Introduction to LaTeX

Dave Armstrong
University of Wisconsin - Milwaukee
Department of Political Science

e: armstrod@uwm.edu
w: http://www.quantoid.net/teachicpsr/latex (Course Materials)

Contents

List of Tables

List of Figures

1 Introduction and What You Need

1.1 Distribution of LaTeX

- For Windows, you could use:
  - MiKTeX or proTeXt, which is a MiKTeX-based distribution that guides you through the installation process.
  - TeXLive
- For Mac, you could use:
  - MacTeX, which is a TeXLive version for Mac.
  - XeTeX, which has additional font support.
- For Linux, you could use:
  - TeXLive
- In general, you can look here for advice on getting started with LaTeX.

1.2 Text Editor

While you could use any text editor (including Notepad) for LaTeX, there are many options that come with built-in and easily customizable LaTeX functionality. The "Packages and programs" section of this link gives a number of free options. I also have had good luck with WinEdt on Windows, $40 educational, $30 student) and TextMate, ∼$50, educational for version 1, version 2 is open source). Many, though certainly not all, of the free LaTeX editors provide environments specific to LaTeX. However, WinEdt, Textmate, TextWrangler/BBEdit, Emacs and others provide support for many environments (including R, Stata, Sweave, etc...). I would suggest TeXStudio for the beginners because it offers good syntax highlighting and some nice interactivity with citations. It is also quite customizable.

Ultimately, all of these editors are simply GUIs that call command-line tools (e.g., `pdflatex`, `bibtex`, etc...) by pushing a button. Often, you can configure what happens exactly when different buttons are pushed.

1.3 Ancillary Stuff

Windows users would do well to install Ghostscript and Ghostview which are programs that facilitate working with postscript documents. These can be downloaded from here:

- Ghostscript
- Ghostview

Mac users will likely already have facility to work with these files.

You'll also need a PDF viewer, like Acrobat, Foxit, Skim or Prview. Though you probably already have one of these. It is worth installing LaTeX, the postscript software and the pdf viewer *before* installing your text editor. Most of them will query the file system to find where all of the necessary software is installed.

2 The LaTeX Document

The LaTeX document has two main parts - the Preamble and the Body. The Preamble sets up the document by including any packages that you *might* use, though you don't have to use all of the packages that you load. It also allows you to define new commands and set properties of the document (like author, title, margins, header style, etc...).

```
\documentclass[12pt]{article}
\usepackage{amsfonts, amsmath, amssymb, bm} %Math fonts and symbols
\usepackage{dcolumn, multirow} % decimal-aligned columns, multi-row cells
\usepackage[colorlinks="red"]{hyperref}
\usepackage{graphicx, subfigure, float} % graphics commands
\usepackage[margin=1in]{geometry} % sets page layout
\usepackage{setspace}% allows toggling of double/single-spacing
\usepackage{verbatim}% defines environment for un-evaluated code
\usepackage{natbib}% defines citation commands and environments.
\singlespace % set document spacing to single
\bibpunct[, ]{(}{)}{,}{a}{}{,} % sets the punctuation of the bibliography entires.
\newcolumntype{d}[1]{D{.}{.}{#1}} % defines a decimal-aligned column

\title{Introduction to \LaTeX{}} % set the title of the document
\author{Dave Armstrong \\ University of Wisconsin - Milwaukee \\
    Department of Political Science}
% set the author information use \and to include other authors
```

The Preamble is everything the between `\documentclass[]{}` and `\begin{document}`.

The Body of the document is where all of the text and code that will be compiled by LaTeX goes. With few exceptions, the things you write in the preamble won't show up in the document (except for the title commands). As you will see, LaTeX is a language that treats text differently depending on the *environment*. Environments are basically containers for text that define different ways of dealing with that text. In LaTeX, different environments can be invoked with `\begin{environment-name}` and can be stopped with `\end{environment-name}`. These environments can be nested to produce specific output. In fact, all of the text we're typing here is in the *document* environment, that we invoked with `\begin{document}` and will close after all of our writing is done with `\end{document}`.

3 Typing in LaTeX

Before we go extensively into some of the other material, there are a few hints that will reduce *some* of the initial frustration of working with LaTeX. First, if you're making pdf or picture (jpg, bmp, png) figures, you'll want to typeset your document with `pdflatex` (this should be an option in whatever editor you're using). This typesets your document as a pdf. The other option, using `latex`, will produce a dvi file that will have to be converted either to ps or pdf (through ps). This option also requires ps or eps figures as input. When I say "LaTeXthe document," generally I mean run `pdflatex`.

- `\\` and `\newline` - both move whatever follows them to a new line. However, if there is "no line to end", then you will get complaints about this.

- `\newpage` will put everything below starting on a new page.

- `` `` `` is the open double-quote symbol and `''` is the closed double-quote symbol.

- `\section`, `\subsection` and `\subsubsection` - all generate sequentially nested numbered sections.

  ```
  \section{Section 1 }
  \subsection{Subsection 1}
  \subsubsection{Subsubsection 1}
  ```

  will produce:

  # 1 Section 1

  ## 1.1 Subsection 1

  ### 1.1.1 Subsubsection 1

- `\footnote{}` puts a footnote with the text between the curly braces at the bottom of the page. You can use the package `endnotes.sty` to convert from footnotes to endnotes. See this section of The LaTeX Companion for more details on footnotes and endnotes.

3.1 Common and Useful Environments

There are a number of LaTeX environments that you will likely find yourself using quite frequently.

- `verbatim` is an environment that allows you to print commands that would otherwise be interpreted. This is useful for including statistical output in your document. This has its own environment for long segments of `verbatim` text: `\begin{verbatim}` and `\end{verbatim}`. It also has a command for `verbatim` text that fits on a single line:

```
\verb""
```

some text `mod <- lm(y ~ x, data=dat)` some more text

```
mod <- lm(y ~ x, data=dat)
```

- Enumerate, Itemize and Description - all are environments that give sequential bullet-like points. `enumerate` uses numbers and letters, `itemize` uses different character symbols, and `description` uses words to offset different items. See the ShareLaTeX Reference Guide for help on changing the enumeration.

3.1.1 itemize

```
\begin{itemize}

\item First bullet point.
\end{itemize}
```

- First bullet point.

3.1.2 enumerate

```
\begin{enumerate}

\item First bullet point.
\end{enumerate}
```

1. First bullet point.

You can also next enumerate environments, which would produce the following:

```
\begin{enumerate}
\item First item
\begin{enumerate}
\item First sub-item
\begin{enumerate}
\item First sub-sub-item
\begin{enumerate}
\item First sub-sub-sub-item.
\end{enumerate}
\end{enumerate}
\end{enumerate}
\end{enumerate}
```

1. First item

   (a) First sub-item

      i. First sub-sub-item

         A. First sub-sub-sub-item.

3.1.3 description

```
\begin{description}

\item[First bullet] point.
\end{description}
```

First bullet point.

You can also change the itemize symbol by putting the symbol you want it `[ ]` after the `\item` statement.

○ `\item[$\circ$]`

• `\item[$\bullet$]`

⋆ `\item[$\star$]`

⧫ `\item[$\blacklozenge$]`

■ `\item[$\blacksquare$] \item[\tiny{$\blacktriangleright$} ]`

3.2 Text Size and Spacing

Generally, it is not as easy to change font point sizes in LaTeXas it is in Word or other word processing programs. However, there are a number of font sizes from which you can choose.

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

These can also define environments (as in `\begin{tiny}` and `\end{tiny}`, with everything in between those two statements being printed in tiny text. There are other document classes that permit different font sizes. The `article` and `report` document classes permit only point sizes 10, 11 and 12. For example,

```
\documentclass[10pt]{article}
```

will make the default point size 10. The `extarticle` and `extreport` classes allow 8, 9, 10, 11, 12, 14, 17 and 20 point fonts. There is a nice article here on font sizes in LaTeX and here on redefining point sizes for the font size categories mentioned above.

With the `setspace` package, you can use either `\singlespace` or `\doublespace` in your preamble to set the default for the document and then you can use either tag in the document to set spacing for a certain section. In general, if you set the default to `\doublespace` and use the `\singlespace` tag in the document, text will be single-spaced until the next `\doublespace` command is used.

```
\doublespace
```

Now we are engaged in a great civil war, testing whether that nation, or any nation so conceived and so dedicated, can long endure. We are met on a great battle-field of that war. We have come to dedicate a portion of that field, as a final resting place for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this.

```
\singlespace
```

Now we are engaged in a great civil war, testing whether that nation, or any nation so conceived and so dedicated, can long endure. We are met on a great battle-field of that war. We have come to dedicate a portion of that field, as a final resting place for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this.

4 Tables and Tabular Material

Tables and tabular material also go in their own environments. The tabular environment allows you to add tabular material. As you will see in the examples below, the ampersand (`&`) separates the contents of each cell and every line has to be ended with `\\`. You will also notice in the tabular environment that you have to declare a justification for each column. For example, let's say we wanted a table with three columns, with the first one left-justified and the other two aligned on the decimal point, you would invoke the tabular environment as follows:

```
\begin{tabular}{ld{2}d{2}}
```

A couple of things to note here. First, to use the decimal-aligned columns, you should put the following lines in your preamble:

```
\usepackage{dcolumn}
\newcolumntype{d}[1]{D{.}{.}{#1}}
```

the first line loads the dcolumn package and the second creates a new column type, so rather than specifying `D{.}{.}{2}` for a decimal-aligned column with two decimal places, you can simply put `d{2}`. Another thing to note is that you don't necessarily *need* two decimal places in each entry, this just tells LaTeXto leave enough room for two decimal places.

The table environment usually envelopes the tabular environment. The table environment adds a caption to tables and allows them to be referenced elsewhere in the text with labels. More on this later.

4.1 Single-tables in the `table` environment

```
\begin{table}[H]
\caption{Percentage of ICPSR students using various statistical
software packages by session}\label{tab:pack1}
\centering
\begin{tabular}{ld{2}d{2}d{2}d{2}}
\hline
 & \multicolumn{4}{c}{Packages}\\
\hline
 & \multicolumn{1}{c}{R} & \multicolumn{1}{c}{SPSS} &
   \multicolumn{1}{c}{STATA} & \multicolumn{1}{c}{SAS} \\
\hline\hline

Session 1 & 40.25 & 30.15 & 20.1 & 10.5 \\
Session 2 & 20.5 & 30.1 & 40.25 & 10.15 \\
\hline\hline
\end{tabular}
\end{table}
```

Table 1: Percentage of ICPSR students using various statistical software packages by session

	Packages			
	R	SPSS	STATA	SAS
Session 1	40.25	30.15	20.1	10.5
Session 2	20.5	30.1	40.25	10.15

```
\begin{table}[H]
\caption{Percentage of ICPSR students using various statistical
software packages by session}\label{tab:pack2}
\centering
\begin{tabular}{l|d{2}d{2}d{2}d{2}}
\hline
          & \multicolumn{4}{c}{Packages}\\
\hline
          & \multicolumn{1}{c}{R} & \multicolumn{1}{c}{SPSS} &
\multicolumn{1}{c}{STATA} & \multicolumn{1}{c}{SAS} \\
\hline\hline

Session 1 & 40.25 & 30.15 & 20.1 & 10.5 \\
Session 2 & 20.5 & 30.1 & 40.25 & 10.15 \\
\hline\hline
\end{tabular}
\end{table}
```

Table 2: Percentage of ICPSR students using various statistical software packages by session

	Packages			
	R	SPSS	STATA	SAS
Session 1	40.25	30.15	20.1	10.5
Session 2	20.5	30.1	40.25	10.15

Typing `Table~\ref{tab:pack1}` will reference Table 1, and `Table~\ref{tab:pack2}` will reference Table 2.

4.2 Using the `subtable` command for multiple tables

```
\begin{table}[h!]
\caption{ICPSR Participants in 2001 and 2006}\label{tab:subtab}

\centerline{\hbox{ \subtable[2001]{
\begin{tabular}{lcc}
  & Men & Women\\
\hline
Session 1 & 200 & 150 \\
Session 2 & 150 & 125 \\
\hline
\end{tabular}\label{tab:subtab1}}
\quad \quad \subtable[2006]{
\begin{tabular}{lcc}
  & Men & Women\\
\hline
Session 1 & 250 & 200 \\
Session 2 & 175 & 150 \\
\hline
\end{tabular}\label{tab:subtab1}}}}

\end{table}
```

Table 3: ICPSR Participants in 2001 and 2006

(a) 2001

	Men	Women
Session 1	200	150
Session 2	150	125

(b) 2006

	Men	Women
Session 1	250	200
Session 2	175	150

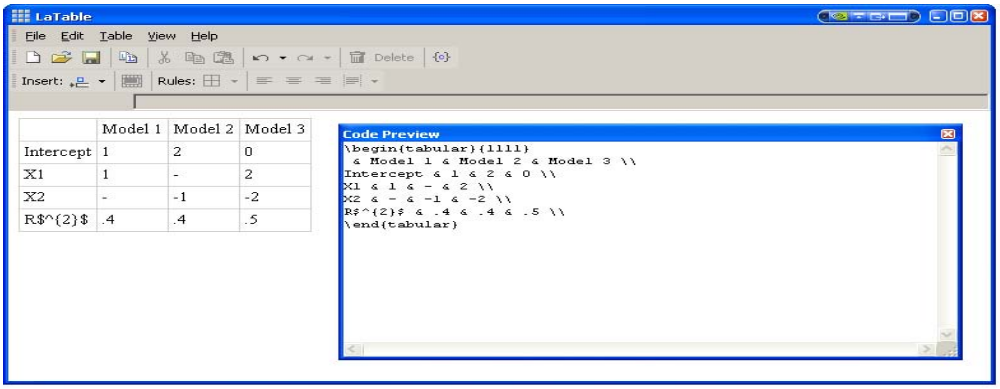
To reference a subtable, simply put the `\label{}` inside the `\subtable{}` command, as above. Then, typing `Table~\ref{tab:subtab1}` will produce Table 3(a). Using `\subref{}` instead of `\ref{}` will print only the letter, rather than the letter and the number.

For more advice on and example code for making tables, see this article from the PracTeX Journal and this article from CTAN.

4.3 Making Tables More Easily

4.3.1 LaTable

LaTable is a stand-alone package that generates table code for LaTeX. Tables are not hard to make, but sometimes can get cumbersome. LaTable is a WYSIWIG-like table editor that produces LaTeX code.



	Model 1	Model 2	Model 3
Intercept	1	2	0
X1	1	-	2
X2	-	-1	-2
R^{2}	.4	.4	.5

```
\begin{tabular}{llll}
 & Model 1 & Model 2 & Model 3 \\
Intercept & 1 & 2 & 0 \\
X1 & 1 & - & 2 \\
X2 & - & -1 & -2 \\
R$^{2}$ & .4 & .4 & .5 \\
\end{tabular}
```

Table 4: Table from LaTable

	Model 1	Model 2	Model 3
Intercept	1	2	0
X1	1	-	2
X2	-	-1	-2
R^2	.4	.4	.5

Mac and Linux users, don't be sad, really - there is a website that does something very similar http://truben.no/table/.

4.3.2 Models from R into LaTeX

In R, you could do the following:

```
> library(car)
> library(apsrtable)
> mod1 <- lm(prestige ~ income + type, data = Duncan)
> mod2 <- lm(prestige ~ income + education, data = Duncan)
> mod3 <- lm(prestige ~ income + education + type, data = Duncan)
> NA

> apsrtable(mod1, mod2, mod3)
> NA
```

Table 5: Example tables from R

	Model 1	Model 2	Model 3
(Intercept)	6.70*	−6.06	−0.19
	(3.22)	(4.27)	(3.71)
income	0.68*	0.60*	0.60*
	(0.09)	(0.12)	(0.09)
typeprof	33.16*		16.66*
	(4.83)		(6.99)
typewc	−4.28		−14.66*
	(5.55)		(6.11)
education		0.55*	0.35*
		(0.10)	(0.11)
N	45	45	45
R^2	0.89	0.83	0.91
adj. R^2	0.89	0.82	0.90
Resid. sd	10.68	13.37	9.74

Standard errors in parentheses

* indicates significance at $p < 0.05$

You can just cut-and-paste the output from R into LaTeX. There are other options for tables of objects (not necessarily model output) such as `xtable` in the package of the same name and `mtable` in the `memisc` package.

4.3.3 Models from Stata into LaTeX

In Stata, do the following:

```
use "/Users/armstrod/Desktop/Duncan.dta"
xi: reg prestige income i.type
est store mod1
reg prestige income education
est store mod2
xi: reg prestige income i.type education
est store mod3
```

Then, issue the following command:

```
estout mod1 mod2 mod3 using "~/Desktop/latexumd/statamods.tex",
    style(tex)
    cells(b(star fmt(3)) t(par fmt(2)))
    starlevels(* .05)
    replace label collabels(, none)
    varlabels(_cons Constant income Income _Itype_2 Professional
        _Itype_3 "White Collar" education Education)
    posthead("") prefoot("") postfoot("")
    varwidth(16) mlabels("" "" "")
```

Table 6: Example tables from Stata

	Model 1	Model 2	Model 3
Income	0.676∗	0.599∗	0.598∗
	(7.21)	(5.00)	(6.69)
Professional	33.156∗		16.658∗
	(6.86)		(2.38)
White Collar	−4.277		−14.661∗
	(−0.77)		(−2.40)
Education		0.546∗	0.345∗
		(5.56)	(3.04)
Constant	6.704∗	−6.065	−0.185
	(2.08)	(−1.42)	(−0.05)

This will put your table in the file `statamods.tex` and will print it to the screen (from which you could just copy and paste into your LaTeX file), or you could do the following:

```
\begin{table}
\caption{Example tables from Stata}\label{tab:Statatab}
\centering
\begin{tabular}{ld{3}d{3}d{3}}
 & \multicolumn{1}{c}{Model 1} & \multicolumn{1}{c}{Model 2} &
   \multicolumn{1}{c}{Model 3}\\
\hline
\input{statamods.tex}
\hline
\end{tabular}
\end{table}
```

5 Figures and Graphics

The `graphicx` package allows for the inclusion of many types of graphics including `.ps, .eps, .pdf, .bmp, .jpg, .wmf`, and doubtless others as well through the `\includegraphics` command.

5.1 The Figure Syntax

- The `figure` environment allows graphics to be included, titled with a caption and automatically numbered. Graphics can be included without putting them in the `figure` environment, but cannot then be titled with automatically numbered captions.
- The command for including a graphic is `\includegraphics[options]{filename}` (note the `s`, not `x`).

- Some useful arguments or options for `\includegraphics` are:

  width, height set the width and height to the values (e.g., [height=2in, width=1in]). See here for the various different ways units can be specified

  keepaspectratio preserves the aspect ratio of the original figure

  angle angle of figure rotation (degrees counterclockwise).

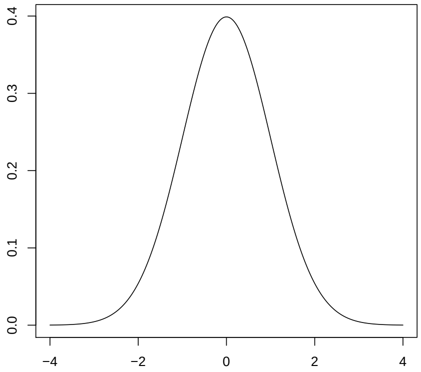
- It is easiest to put graphs in the directory where your document is. That is where LaTeX is going to look for them by default. However, there is a command you can put in the preamble: `\graphicspath{{dir1}{dir2}{dir3}...}`. However,

  1. `dir1` must be a Unix-style path `C:/graphs/` not `C:\graphs\`
  2. LaTeX doesn't like spaces in the directory names, so the directories may not have spaces in them.

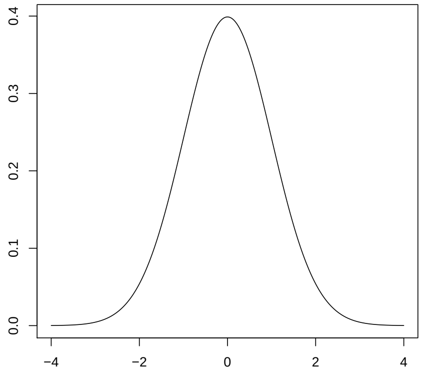
5.2 Single Graphic Figures

```
\begin{figure}[h!]
\caption{Normal Density over [-4,4], $\mu$=0,  $\sigma$=1 with more
words to make the caption really long}\label{fig:onefig}
\centerline{\includegraphics[width=2.5in]{norm1.pdf}}
\end{figure}
```

Figure 1: Normal Density over [-4,4], μ=0, σ=1 with more words to make the caption really long



Now, without the `figure` environment:



I can reference Figure 1 with `Figure~\ref{fig:onefig}`.

5.3 Including Multiple Figures with `subfigure`

```
\begin{figure}[h!]

\caption{Three Normal Densities over [-4,4], $\mu$=0, $\sigma$= a)1,
b)5, c)10}


\centerline{\subfigure[$\sigma$=1]{\includegraphics[width=.3\textwidth]{norm1.pdf}}
\quad
\subfigure[$\sigma$=5]{\includegraphics[width=.3\textwidth]{norm2.pdf}}}

\centerline{\subfigure[$\sigma$=10]{\includegraphics[width=.3\textwidth]{norm3.pdf}}}
\end{figure}
```

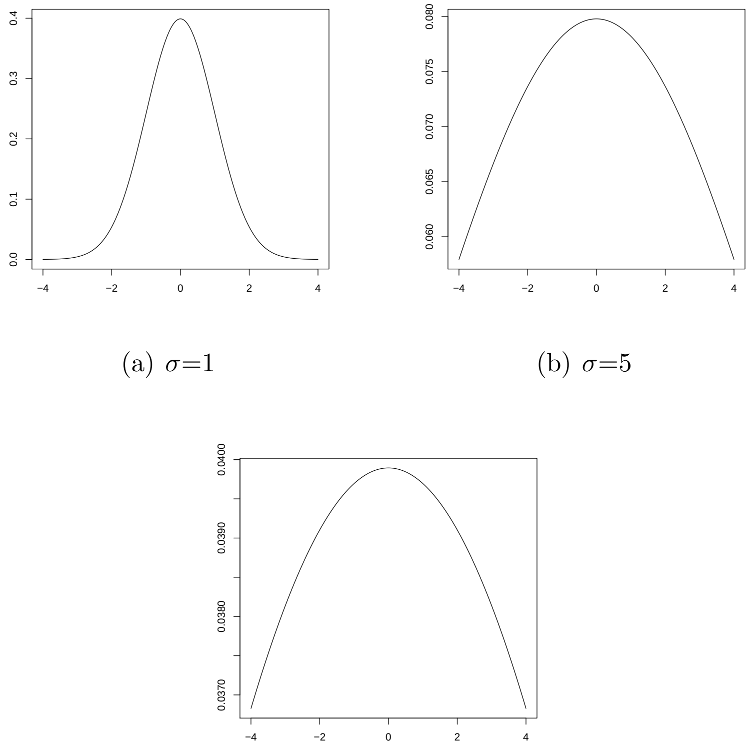
See Mittelbach and Goosens (2004, Chapter 6) for more information on controlling and customizing the appearance of floats.

6 Equations

To me, the big advantage of LaTeX is that you can type equations, rather than hunting and pecking through some pull-down menus. You'll notice that once you've become familiar with the symbols you need (and there are relatively few of them), it is much faster to type than peck through menus. If you want a list of common symbols and how to make them, look at this List of LaTeX Math Symbols

I have prepared some example equations below, so you can see how things work. There are two environments that may be useful here. The equation environment allows you to type single equations and the align environment allows you to type multi-line equations where you can control the center-point of each line. Let's start with something relatively simple:

Figure 2: Three Normal Densities over [-4,4], μ=0, σ= a)1, b)5, c)10



(a) σ=1

(b) σ=5

(c) σ=10

```
\begin{equation}
y_{i} = b_{0} + b_{1}x_{i} + \varepsilon_{i}\label{eq:linreg}
\end{equation}
```

$$y_i = b_0 + b_1 x_i + \varepsilon_i \tag{1}$$

Typing `equation~\ref{eq:linreg}` will refer to equation 1

Using the `equation*` environment will suppress the equation number

```
\begin{equation*}
\hat{\beta} =
\left(\mathbf{X^{\prime}X}\right)^{-1}\mathbf{X^{\prime}Y}
\end{equation*}
```

$$\hat{\beta} = \left(\mathbf{X^{\prime}X}\right)^{-1}\mathbf{X^{\prime}Y},$$

but so would doing the following:

```
\begin{equation}
\hat{\beta} =
\left(\mathbf{X^{\prime}X}\right)^{-1}\mathbf{X^{\prime}Y}\nonumber
\end{equation}
```

$$\hat{\beta} = \left(\mathbf{X'X}\right)^{-1}\mathbf{X'Y},$$

To ensure that parentheses and other delimiters are big enough to accommodate fractions and sums, you can use the `\left(` and `\right)` which will make the parentheses big enough to completely cover the expression inside them.

```
\begin{equation}
\bar{X} = \left(\frac{1}{n}\right)\sum X
\end{equation}
```

$$\bar{X} = \left(\frac{1}{n}\right)\sum X \tag{2}$$

In general, `\left\{`, `\left(`, and `\left[`, will produce curly braces, parentheses, and square brackets that are the appropriate size for the expression. They are offset by their `\right\}`, `\right)` and `\right]` counterparts.

The align environment lines multi-line equations up wherever you put the ampersand on each line. For example,

```
\begin{align}
x + 11 &= 3\\
x &= 3-11\\
x &=-8
\end{align}
```

will produce

$$\begin{align} x + 11 &= 3 \tag{3}\\ x &= 3-11 \tag{4}\\ x &= -8 \tag{5}\end{align}$$

Note that to align the equations on the equal sign, you can simply place the ampersand before the equal sign in each equation. Also note that you need the `\\` and the end of each line. Using `\nonumber` (before the `\\`) will shut the equation number off for that particular line, but will leave other equation numbers intact.

You can include math symbols in in-line text by using `$ $` as follows: The matrix algebra generating regression coefficients is $\left(\mathbf{X'X}\right)^{-1}\mathbf{X'Y}$, which was obtained as follows: `$\left(\bm{X}^{\prime}\bm{X}\right)^{-1}\bm{X}^{\prime}\bm{Y}$`.

7 BibTeX and Citations

To use BibTeX, you will need to make an appropriate bibliography file and then tell LaTeX how you want those citations to show up in the document. I use the natbib package, but others (like harvard) exist as well. In fact, the file necessary for making APSR-style bibliographies is in the harvard package, so having the harvard package around is not such a bad idea.

The `.bib` file is where you store all of your references. I would suggest that you have one master `.bib` file where all of your references are stored and BibTeX will simply take out the ones it needs each time. The particulars for doing this are different depending on your operating system.

- On the mac, you can simply place your bibliography file in the following folder: `~/Library/texmf/bibtex/bib/` and or `usr/local/texlive/2012/texmf-dist/bibtex/bib|` regardless of where your document is, it will look in this folder for your bibliography file. Otherwise, it will look in the directory where your document lives.

- On the PC, using MikTeX, you can put your bibliography file in:
  `C:\Program Files\Miktex 2.9\bibtex\bib\`,
  `C:\Program Files\texlive\2012\texmf-dist\bibtex\bib` (or use the appropriate path to where MikTeX lives). Otherwise (generally if you don't have write permission to that drive), use MikTeX Guide to Local Additions to guide you through the process of setting up a local tex directory structure and putting your bibliography file there.

- On a Linux machine, I suspect, though don't know for sure, that the solution is quite similar to the mac.

This `.bib` file is a flat text file that has a series of lines for each entry of the following type.[1] I just compiled my `.bib` file by typing my references in a file in this format. However, there are some BibTeX front-ends that make it easier to manage bibliographies. One of the most popular is a java-based program called Jabref. There are other options, like Papers (if you don't mind spending $80 for a polished piece of software) or Bibdesk if you're on the Mac. If you're just getting started, this a good way to begin.

If you're using `natbib`, the documentation is quite useful, but here are a few example citations:

- `\citet{MG2004}` produces Mittelbach and Goosens (2004)

- `\citep{MG2004}` produces (Mittelbach and Goosens 2004)

- `\citep[See][251]{MG2004}` produces
  (See Mittelbach and Goosens 2004, 251)

- `\citeyearp{MG2004}` produces (2004)

[1] See Mittelbach and Goosens (2004, Chap. 13) for more details on Bibliography generation.

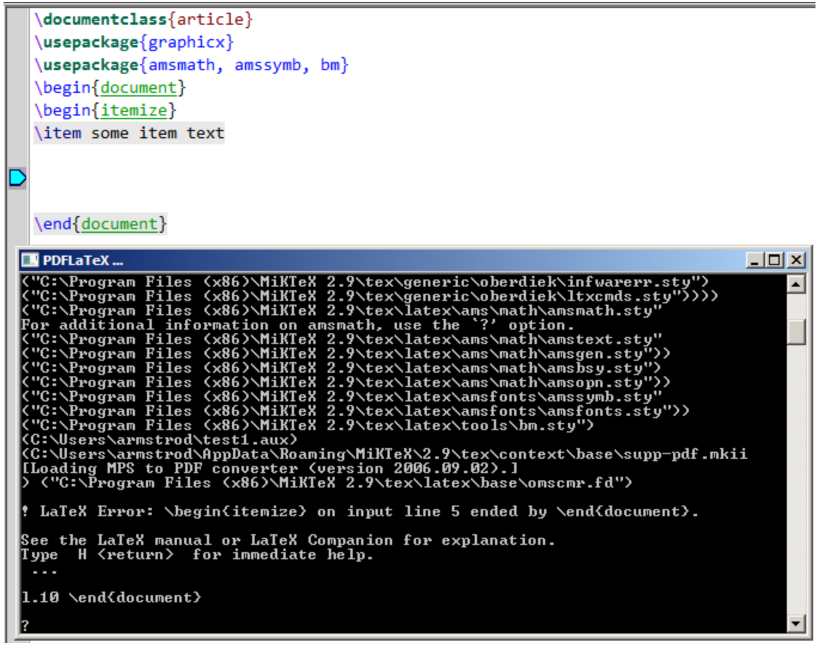
Remember the sequence: LaTeX, BibTeX, LaTeX, LaTeX, you have to do this to get the references to appear. In fact, there are lots of times you have to typeset the document more than once because it takes a couple of times for references to resolve and get printed back into the document. So, if something doesn't show up the first time, just try typesetting the document again. I won't have time to go into this, but there is a utility called `latexmk` which is a highly-configurable piece of software that automatically typesets your document, including running BibTeX (and other ancillary programs) enough times to resolve all cross-references. If you're using TexStudio or TextMate (and presumably some others, too), you can specify that you want the entire chain of commands to be run that will resolve all (cross)references. In TextMate, go into the Preferences panel in the LaTeX bundle and check the "use latexmk.pl" check box. In TeXStudio, go to Preferences (on the Mac) or Options→Configure TeXStudio (on Windows). Click the check box for "Show Advanced Options". Then, click on the "Build" link in the left-hand panel. By "Build & View", click the down arrow in the text box and choose `|txs:///pdf-chain`. Then, when you click the "Build & View" button, it will do everything needed to typeset the pdf.
Then, just type `\bibliography{bibname}` where `bibname` is the name of your bibliography (.bib) file without the .bib extension. For example, I used `\bibliography{latex}` for this document because I stored the references above in a file called `latex.bib`.

8 Troubleshooting

When you are typesetting a document, generally some sort of console window will appear that will show you the progress of the typesetting endeavor. When you encounter an error, the WinEDT console (though this differs across software) will ask for user input with "`?`". If you type `r` at the question mark, then LaTeXwill try to recover from the error and produce something. It may produce something that looks fine and it may not. If you type `q` at the question mark, LaTeXwill quit and allow you to fix the error.

One of the most common ways to generate an error is to not close a delimiter (that LaTeXis expecting to be closed). This doesn't so much matter for parentheses or other delimiters that you might use in prose, but matters a lot for things like environments and delimiters within mathematical expressions (though not always). Another common mistake is excluding a file that LaTeXthought it should find. A few examples should clarify:

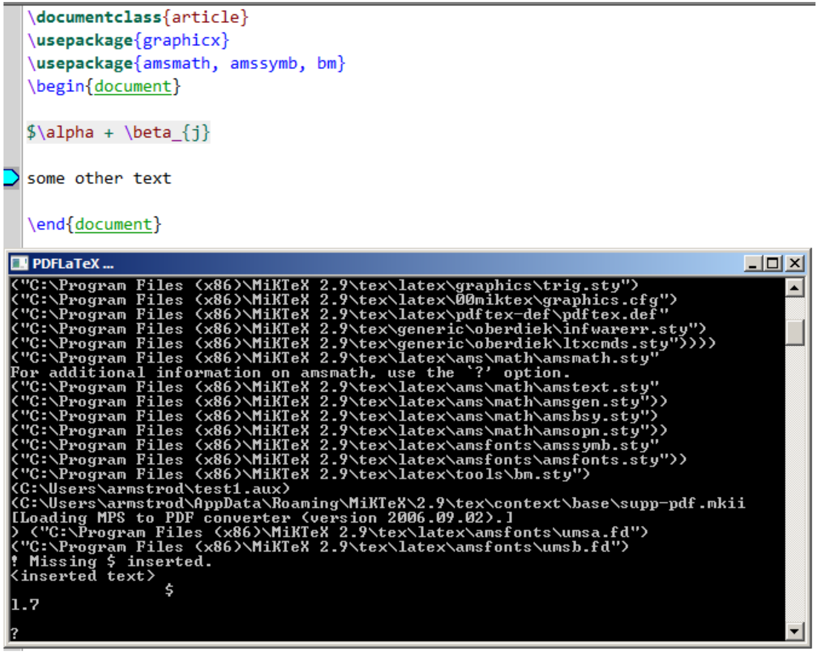
- Here, I forgot to close the `\begin{itemize}` with `\end{itemize}`. Notice that the error is actually quite informative, it suggests where the error occurs (`l.10` means that it encountered the error on line 10 of the document), but the error indicates that the problem started with the tag on line 5 not being closed.



```
\documentclass{article}
\usepackage{graphicx}
\usepackage{amsmath, amssymb, bm}
\begin{document}
\begin{itemize}
\item some item text



\end{document}
```

```
("C:\Program Files (x86)\MiKTeX 2.9\tex\generic\oberdiek\infwarerr.sty")
("C:\Program Files (x86)\MiKTeX 2.9\tex\generic\oberdiek\ltxcmds.sty"))))
("C:\Program Files (x86)\MiKTeX 2.9\tex\latex\ams\math\amsmath.sty"
For additional information on amsmath, use the `?' option.
("C:\Program Files (x86)\MiKTeX 2.9\tex\latex\ams\math\amstext.sty"
("C:\Program Files (x86)\MiKTeX 2.9\tex\latex\ams\math\amsgen.sty"))
("C:\Program Files (x86)\MiKTeX 2.9\tex\latex\ams\math\amsbsy.sty")
("C:\Program Files (x86)\MiKTeX 2.9\tex\latex\ams\math\amsopn.sty"))
("C:\Program Files (x86)\MiKTeX 2.9\tex\latex\amsfonts\amssymb.sty"
("C:\Program Files (x86)\MiKTeX 2.9\tex\latex\amsfonts\amsfonts.sty"))
("C:\Program Files (x86)\MiKTeX 2.9\tex\latex\tools\bm.sty")
(C:\Users\armstrod\test1.aux)
(C:\Users\armstrod\AppData\Roaming\MiKTeX\2.9\tex\context\base\supp-pdf.mkii
[Loading MPS to PDF converter (version 2006.09.02).]
) ("C:\Program Files (x86)\MiKTeX 2.9\tex\latex\base\omscmr.fd")

! LaTeX Error: \begin{itemize} on input line 5 ended by \end{document}.

See the LaTeX manual or LaTeX Companion for explanation.
Type  H <return>  for immediate help.
 ...

l.10 \end{document}

?
```

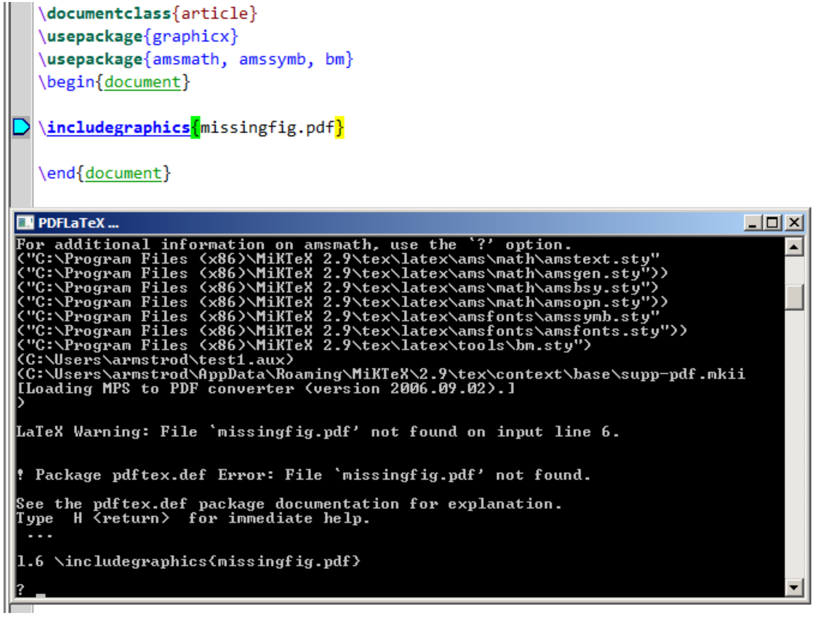
- Here, I forgot to close a math delimiter. I forgot to put a closing dollar sign. Notice that the error suggests the problem and identifies that LaTeXwill try to solve it by inserting the dollar sign.



```
\documentclass{article}
\usepackage{graphicx}
\usepackage{amsmath, amssymb, bm}
\begin{document}

$\alpha + \beta_{j}

some other text

\end{document}
```

```
("C:\Program Files (x86)\MiKTeX 2.9\tex\latex\graphics\trig.sty")
("C:\Program Files (x86)\MiKTeX 2.9\tex\latex\00miktex\graphics.cfg")
("C:\Program Files (x86)\MiKTeX 2.9\tex\latex\pdftex-def\pdftex.def"
("C:\Program Files (x86)\MiKTeX 2.9\tex\generic\oberdiek\infwarerr.sty")
("C:\Program Files (x86)\MiKTeX 2.9\tex\generic\oberdiek\ltxcmds.sty"))))
("C:\Program Files (x86)\MiKTeX 2.9\tex\latex\ams\math\amsmath.sty"
For additional information on amsmath, use the `?' option.
("C:\Program Files (x86)\MiKTeX 2.9\tex\latex\ams\math\amstext.sty"
("C:\Program Files (x86)\MiKTeX 2.9\tex\latex\ams\math\amsgen.sty"))
("C:\Program Files (x86)\MiKTeX 2.9\tex\latex\ams\math\amsbsy.sty")
("C:\Program Files (x86)\MiKTeX 2.9\tex\latex\ams\math\amsopn.sty"))
("C:\Program Files (x86)\MiKTeX 2.9\tex\latex\amsfonts\amssymb.sty"
("C:\Program Files (x86)\MiKTeX 2.9\tex\latex\amsfonts\amsfonts.sty"))
("C:\Program Files (x86)\MiKTeX 2.9\tex\latex\tools\bm.sty")
(C:\Users\armstrod\test1.aux)
(C:\Users\armstrod\AppData\Roaming\MiKTeX\2.9\tex\context\base\supp-pdf.mkii
[Loading MPS to PDF converter (version 2006.09.02).]
) ("C:\Program Files (x86)\MiKTeX 2.9\tex\latex\amsfonts\umsa.fd")
("C:\Program Files (x86)\MiKTeX 2.9\tex\latex\amsfonts\umsb.fd")
! Missing $ inserted.
<inserted text>
                $
l.7

?
```

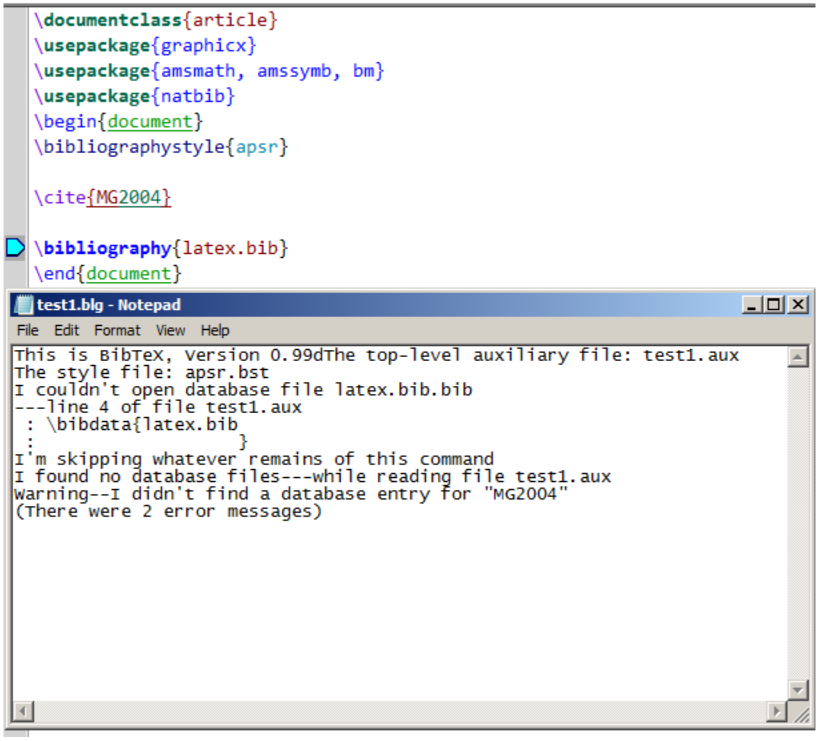
- Here, I forgot to include a figure that LaTeXexpected to be in the same directory as my file. Notice that LaTeXtells me that the file is missing. Typing `r` will allow LaTeXto continue with a stand-in for the missing figure.



```
\documentclass{article}
\usepackage{graphicx}
\usepackage{amsmath, amssymb, bm}
\begin{document}

\includegraphics{missingfig.pdf}

\end{document}
```

```
For additional information on amsmath, use the `?' option.
("C:\Program Files (x86)\MiKTeX 2.9\tex\latex\ams\math\amstext.sty"
("C:\Program Files (x86)\MiKTeX 2.9\tex\latex\ams\math\amsgen.sty"))
("C:\Program Files (x86)\MiKTeX 2.9\tex\latex\ams\math\amsbsy.sty")
("C:\Program Files (x86)\MiKTeX 2.9\tex\latex\ams\math\amsopn.sty"))
("C:\Program Files (x86)\MiKTeX 2.9\tex\latex\amsfonts\amssymb.sty"
("C:\Program Files (x86)\MiKTeX 2.9\tex\latex\amsfonts\amsfonts.sty"))
("C:\Program Files (x86)\MiKTeX 2.9\tex\latex\tools\bm.sty")
(C:\Users\armstrod\test1.aux)
(C:\Users\armstrod\AppData\Roaming\MiKTeX\2.9\tex\context\base\supp-pdf.mkii
[Loading MPS to PDF converter (version 2006.09.02).]
)

LaTeX Warning: File `missingfig.pdf' not found on input line 6.


! Package pdftex.def Error: File `missingfig.pdf' not found.

See the pdftex.def package documentation for explanation.
Type  H <return>  for immediate help.
 ...

l.6 \includegraphics{missingfig.pdf}

? 
```

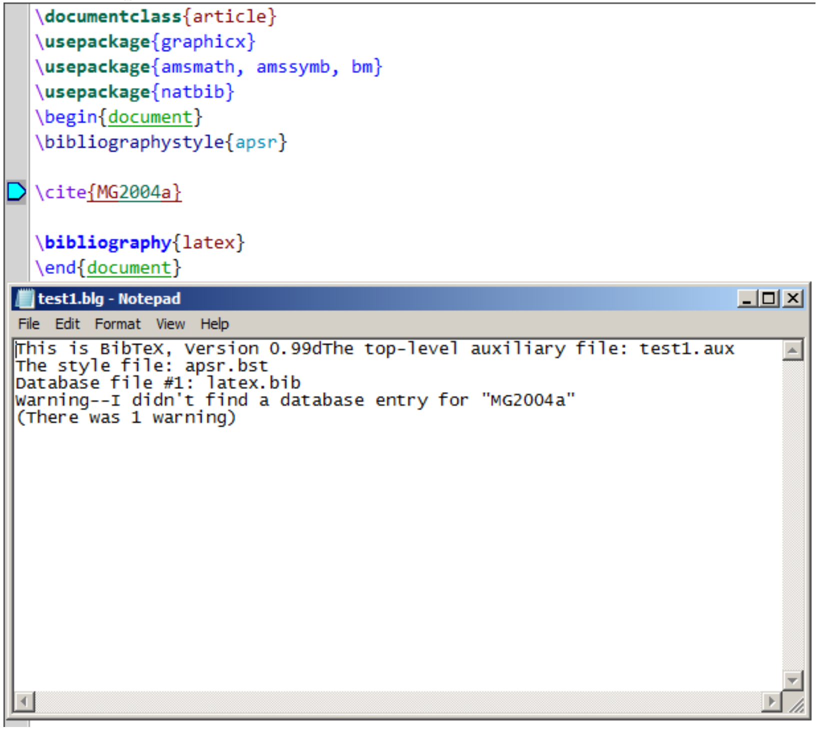
- Here, I used a bib file that didn't exist. Look at the `<<filename>>.blg` to diagnose problems with the BibTeXrun.



```
\documentclass{article}
\usepackage{graphicx}
\usepackage{amsmath, amssymb, bm}
\usepackage{natbib}
\begin{document}
\bibliographystyle{apsr}

\cite{MG2004}

\bibliography{latex.bib}
\end{document}
```

```
This is BibTeX, Version 0.99dThe top-level auxiliary file: test1.aux
The style file: apsr.bst
I couldn't open database file latex.bib.bib
---line 4 of file test1.aux
 : \bibdata{latex.bib
 :                   }
I'm skipping whatever remains of this command
I found no database files---while reading file test1.aux
Warning--I didn't find a database entry for "MG2004"
(There were 2 error messages)
```

- Here, I used a citation that didn't exist in a file that does exist. Again, the `<<filename>>.blg` helps diagnose the problem.



```
\documentclass{article}
\usepackage{graphicx}
\usepackage{amsmath, amssymb, bm}
\usepackage{natbib}
\begin{document}
\bibliographystyle{apsr}

\cite{MG2004a}

\bibliography{latex}
\end{document}
```

```
This is BibTeX, Version 0.99dThe top-level auxiliary file: test1.aux
The style file: apsr.bst
Database file #1: latex.bib
Warning--I didn't find a database entry for "MG2004a"
(There was 1 warning)
```

9 Things We Didn't Cover

There are a number of things we didn't cover that I would be happy to talk with people about in smaller groups or individually as there is interest:

- Sweave, which is a method for integrating R into your LaTeXdocument to ensure perfect reproducibility of your results and to keep your statistics code and writing in the same place.

- Presentations in LaTeX. There are a number of tools for making high-quality presentations in LaTeX. The one I use is called Beamer.

- Games (normal and extensive form) can be made with Martin Osborne's `egameps.sty` file. You can get information about those from CTAN or Drew Dimmery's Page on Game Trees with TikZ.

References

Mittelbach, Frank and Michael Goosens. 2004. *The LaTeX Companion, second edition.* Boston, MA: Pearson Education Inc.