Introduction to LaTeX

Raphael Cunha

Program in Statistics and Methodology – PRISM
Department of Political Science
The Ohio State University

cunha.6@osu.edu

October 16, 2013

What is LaTeXand why use it?

- LaTeX(pronounced “lah-tech”) is a document preparation system for high-quality typesetting.
- It allows users to prepare technical and scientific documents of various kinds, such as articles, books, and presentations.
- Designed by academics to accommodate the needs of scientific publishing (equations, tables, figures, etc).
- It makes it easy to produce citations and compile bibliographies, as well as to convert citation formats.
- It creates nice-looking documents.
- It’s free and works across platforms (Windows, Mac OS, Linux).

Getting Started

In order to start using LaTeX, you need to download and install a LaTeXdistribution and an editor.

- Mac users:
  1. Download and install the MacTeX distribution (`http://www.tug.org/mactex/`). MacTeX includes the TeXShop and TeXworks editors. TeXShop is the most popular and user-friendly.

- PC users:
  1. Download and install MiKTeX (`http://www.miktex.org/`).
  2. Download and install an editor, such as TeXstudio, Texmaker, WinEdt, Lyx, among others. The TeXworks editor comes with the MiKTeX distribution, but most users prefer a more user-friendly editor. Try a couple of different ones to find your favorite.

Getting Started: The Basic Setup

1. **Drafting your document**: Draft your document using an editor (see previous slide). All the content and formatting information is saved in a `.tex` file.

2. **Compile your document**: To produce your final document, you compile (or typeset) your TeX file to produce a PDF file with all of the content and formatting.

3. **Produce your bibliography**: All the bibliographical information used in the document must be stored in a `.bib` file. When you compile a document containing bibliographical information, LaTeXcalls the `.bib` file to produce the citations and bibliography in your document.

Basic Document Structure

There are 3 basic parts of a typical LaTeXdocument:

1. The preamble
   - Specifies the type of document (article, presentation, etc).
   - Specifies the packages that need to be loaded.
2. The front matter
   - Title, author, abstract.
3. The body of the document
   - Contents of your document.

The Preamble

- The preamble is the first part of your document.
- This is where you specify what type of document you are producing. To write an article with 12pt font, for example, we use:
  - `\documentclass[12pt]{article}`
- This is also where you load all the packages that you will need to control the look and feel of your document (e.g., margins, line spacing, citation styles, etc):
  - ```
    \usepackage[margin=1in]{geometry}
    \usepackage{setspace}
    \usepackage{harvard}
    ```
    ```

The Front Matter

- The front matter is where you begin to write the content of your document. All the content must be placed between the `\begin{document}` and `\end{document}` commands.
- This is where you can set up a title page with your name, date, title of your article, and an abstract.
  - `\title{}`
    `\author{}`
    `\date{}`
    `\maketitle`
  - The `\maketitle` command will take all the specified information and produce a title page.
- The abstract is written between the following commands: `\begin{abstract}`, `\end{abstract}`.

The Body

- The body of the document is where the main text of your document will appear.
- You can use sections and subsections to divide your text:
  `\section{Section title}`
  `\subsection{Subsection title}`
- If you don't want LaTeXto number your sections, you can use `\section*{Section title}` instead.
- To indent a paragraph, simply skip a line between two blocks of text.
- You can use `\\` to force subsequent text onto the next line.

Text Formatting: Font Face

You can format your text using two types of code: commands or declarations. For example, for setting the font face:

Effect	*Command*	*Declaration*
Roman family	`\textrm{text}`	`{\rmfamily text}`
Sans serif family	`\textsf{text}`	`{\sffamily text}`
Bold series	`\textbf{text}`	`{\bfseries text}`
Italic shape	`\textit{text}`	`{\itshape text}`
SMALL CAPS SHAPE	`\textsc{text}`	`{\scshape text}`

Text Formatting: Font Size

- The general font size in your text is specified in the preamble (see previous slides) by the command:
  `\documentclass[12pt]{article}`

- However, you can also specify the font size for specific parts of the text using the following declarations:

`\tiny`	tiny	`\large`	large
`\scriptsize`	scriptsize	`\Large`	Large
`\footnotesize`	footnotesize	`\LARGE`	LARGE
`\small`	small	`\huge`	huge
`\normalsize`	normalsize	`\Huge`	Huge

- Note that these are declarations and should be used in the form `{\small text}`, or without braces to affect the entire document.

Text Formatting: Spacing

Text spacing can be controlled with:

`\singlespacing, \doublespacing`

or

`\begin{singlespacing}, \end{singlespacing}`

Mathematical Notation

One of the big advantages of LaTeXover other word-processing programs are its capabilities for writing mathematical notation. There are two main ways to write equations or mathematical expressions:

1. **"Math mode"**:
   - Math mode begins and ends by $.
   - For example, to produce any greek letter, you simply need to use math mode and the backslash: `$\beta$` produces β, `$\omega$` produces ω.
   - To capitalize the greek letter, simply capitalize the word, so `$\Omega$` produces Ω.

Mathematical Notation

Full equations work much in the same way in math mode:

- For example, the regression equation
  $y = x_1\beta_1 + x_2\beta_2 + x_2^2\beta_3 + \varepsilon$ is produced with
  ```
  $ y = x_1\beta_1 + x_2\beta_2 + x_2^2\beta_3
  + \varepsilon $.
  ```

- Or, $y_t = \gamma^t + \frac{2-1^{t-1}}{1+1}$ is produced with
  ```
  $ y_t = \gamma^t + \frac{2-1^{t-1}}{1+1} $.
  ```

Mathematical Notation

2. **The equation environment**: Equations can also be written by using `\begin{equation}`, `\end{equation}`.

   The following equation

$$y = x_1\beta_1 + x_2\beta_2 + \varepsilon \tag{1}$$

   is produced with

```
\begin{equation}
y = x_1\beta_1 + x_2\beta_2 + \varepsilon
\end{equation}
```

   You can suppress the equation numbers by using `\begin{equation*}`, `\end{equation*}`.

Figures

- You can insert figures in your document with:

```
\begin{figure}
\centering
   \includegraphics{file_name}
        \caption{} % Title that will be displayed
        \label{} %Internal reference (see below)
\end{figure}
```

- The `\label{}` command in the figure environment allows you to reference that figure throughout the text. E.g., if you use `\label{my_scatter_plot}`, you can reference that figure with the `\ref{}` command, as follows:
  `In Figure \ref{my_scatter_plot}, we can see that....`
- You can either keep the figure in the same folder as the Tex document, or you can specify the file path to where it is saved.

Subfigures

- You might also want to display two or more images side by side in a single figure. The `subfigure` package can be used for this:

```
\begin{figure}
\centering
\caption{Name of Figure}
   \subfigure[Name of first figure]
        {\includegraphics{file_name_1}
        \label{}}
   \subfigure[Name of second figure]
        {\includegraphics{file_name_2}
        \label{}}
\end{figure}
```

Tables

- Tables are constructed in LaTeXusing two environments: `table` and `tabular`.
- The `table` environment lets you set a position, caption and label for your table.
- The `tabular` environment creates the underlying structure of the table, such as the number of rows and columns, cell alignment, etc.
- The number of columns is determined by: `\begin{tabular}{l c r}`. In this example, we are creating a table with 3 columns (indicated by the 3 letters inside the brackets). These columns will be left-, center-, and right-aligned, respectively. If you wanted to add a fourth, right-aligned column, you would use `\begin{tabular}{l c r r}`.
- Cells are separated by &.
- Rows are separated by \\.

Tables

Example:

```
\begin{table}[!p] %Set up table environment
\caption{Summary of Conclusions from Diagnostic Tests} %Caption/title
\begin{tabular}{ll|l|l} %Columns, alignment, vertical lines
\hline %Horizontal line
Test &Variable 1 & Variable 2 & Variable 3\\ %Contents separated by &
\hline
Joint F test & $d=1$ & $d=1$ & $d=0$ \\
VR test & $0<d\leq1$ & $d=1$ & $0<d\leq1$ \\
\hline
\multicolumn{4}{l}{See appendix.}\\ %Merges columns
\end{tabular}
\end{table}
```

Tables

Example:

Table : Summary of Conclusions from Diagnostic Tests

Test	Variable 1	Variable 2	Variable 3
Joint F test	$d = 1$	$d = 1$	$d = 0$
VR test	$0 < d \leq 1$	$d = 1$	$0 < d \leq 1$

See appendix.

Placement of Floats (Tables and Figures)

- The location of figures and tables in the final document roughly corresponds to the location of the figure and table commands in the `.tex` file.
- LaTeXwill try to optimize the use of page space in your document, which means floats will not always appear exactly where you placed them.
- You have, however, some degree of control over their location. The following options are called *placement specifiers*: t, b, h, p (top, bottom, here and (separate) page, respectively).
- These options are used as follows:
  - `\begin{table}[t]`: the table should be placed at the top of a page;
  - `\begin{table}[!hbp]`: the table can be placed right here (h) or at the bottom (b) of some page or on a special floats page (p), and all this even if it does not look that good (!).

Exporting Tables from Stata

- Download and install the `estout` package (`findit estout`) in Stata.
- Here's an example of how to export regression tables in LaTeXformat:

```
eststo: reg y x1 x2 x3
eststo: reg y x1 x2 x3 x4
esttab using filename.tex, se r2 b(3) se(3)
```

- This code estimates two regression models, stores the results and exports them in table format to a Tex file called filename.tex. The `esttab` command specifies that the table should include standard errors (se), R-squared (r2), and report betas and standard errors with 3 decimal points.
- Read the help file (`help estout`) to learn how to further customize your tables.

Exporting Tables from R

- In R, download and use the package `apsrtable` to produce LaTeXcode for tables reporting regression results.
- Check the R help files (`?apsrtable`) to learn how to produce and customize your regression tables.

Miscellanea

- Bullet points take place within `\begin{itemize}` `\end{itemize}`, with each bullet point denoted by `\item`.
- Enumerated items work just the same with `\begin{enumerate}` `\end{enumerate}`.
- Block quotes can be set with `\begin{quote}` `\end{quote}`.
- Many symbols you would typically use in a paper actually double as commands within LaTeX. To avoid this, use the backspace before the symbol (e.g., \% to produce %, or \$ to produce $).
- A typical mistake involves quotation marks. To produce “, you need to type ‘‘ rather than “, which will result in ”.
- To insert footnotes, use `\footnote{}` and write the text of the footnote within the brackets.
- `\pagebreak` tells LaTeXto break the current page at the point of the command.

BibTeX and Citations

- BibTeX is the standard LaTeXbibliography format.
- The `.bib` file is where you store all of your references. It is recommended that you have one master `.bib` file where all of your references are stored, and BibTeX will simply use the ones it needs each time you prepare a document.
- The `.bib` file is a flat text file that has a series of lines for each entry. This is what an entry in a `.bib` file looks like:

```
@article{GrossmanHelpman94,
Author = {Gene M. Grossman and Elhanan E. Helpman},
Journal = {American Economic Review},
Number = {4},
Pages = {833-850},
Title = {Protection for Sale},
Volume = {84},
Year = {1994},}
```

BibTeX and Citations

- Of course, manually entering information in a `.bib` file is tedious, time-consuming, and error-prone.
- Therefore, it is recommended that you use a front-end program, such as BidDesk (which comes with the MacTeX distribution) or JabRef (Windows).
- Also, notice that some reference-management programs, such as Zotero, Mendeley, and EndNote, will export your library to BibTeX format, which can save you a lot of time and effort.

Using BibTeX

- Load the bibliography style and citation packages in the preamble, for example:
  `\usepackage{cite}`
  `\usepackage{harvard}`
- Specify the citation style you want to use in the body of the text, for example:
  `\bibliographystyle{apsr}`
- To create a list of references at the end of your document, call the `.bib` file by inserting the following command in the location where you want your references section to appear:
  `\bibliography{My_bib_file}`
- Make sure the `.bib` file and the `.tex` file (and also the `.sty` file, if you're using a special style) are in the same folder.

Citation formats

- When you use `\bibliography{}`, BibTeX searches your document for citations and uses them to compile your bibliography.
- Citation technique varies with the citation package you use. Here I will give examples for the `cite` package, but there are others (e.g., `natbib`):

`\cite{key1}`: In text author-year citation.
e.g. (Grossman & Helpman 1994)

`\cite{key1,key2}`: In text author-year citation for two works.
e.g. (Grossman & Helpman 1994, Jones 2012)

`\citeyear{key1}`: In text year only citation.
e.g. (1994)

`\cite[#]{key1}`: In text author-year citation with page number #.
e.g. (Grossman & Helpman 1994, 842)

Typesetting with .bib files

- *Important*: To get the references and citations to appear in your final document, you need to typeset your document four times in this specific order:

  1. LaTeX
  2. BibTeX
  3. LaTeX
  4. LaTeX

Getting Help

- The Not So Short Introduction to LaTeX
- LaTeXCheat Sheet
- LaTeXMathematical Symbols
- The LaTeXQ&A forum at Stack Exchange
- Google is your friend
- PRISM Fellows!