic Structure of the Input File

For a file to be correctly processed by LaTeX there are a few essential components that must be in the file.

1. Defining the document class
2. Loading Packages
3. Beginning and ending the document

Document Class

Definition

The Document Class specifies the type of document you intent to write. For example, most stuff can be done with the `article` class, the `book` class is for longer documents, and the `beamer` class is used for presentations (like this one).

Document Class

- Every file must begin by specifying the document class with the following command:

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

- This can be as simple as defining the class:

  ```
  \documentclass{article}
  ```

- Or can be used to specify font size, paper type, etc. For example:

  ```
  \documentclass[11pt, letter]{article}
  ```

- Would specify that you wanted to write an article in 11 point font using letter paper. (Note: This is what I usually use.)

Packages

Definition

Basic LaTeX often does not include everything you need to include in a document. For example, if you wish to include graphics, most mathematical symbols, colored text, or various other things, you will have to use Packages. Packages allow you to expand the functionality of LaTeX.

Packages

- To activate a package, the following command is used:

  ```
  \usepackage[options]{package}
  ```

- For example, in order to typeset many mathematical symbols and formulae you need to load the follow two packages:

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

- Note: options are rarely used with packages.

Identifying Information

- The last items generally included in the preamble to your file are the identifying information.
- This is done with the following three commands:

```
\title{The Title}
\author{Your Name}
\date{The Date}
```

or

```
\date{\today}
```

The Body of the Document

- With the preamble to your input file complete, you are now ready to begin writing the body of the document. It is important to tell LaTeX when to begin the document with the following command:

  ```
  \begin{document}
  ```

- . . . and when to end the document with this command:

  ```
  \end{document}
  ```

The Body of the Document

- Simply insert the following command at the beginning of your document to add the title, author info, and date:

  `\maketitle`

- Note: You have to insert the `\maketitle` command after `\begin{document}` or your title and name will not appear.

A Simple Document

Now that we've got the basics down, we can put it all together to create a simple document. Try this:

```
\documentclass{article}
\title{A Short Paper}
\author{Me}
\date{\today}
\begin{document}
\maketitle
This is a very short paper! I should get an A for
being concise.
\end{document}
```

A Simple Document

If you were successful your document should appear like this:

A Short Paper

Me

July 14, 2009

This is a very short paper! I should get an A for being concise.

A Technical Note — File Types

Each time you create a new document, several files will be created. The most important of these is your input file (i.e. the file you create in your text editor).

A Technical Note — File Types

Each time you create a new document, several files will be created. The most important of these is your input file (i.e. the file you create in your text editor).

This file will look something like:

`filename.tex`

A Technical Note — File Types

Each time you create a new document, several files will be created. The most important of these is your input file (i.e. the file you create in your text editor).

This file will look something like:

`filename.tex`

Your output file will be saved in the same folder as something like:

`filename.pdf`

A Technical Note — File Types

Each time you create a new document, several files will be created. The most important of these is your input file (i.e. the file you create in your text editor).

This file will look something like:

`filename.tex`

Your output file will be saved in the same folder as something like:

`filename.pdf`

There will also be numerous other files created with extensions like:

`filename.log`, `filename.aux`, `etc.`

Writing Text for Your Document

- One of the best things about creating LaTeX documents is the simplicity of writing.
- Since you are working with a text editor, you are not overwhelmed with the graphical interface of WYSIWYG word processors.
- Instead of hunting through menu bars to find everything you need to properly typeset your document, you just need a few simple (and easy to look up) commands to instruct the program and it does all the work for you.
- That is what is section of the presentation is all about . . .

Spacing

Spacing in LaTeX is largely automatic. Like a professionally printed book or journal, it ensures relatively uniform spacing between characters, words, lines, and paragraphs. Thus you do not need to worry about these details. But there are a few things you must keep in mind.

The Essentials for Proper Spacing I

First, you must note that spacing between words is treat uniformly. This regardless of whether you type:

```
The dog ran fast.
```

or

```
The   dog   ran        fast.
```

or

```
The                  dog ran                  fast.
```

The Essentials for Proper Spacing I

First, you must note that spacing between words is treat uniformly. This regardless of whether you type:

```
The dog ran fast.
```

or

```
The   dog   ran        fast.
```

or

```
The                    dog ran                    fast.
```

The output will be the same:
The dog ran fast.

The Essentials for Proper Spacing II

For paragraphs, LaTeX also automatically intents the first line and provides proper spacing. You simply need to let it know you want to start a new paragraph by leaving a blank line in your input file. For example:

```
This is a paragraph.

This is the next paragraph.
```

instead of:

```
This is a paragraph.
This is the next paragraph.
```

Emphasizing Text

Sometimes it is necessary to *italicize* or **bold face** text.
To do this you use the following commands:

```
{\it  . . . }
```

for italicized text.

Or

```
 {\bf  . . . }
```

for bold faced text.

Special Characters

LaTeX has commands for almost any special character that you could ever need. Here are some common examples.

Special Characters

- Quotation Marks: you must use two ‘ to open and two ’ to close
- Elipses: `\ldots`
- Alost many symbols such as #, &, and $ are used for commands, to typeset these, simply put a \ in front of the symbol

Section Headings

- LaTeX supports various ways to subdivide your document into sections. The most common of these are:

  ```
  \section{ . . . }   or   \section*{ . . . }
  \subsection{ . . . }  or  \subsection*{ . . . }
  \subsubsection{ . . . }  or  \subsubsection*{ . . . }
  ```

- Note: the `section`, `subsection`, and `subsubsection` commands will automatically number the sections. The `section*`, `subsection*`, and `subsubsection*` commands print only the name of the section.

Footnotes

Inserting footnotes is extremely simple. Simply insert the following command where you would like the footnote marker to appear.

```
\footnote{ . . . }
```

Footnotes

Inserting footnotes is extremely simple. Simply insert the following command where you would like the footnote marker to appear.

```
\footnote{ . . . }
```

. . . and simply type the text of the footnote inside the braces.

Environments

Definition

Environments are used in LaTeX for writing special kinds of text. Think of them as creating a special section of your document that follows a different set of rules.

To use an environment, you use the following commands:

`\begin{environment}` and `\end{environment}`

. . . with the text entered between the two commands.

Environments – Lists

- Making lists in LaTeX is very quick and easy.
- Additionally, you control the nesting of lists, so you never have to deal with the auto-formatting errors in *MS Word*.
- The two most common types of lists use the `enumerate` environment (for a numbered list) or the `itemize` environment (for items where item are identified by a symbol).

Environments – Lists

Try the following example:

```
\documentclass{article}
\title{A Simple List}
\author{Me}
\begin{document}
\maketitle
\begin{enumerate}
\item This is the first item.
\item This is the second item.
   \begin{itemize}
   \item I like the second item.
   \item {\it It is good}.
   \end{itemize}
\item This is the third --- and last --- item.
\end{enumerate}
\end{document}
```

Environments – Lists

You document should look like this:

A Simple List

Me

October 11, 2011

1. This is the first item.
2. This is the second item.
   - I like the second item.
   - *It is good.*
3. This is the third — and last – item.

Environments – Others

- There are many additional environments available
- Excluding ones for mathematics some of the more common are:
  - `doublespacing` to doublespace text
  - `quote` for block quotes
  - `abstract` to make an abstract
  - `titlepage` to define your title page
  - `flushleft`, `flushright`, and `center` to set alignment
  - `tabular` is used to create tables. We will discuss this later.
- Note: Environments can be nested

Introduction to Typesetting Math in LaTeX

- As I'm sure you are well aware, the ability to typeset mathematical formulae with an elegance unmatched by any other program is the greatest strength of LaTeX.

Introduction to Typesetting Math in LaTeX

- As I'm sure you are well aware, the ability to typeset mathematical formulae with an elegance unmatched by any other program is the greatest strength of LaTeX.
- It would be impossible to show you everything you can possibly do, without spending days waking through example after example.

Introduction to Typesetting Math in LaTeX

- As I'm sure you are well aware, the ability to typeset mathematical formulae with an elegance unmatched by any other program is the greatest strength of LaTeX.
- It would be impossible to show you everything you can possibly do, without spending days waking through example after example.
- Thus, I am simply going to provide a brief overview to show you how it's done.

Introduction to Typesetting Math in LaTeX

- As I'm sure you are well aware, the ability to typeset mathematical formulae with an elegance unmatched by any other program is the greatest strength of LaTeX.
- It would be impossible to show you everything you can possibly do, without spending days waking through example after example.
- Thus, I am simply going to provide a brief overview to show you how it's done.
- Fortunately, there are countless reference sources available for anything you could ever need to do.

A Note on Packages

- Just about any mathematical symbol or formulae you could ever need to created is contained in $\mathcal{AMS}$-LaTeX.

A Note on Packages

- Just about any mathematical symbol or formulae you could ever need to created is contained in $\mathcal{AMS}$-LaTeX.
- This was created by the *American Mathematical Society* and is as comprehensive as possible – it is also how they create all of their documents.

A Note on Packages

- Just about any mathematical symbol or formulae you could ever need to created is contained in $\mathcal{AMS}$-LaTeX.
- This was created by the *American Mathematical Society* and is as comprehensive as possible – it is also how they create all of their documents.
- As it is included in any basic installation of LaTeX, you do not need to install anything extra.

A Note on Packages

- Just about any mathematical symbol or formulae you could ever need to created is contained in $\mathcal{AMS}$-LaTeX.
- This was created by the *American Mathematical Society* and is as comprehensive as possible – it is also how they create all of their documents.
- As it is included in any basic installation of LaTeX, you do not need to install anything extra.
- Just remember to load two package every time you start:

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

A Note on Environments

All mathematical formulae and equations must be written inside special environments to appear correctly in your output file.

A Note on Environments

All mathematical formulae and equations must be written inside special environments to appear correctly in your output file.

The two most common are: `equation` **for numbered equations and** `equation*` **for unnumbered ones.**

A Note on Environments

All mathematical formulae and equations must be written inside special environments to appear correctly in your output file.

The two most common are: `equation` for numbered equations and `equation*` for unnumbered ones.

Alternatively, if you wish to include mathematical symbols or formulas in the body of your text, you should use:

`$ . . . $`

Math is Easy as π

We now know everything we need to know to typeset simple equations in LaTeX.

Math is Easy as π

We now know everything we need to know to typeset simple equations in LaTeX.

Let's try some examples.

Example 1: Let's say we want to include some math in the body of a paragraph:

```
Einstein's most famous equation is $e = mc ^ 2$.\\
```

Math is Easy as π

Example 2: Let's say we want to set off an important equation, but we do not want it to be numbered:

```
Another important contribution to scientific knowledge
is the Pythagorean Theorem.\\
\begin{equation*}
a^2 + b^2 = c^2
\end{equation*}
```

Math is Easy as π

Example 3: Finally, let's say we need to use multiple equations and we want to number them.

```
Since we know that:
\begin{equation}
3+3=6
\end{equation}
and we know that:
\begin{equation}
2*3 =6
\end{equation}
Then, from (1) and (2), we know that:
\begin{equation}
2*3 =3+3
\end{equation}
```

A Note on Importing Figures

Importing figures into your LaTeX is very simple.

A Note on Importing Figures

Importing figures into your LaTeX is very simple.

Figures are made within the `figure` environment:

```
\begin{figure}[placement]
```

A Note on Importing Figures

Importing figures into your LaTeX is very simple.

Figures are made within the `figure` environment:

`\begin{figure}[placement]`

Where the `placement` option takes the following values:
`h` here (the exact location)
`t` at the top of the page
`b` at the bottom of the page
`p` on a special page for floating bodies

A Note on Importing Figures

Importing figures into your LaTeX is very simple.

Figures are made within the `figure` environment:

```
\begin{figure}[placement]
```

Where the `placement` option takes the following values:
`h` here (the exact location)
`t` at the top of the page
`b` at the bottom of the page
`p` on a special page for floating bodies

Note: the placement option is often left blank with the default `[tbp]`.

A Note on Importing Figures

An example should clarify:

A Note on Importing Figures

An example should clarify:

Say we to import a picture into our document. We could use the following commands:

```
\begin{figure}
\centering
\includegraphics[scale=.5]{baby.jpg}
\caption{Cute Kid!}
\end{figure}
```

A Note on Importing Figures

. . . to get:

Figure: Cute Kid!

A Note on Importing Figures

Remember to keep in mind that for this to work the file you want to import must be located in the proper folder.

A Note on Importing Figures

Remember to keep in mind that for this to work the file you want to import must be located in the proper folder.

This generally requires that it be stored in the same folder as the `.tex` file

A Note on Importing Figures

Remember to keep in mind that for this to work the file you want to import must be located in the proper folder.

This generally requires that it be stored in the same folder as the `.tex` file Finally, note that his only works with certain file types.

While not an exhaustive list, some of the most common include `pdf`, `jpg`, `png`, and `eps`.

Making Tables in LaTeX

Making tables in LaTeX requires a relatively simple series of commands, however it is important that you follow a few simple rules.

Making Tables in LaTeX

Making tables in LaTeX requires a relatively simple series of commands, however it is important that you follow a few simple rules.

Let's start with how to make a very simple 2 x 2 table like the one below:

	x_1	x_2
y_1	.4	.6
y_2	.6	.4

Making Tables in LaTeX

The first thing that we need to keep in mind when making table is that they are made inside the `tabular` environment.

```
\begin{tabular}{specifications}
```

Making Tables in LaTeX

The first thing that we need to keep in mind when making table is that they are made inside the `tabular` environment.

```
\begin{tabular}{specifications}
```

Unlike many other environments, we must make an additional declaration in in `\begin{environment}` command. Specifically, we must define the specifications for our table.

Making Tables in LaTeX – Specifications

In defining the specifications for our table, we must define three things:

Making Tables in LaTeX – Specifications

In defining the specifications for our table, we must define three things:

1. The number of columns.

Making Tables in LaTeX – Specifications

In defining the specifications for our table, we must define three things:

1. The number of columns.
2. The alignment of each column.

Making Tables in LaTeX – Specifications

In defining the specifications for our table, we must define three things:

1. The number of columns.
2. The alignment of each column.
3. Whether we want vertical lines on the left margin, right margin, and/or between any of the columns.

Making Tables in LaTeX – Specifications

We use the following format to declare our table specification:

```
\begin{tabular}{  l   c   c  }
```

Making Tables in LaTeX – Specifications

We use the following format to declare our table specification:

```
\begin{tabular}{  l   c   c  }
```

Each entry represents a column. Thus the above example would be for a table with three columns.

Making Tables in LaTeX – Specifications

We use the following format to declare our table specification:

```
\begin{tabular}{  l   c   c  }
```

Each entry represents a column. Thus the above example would be for a table with three columns.

The letters tell us how we want each column aligned, and we use:

`l`	left-aligned column
`c`	centered column
`r`	right-aligned column

Making Tables in LaTeX – Specifications

We use the following format to declare our table specification:

```
\begin{tabular}{  l  c  c  }
```

Each entry represents a column. Thus the above example would be for a table with three columns.

The letters tell us how we want each column aligned, and we use:

`l`	left-aligned column
`c`	centered column
`r`	right-aligned column

Thus in the example, the first column is left-aligned, the second is centered, and the last is also centered.

Making Tables in LaTeX – Specifications

Finally, we must also declare any vertical lines we would like to have appear in our table.

Making Tables in LaTeX – Specifications

Finally, we must also declare any vertical lines we would like to have appear in our table.

To do this, we use the following symbols:

|	vertical line
||	double vertical line

Making Tables in LaTeX – Specifications

Finally, we must also declare any vertical lines we would like to have appear in our table.

To do this, we use the following symbols:

`|`	vertical line
`||`	double vertical line

Since our example has a line on each side for the edge of the table and a line between the first and second column, we must declare this in our specifications:

```
\begin{tabular}{ | l | c  c |}
```

Making Tables in LaTeX – Additional Commands

Before we can fill in our table, we need to familiarize ourselves with three more commands.

Making Tables in LaTeX – Additional Commands

Before we can fill in our table, we need to familiarize ourselves with three more commands.

First, we need to know how to make the horizontal lines at the top and bottom of our table and after the first row. To do this we use:

```
\hline
```

Making Tables in LaTeX – Additional Commands

Before we can fill in our table, we need to familiarize ourselves with three more commands.

First, we need to know how to make the horizontal lines at the top and bottom of our table and after the first row. To do this we use:

```
\hline
```

Second, we need a symbol to separate the data for each column within a single row. For this we use:

&

Making Tables in LaTeX – Additional Commands

Before we can fill in our table, we need to familiarize ourselves with three more commands.

First, we need to know how to make the horizontal lines at the top and bottom of our table and after the first row. To do this we use:

`\hline`

Second, we need a symbol to separate the data for each column within a single row. For this we use:

`&`

Finally, we need a symbol to separate the rows of our table. For this we use:

`\\`

Making Tables in LaTeX – Putting It Together I

Let's look at the example together with the code to create it:

	x_1	x_2
y_1	.4	.6
y_2	.6	.4

```
\begin{tabular}{|l|cc|}
\hline
& $x_1$ & $x_2$ \\
\hline
$y_1$ & .4 & .6 \\
$y_2$ & .6 & .4 \\
\hline
\end{tabular}
```

Making Tables in LaTeX – The `table` environment

You can place your table into the `table` environment. That will allow you to assign each table a caption. Latex will also create a running counter for your tables to generate a 'List of Tables'

Making Tables in LaTeX – The `table` environment

You can place your table into the `table` environment. That will allow you to assign each table a caption. Latex will also create a running counter for your tables to generate a 'List of Tables'

```
\begin{table}
\caption{Predicted Probabilities}
\begin{tabular}
. . .
\end{tabular}
\end{table}
```

Making Tables in LaTeX – The table environment

In our case:

Table: Predicted Probabilities

	x_1	x_2
y_1	.4	.6
y_2	.6	.4

```
\begin{table}
\caption{Predicted Probabilities}
\label{Pred}
\begin{tabular}{ | l | c c |}
\hline
& $x_1$ & $x_2$ \\
\hline
$y_1$ & .4 & .6 \\
$y_2$ & .6 & .4 \\
\hline
\end{tabular}
\end{table}
```

Making Tables in LaTeX – A Few Other Things

This should provide a good grasp on the basics of the tabular environment. There are several resources out there that can provide you with any details you may need.

Making Tables in LaTeX – A Few Other Things

This should provide a good grasp on the basics of the tabular environment. There are several resources out there that can provide you with any details you may need.

. . . A good place to start is:
`http://en.wikibooks.org/wiki/LaTeX/Tables`.

Making Tables in LaTeX – A Few Other Things

This should provide a good grasp on the basics of the tabular environment. There are several resources out there that can provide you with any details you may need.

. . . A good place to start is:
`http://en.wikibooks.org/wiki/LaTeX/Tables`.

Remember that you may have to 'load' additional packages in the front end of your document for some features.

Making Tables in LaTeX – A Few Other Things

While it may seem a bit complicated to produce tables in LaTeX, they

Making Tables in LaTeX – A Few Other Things

While it may seem a bit complicated to produce tables in LaTeX, they

1. . . . look professional

Making Tables in LaTeX – A Few Other Things

While it may seem a bit complicated to produce tables in LaTeX, they

1. . . . look professional
2. . . . can save time in the long run

Making Tables in LaTeX – A Few Other Things

Both for STATA and R, commands for automated LaTeX output exist.

Making Tables in LaTeX – A Few Other Things

Both for STATA and R, commands for automated LaTeX output exist.

In other words: You can . . .

Making Tables in LaTeX – A Few Other Things

Both for STATA and R, commands for automated LaTeX output exist.

In other words: You can ...

- Run your model(s)
- Have the program generate the table *for* you
- Copy/paste it into your file
- `https://www.dropbox.com/s/u95comsvrwocu80/usmurder_with_region.dta?dl=0`

Making Tables in LaTeX – STATA

Using the outtex package in STATA:

```
. regress murder south
. outtex, label
%------- Begin LaTeX code -------%
{
\begin{table}[htbp]\centering
 \caption{Estimation results : regress
\label{tabresult regress}}
\begin{tabular}{l c c }\hline\hline
\multicolumn{1}{c}
{\textbf{Variable}}
 & {\textbf{Coefficient}}  & \textbf{(Std. Err.)} \\ \hline
Region: South  &  4.410  & (1.323)\\
Intercept  &  6.356  & (0.573)\\
\hline
\end{tabular}
\end{table}
}
%------- End LaTeX code -------%
```

Making Tables in LaTeX – STATA

Which produces the following table:

Table: Estimation results : regress

Variable	**Coefficient**	**(Std. Err.)**
Region: South	4.410	(1.323)
Intercept	6.356	(0.573)

Making Tables in LaTeX – STATA

The `estout` package in STATA . . .
. . . can produce summary tables:

```
. estpost summarize murder south high poverty single
. esttab, using example.tex cells("count mean sd min max")
```

	(1)				
	count	mean	sd	min	max
Murders per 100,000 Population	48	7.183333	3.944365	1.6	20.3
Region: South	48	.1875	.3944428	0	1
Percent High School Graduates	48	76.575	5.462425	64.6	86.6
Poverty Rate	48	13.71458	4.041223	8	26.4
Single Parent Homes	48	11.04583	1.408781	8.4	14.9

Making Tables in LaTeX – STATA

. . . and Regression output tables:

```
. regress murder south
. estimates store model1
. regress murder high poverty single
. estimates store model2
. estout model1 model2 using example2.tex, ///
cells("b(star fmt(3)) se(fmt(3))") ///
stats(N chi2, star(chi2)) style(tex) label r
```

Making Tables in LaTeX – STATA

. . . would produce the following output in STATA:

```
                    &      model1   &            &      model2   &
                    &           b   &         se&           b   &
Region: South       &       4.410   &      1.323&               &
Percent High School Graduates&      &          &               &      -0.283
Poverty Rate        &               &          &       0.083   &
Single Parent Homes &               &          &       1.757   &
\_cons               &       6.356   &      0.573&       8.275   &
N                   &      48.000   &          &      48.000   &
chi2                &               &          &               &
```

Making Tables in LaTeX – STATA

. . . copy/pasting it into your LaTeX file produces:

	model1		model2	
	b	se	b	se
Region: South	4.410	1.323		
Percent High School Graduates			-0.283	0.089
Poverty Rate			0.083	0.128
Single Parent Homes			1.757	0.246
_cons	6.356	0.573	8.275	8.029
N	48.000		48.000	
chi2				

Making Tables in LaTeX – Concluding Notes

- The key is to be careful in declaring the specifications for your table, and ensuring that all data gets entered properly and in the correct location.

Making Tables in LaTeX – Concluding Notes

- The key is to be careful in declaring the specifications for your table, and ensuring that all data gets entered properly and in the correct location.
- You can automate much of the tedious process by letting your statistical software do the work for you
- Many pages on the web are dedicated to helping you draft your table in LaTeX if you ever get stuck
- Estout produces tables in separate `.tex` files that can be added to your manuscript by specifying:

  ```
  \insert{example}
  ```

- That means: rerunning the estout command on a changed model will automagically update in your manuscript, too.

Introduction to Bibliographies

Bibliographies are tedious work. LaTeX will do much of the work for you.

Introduction to Bibliographies

Bibliographies are tedious work. LaTeX will do much of the work for you.

When using BibTeX you create a separate file containing information on all your references with the extension `.bib` in the same folder as your `.tex` file.

Introduction to Bibliographies

Bibliographies are tedious work. LaTeX will do much of the work for you.

When using BibTeX you create a separate file containing information on all your references with the extension `.bib` in the same folder as your `.tex` file.

Using a reference manager such as JabRef (or Zotero) simplifies and automates this process

The BibTEXfile

The entries in the BibTEXfile will look something like this

The BibTEXfile

The entries in the BibTEXfile will look something like this

```
@BOOK{Cox1993,
  title = {Legislative Leviathan},
  publisher = {University of California Press},
  year = {1993},
  author = {Cox, Gary W. and Matthew D. McCubbins},
  address = {Berkeley}
}
```

The BibTEXfile

The entries in the BibTEXfile will look something like this

```
@BOOK{Cox1993,
  title = {Legislative Leviathan},
  publisher = {University of California Press},
  year = {1993},
  author = {Cox, Gary W. and Matthew D. McCubbins},
  address = {Berkeley}
}
```

Where `Cox1993` is what we call a *Marker* or *Key* that uniquely identifies each entry.

Citing with BibTeX

You will need to tell LaTeX which citation style to use

Citing with BibTeX

You will need to tell LaTeX which citation style to use

Specifically, we recommend using the `natbib` package which provides style files consistent with most of the major citation formats (including APSA). To do this you need to add the following lines to your preamble:

```
\usepackage{natbib}
\bibpunct{(}{)}{;}{a}{}{;}
\bibliographystyle{name of style file}
```

Citing with BibTeX

You will need to tell LaTeX which citation style to use

Specifically, we recommend using the `natbib` package which provides style files consistent with most of the major citation formats (including APSA). To do this you need to add the following lines to your preamble:

```
\usepackage{natbib}
\bibpunct{(}{)}{;}{a}{}{;}
\bibliographystyle{name of style file}
```

Note that the `apsr` style file included in the `harvard` and `natbib` packages is NOT fully consistent with the APSA style manual. Michael Fix and Susanne Schorpp provide a corrected version available from the Society of Political Methodology.

Citing in LaTeX

Back to our example

Citing in LaTeX

Back to our example

```
@BOOK{Cox1993,
  title = {Legislative Leviathan},
  publisher = {University of California Press},
  year = {1993},
  author = {Cox, Gary W. and Matthew D. McCubbins},
  address = {Berkeley}
}
```

Citing in LaTeX

Back to our example

```
@BOOK{Cox1993,
  title = {Legislative Leviathan},
  publisher = {University of California Press},
  year = {1993},
  author = {Cox, Gary W. and Matthew D. McCubbins},
  address = {Berkeley}
}
```

Where `Cox1993` is what we call a *Marker*

Citing in LaTeX

The marker can be used to cite the item within the document as follows:

Citing in LaTeX

The marker can be used to cite the item within the document as follows:

```
\citep{Cox1993}
```

Citing in LaTeX

The marker can be used to cite the item within the document as follows:

```
\citep{Cox1993}
```

. . . which will lead to the following output:
(Cox and McCubbins 1993)

Citing in LaTeX

Example: Typing this:

```
According to \citet{Cox1993}, as well as other work
on the subject \citep{Carmines1987,Kiewiet1988},
political science is a really important subject.
```

Will give you the following output:

According to Cox and McCubbins (1993), as well as other work on the subject (Carmines, McIver, and Stimson 1987; Kiewiet and McCubbins 1988), political science is a really important subject.

Bibliographies with BibTEX

Finally, at the end of the document, where you would like your reference pages to appear, you add the following line:

```
\newpage
\bibliography{filename.bib}
```

Bibliographies with BibTeX

Finally, at the end of the document, where you would like your reference pages to appear, you add the following line:

```
\newpage
\bibliography{filename.bib}
```

It is important to note that you will need to compile your document again after running BibTeX, sometimes 2-3 additional times.

Bibliographies with BibTEX

Carmines, Edward G., John McIver, and James A. Stimson. 1987. "Unrealized Partisanship: A Theory of Dealignment." *Journal of Politics* 49: 376–400.

Cox, Gary W., and Matthew D. McCubbins. 1993. *Legislative Leviathan: Party Government in the House*. Berkeley: University of California Press.

Kiewiet, D. Roderick, and Mathew D. McCubbins. 1988. "Presidential Influence on Congressional Appropriations Decisions." *American Journal of Political Science* 32: 713–736.