Introduction to LaTeX

Chelsea Goforth
Statistical Consulting Associate

StatLab: Statistics and Data Analysis
University of Virginia Library

9 February 2016

Many thanks to Dave Armstrong, University of Wisconsin–Milwaukee and ICPSR Summer Program 2014, for many of the materials and examples used in this presentation.

Contents

List of Tables

List of Figures

1 Introduction and Getting Started

LaTeX is a typesetting language. Instead of visually formatting your text, you enter your manuscript text intertwined with TeX commands in a plain text file. You then compile this script to produce formatted output, such as a PDF file. Thus, in contrast to standard word processors, your document is a separate file that does not pretend to be a representation of the final typeset output, and so can be easily edited and manipulated.

1.1 Why Use LaTeX?

LaTeX is mainly used for typesetting professional research papers; it is especially useful for the integration of mathematical formulas and equations and statistical output. Other advantages include:

- high typographical quality of the documents
- allows you to clearly separate the content from the format of your document
- integrates nicely with other programs you'll be using, e.g., R, Stata
- cleaner, more refined, more polished documents (some would argue)
- command line typing instead of point-and-click interface
- it's free!
- all the cool kids are doing it!

But there are certainly disadvantages as well. For one thing, it's time-consuming to learn. It's also especially difficult to convert Word and LaTeX documents, which makes collaboration with those using other software difficult as well. The fact that you don't have to worry much about the design is both an advantage and a disadvantage: it's great to let the software compile your document for you when you need to be more concerned with the content, but if you want more control over the design, LaTeX arguably does this less well than other options.

1.2 Getting Started

See Section 8.1 for options for downloading the LaTeX typesetting program by operating system. Most versions come with a plain text editor to write your script, but I *strongly* suggest new users also download TeXStudio because it offers good syntax highlighting, quick templates, helpful references, and some other nice interactivity; see Section 8.2 for other options. Finally, you'll also need a PDF viewer (e.g., Acrobat, Foxit, Skim, Preview), though you probably already have one. You should install LaTeX, the pdf viewer, and any postscript software *before* installing a text editor because most will query your file system to find where all of the necessary software is installed so you won't have to do this manually.

2 The LaTeX Document

The LaTeX document has two main parts—the **preamble** and the **body**. At the bare minimum to start typing/compiling documents in LaTeX, you will need to specify a document class, e.g., `\documentclass{article}` at the beginning of your document and use:

```
\begin{document}
...
...
\end{document}
```

to envelop all text. See this for more information about other document classes you may want to use and this for various templates using these document classes.

However, these three lines will not get you very far before needing to invoke **packages** in order to further customize your document. The preamble sets up the document by invoking any packages that you *might* use, though you don't have to use all of the packages that you load. Be careful, though, to not invoke too many unnecessary packages; because LaTeX relies on open-source coding and user-written functions, it's possible that some packages will conflict with each other, resulting in some very confusing error messages. For this reason, it's often a good idea to add packages one at a time so that 1) you know what each does and 2) you can isolate any errors that are produced. The preamble is also the space that will allow you to define new commands and set properties of the document (like author, title, margins, header style, etc...). The preamble is everything the between `\documentclass[]{}` and `\begin{document}`. The preamble for this handout is as follows:

```
\documentclass[12pt]{article}

\usepackage[american]{babel} % sets language
\usepackage[margin=1in]{geometry} % sets page layout
\usepackage{color,soul} % sets text and background colors
\usepackage{verbatim} % defines environment for un-evaluated code
\usepackage{amsfonts,amsmath,amssymb,amsthm,bm} % math fonts and symbols
\usepackage{setspace} % allows toggling of double/single-spacing
\singlespacing % sets default spacing; also: \doublespacing & \onehalfspacing

\usepackage{parskip,titlesec} % paragraph customization
\usepackage{indentfirst} % indents first paragraphs by default
% use \noindent to manually change just one paragraph with this option
\setlength{\parindent}{0.5in} % sets the indent length

\usepackage[colorlinks=true]{hyperref} % command href for urls
\hypersetup{colorlinks,
citecolor=blue,
urlcolor=blue} % changes default cite color from green to blue and the
```

```
default url color from magenta to blue
%\usepackage{url} % another url package, incl. \url{} field

\usepackage{caption} % customizes captions in floating environments like
figures and tables
\usepackage{graphicx,subfigure,float,wrapfig} % graphics commands
\usepackage{dcolumn,multirow,array} % table properties: decimal-aligned
columns, multi-row cells, column specification
\newcolumntype{d}[1]{D{.}{.}{#1}} % defines a decimal-aligned column

\usepackage{fancyhdr} % increases control over headers
\pagestyle{fancy} % defines new header
\lhead{Intro to \LaTeX}
\rhead{9 February 2016}

\title{{\huge \textbf{Introduction to \LaTeX{}}}} % sets the title of the
document, including bold font and huge font size

\author{{\LARGE Chelsea Goforth} \vspace{5pt} \\ {\Large Statistical
Consulting Associate} \\ \\ {\Large \textsc{StatLab: Statistics and Data
Analysis}} \vspace{4pt} \\ \textsc{University of Virginia Library}} % sets
author information, including font size, bold face, small caps, and a extra
vertical spacing
% use \and to include other authors, for example:
% \author{Chelsea Goforth\\UVA \and Someone else\\U of Somewhere}

\date{9 February 2016} % or, \today to update with every compilation
```

Note my use of the percentage sign – `%` – in this preamble. Using this symbol in LaTeX documents is how you insert comments that will not show up in the compiled documents. LaTeX will ignore anything that follows this symbol when compiling the document, regardless of whether it comes at the beginning or in the middle of a line. Also note that I'm invoking the package names in the curly braces, and options to customize those packages are included within the square brackets.

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}`.

Finally, a small amount of document set-up occurs after you begin the document environment. For example, in this document immediately following the preamble, the following code is all inserted before my first introductory section header where the document begins after the table of contents pages.

```
\begin{document}

\bibliographystyle{apsr} % defines bibliography style
\maketitle % uses title, author, and date info above to make the title
\tableofcontents % creates table of contents
\listoftables % includes tables in table of contents
\listoffigures % includes figures in table of contents
\newpage % starts the rest of the document on the next page
```

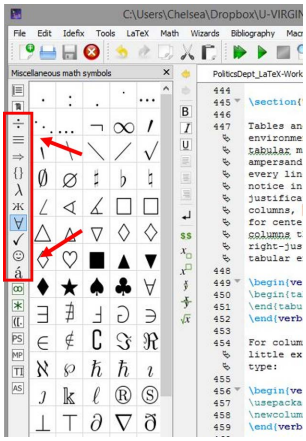
3 Typing in LaTeX

Before we go 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 "LaTeX document," generally I mean one compiled with `pdflatex`; this is the default compiler used with several LaTeX editors, such as TeXStudio.

The characters `# % ^ $ & _ { } ~ \` are LaTeX "special characters," which are either reserved for a different purpose or have a different meaning within a particular environment. To use these characters in your document, use a prefix backslash, e.g., `\&` to produce &. For example, if you're using mathematical symbols or operators outside of a math environment, you'll need to surround the symbol/operator with `$ $`, e.g., `$\alpha$` to produce α.

If you don't know the code for a particular character or symbol that you need, google it! A few helpful site are Detexify, which will allow you to draw exactly what you want to type, and AoPS LaTeX Symbols and this list of LaTeX Math Symbols include a list of helpful symbols, but there are plenty other similar sites out there. If you're using TeXStudio, you can also consult the left hand tool bar, as in Figure 1.

Figure 1: Symbols with TeXStudio



In addition:

- `` is the open double-quote symbol and '' is the closed double-quote symbol (same goes for single quotes, but only one ` or ' in each case).
- `\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.
- `verbatim` is another useful 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|` | or `\verb" "` — for example, the code:

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

  will produce:

  some text `mod <- lm(y ~ x, data=dat)` some more text
- `\vspace{}` and `\hspace{}` will manually insert white space vertically and horizontally, respectively. These are good hacks if the document layout isn't exactly how you'd like it to be. Note that specifying negative length, e.g., `\vspace{-15pt}` will remove white space. You can specify lengths using several different increments depending on your purposes:

  `mm` → millimeter

  `cm` → centimeter (1 cm = 10 mm)

  `in` → inch (1 in = 25.4 mm)

  `pt` → point (1 in = 72.27 pt)

  `pc` → pica (1 pc = 12 pt)

3.1 General Formatting

- `\textbf{bold text}` → **bold text**
- `\textit{italicized text}` OR `\emph{italicized text}` → *italicized text*

  If you use `\emph{}` inside of an italic text environment, it will remove the italics on the emphasized text.
- `\underline{underlined text}` → underlined text
- `\texttt{fixed-width text}` → `fixed-width text` [aka teletype font]
- `\textsc{Small Caps}` → Small Caps

- `text$_{subscript}$` → text$_{subscript}$ and `text$^{superscript}$` → text^{superscript}

  Note again the use of the special symbol `$ $` – they are only necessary outside of a math environment for this formatting as discussed above.

- `\hl{highlighted text}` → highlighted text

- `\textcolor{green}{green text}` → green text

  The predefined colors available through the `color` package are: `white`, `black`, `red`, `green`, `blue`, `cyan`, `magenta`, and `yellow`. For access to more colors and more control over color in general, see the `xcolor` package, but note that if you're also using the `tikz` package, you must declare the `xcolor` package first, otherwise it won't work.

3.2 Page Layout

- `\section`, `\subsection` and `\subsubsection` - all generate sequentially nested numbered sections. The following code:

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

  will produce:

1 Section 1

1.1 Subsection 1

1.1.1 Subsubsection 1

- `\newpage` or `\pagebreak` put everything below starting on a new page.
- `\newline`, `\linebreak` and `\\` all move whatever follows them to a new line; however, if there is "no line to end" then you will get an error message (i.e., they cannot live on a line with no other text in it).

3.2.1 Text Size

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

- `{\tiny tiny sample text}` → tiny sample text
- `{\scriptsize scriptsize sample text}` → scriptsize sample text
- `{\footnotesize footnotesize sample text}` → footnotesize sample text

- `{\small small sample text}` → small sample text
- `{\normalsize normalsize sample text}` → normalsize sample text
- `{\large large sample text}` → large sample text
- `{\Large Large sample text}` → Large sample text
- `{\LARGE LARGE sample text}` → LARGE sample text
- `{\huge huge sample text}` → huge sample text
- `{\Huge Huge sample text}` → Huge sample text

These font sizes 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}` in your preamble 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.

3.2.2 Text Spacing

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. For example:

```
\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.

Like font size commands, spacing commands can also be used to define environments, e.g., `\begin{onehalfspace}` and `\end{onehalfspace}`.

3.3 Other Common and Useful Environments

Enumerate, Itemize, and Description are all 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.

3.3.1 itemize

The following code:

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

will produce:

- First bullet point

You can also change the itemize symbol by putting the symbol you want in square brackets `[ ]` after the `\item` statement:

– is produced by `\item[--]`

∘ is produced by `\item[$\circ$]`

• is produced by `\item[$\bullet$]`

⋆ is produced by `\item[$\star$]`

⧫ is produced by `\item[$\blacklozenge$]`

■ is produced by `\item[$\blacksquare$]`

✓ is produced by `\item[\checkmark]`

→ is produced by `\item[$\rightarrow$]`

▸ is produced by `\item[\tiny{$\blacktriangleright$}]`

[Note that I also changed the font size here; see Section 3.2.1]

3.3.2 enumerate

The following code:

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

will produce:

1. First bullet point

You can also nest enumerate, itemize, and description environments. For example:

```
\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}
```

will produce the following:

1. First item
    (a) First sub-item
        i. First sub-sub-item
            A. First sub-sub-sub-item

3.3.3 description

The following code:

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

will produce:

First bullet point

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 (`l` for left-justified columns, `r` for right-justified columns and `c` for centered columns). For example for a table with four columns that are left-justified, centered, centered, and right-justified, respectively, we would use the following tabular environment:

```
\begin{tabular}{lccr}
\end{tabular}
```

For columns aligned on a decimal point, we need to do a little extra work in our preamble to define a new column type:

```
\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 LaTeX to leave enough room for two decimal places. 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}}
\end{tabular}
```

The table environment usually envelopes the tabular environment. The table environment is a float environment (like figures; see Section 5), which allows you to add a caption to tables and use the `\label{}` command in order to be able to reference and link to them elsewhere in the text. See the following sections for more detailed table examples.

4.1 Single-tables in the `table` environment

```
\begin{table}[H]
\caption{Percentage of Politics students \\ using various statistical
software packages}\label{tab:pack1}
\centering
\begin{tabular}{ld{2}d{2}d{2}d{2}} \hline
& \multicolumn{4}{c}{Packages}\\ \hline
& \multicolumn{1}{c}{SPSS} & \multicolumn{1}{c}{Stata} & \multicolumn{1}{c}{R}
& \multicolumn{1}{c}{SAS} \\ \hline\hline
1st Year & 43.25 & 38.15 & 18.6 & 0.0 \\
2nd Year & 5.5 & 58.15 & 36.35 & 0.0 \\ \hline\hline
```

```
\end{tabular}
\end{table}
```

Table 1: Percentage of Politics students using various statistical software packages

	Packages			
	SPSS	Stata	R	SAS
1st Year	43.25	38.15	18.6	0.0
2nd Year	5.5	58.15	36.35	0.0

The following example produces the same table, but shows how to use the vertical pipe in defining column types to produce a vertical border:

```
\begin{table}[H]
\caption{Percentage of Politics students \\ using various statistical
software packages}\label{tab:pack2}
\centering
\begin{tabular}{l|d{2}d{2}d{2}d{2}} \hline
& \multicolumn{4}{c}{Packages}\\ \hline
& \multicolumn{1}{c}{SPSS} & \multicolumn{1}{c}{Stata} & \multicolumn{1}{c}{R}
& \multicolumn{1}{c}{SAS} \\ \hline\hline
1st Year & 43.25 & 38.15 & 18.6 & 0.0 \\
2nd Year & 5.5 & 58.15 & 36.35 & 0.0 \\ \hline\hline
\end{tabular}
\end{table}
```

Table 2: Percentage of Politics students using various statistical software packages

	Packages			
	SPSS	Stata	R	SAS
1st Year	43.25	38.15	18.6	0.0
2nd Year	5.5	58.15	36.35	0.0

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{Entering Politics Graduate Students in 2013 and 2014}\label{tab:subtab}
\centerline{\hbox{ \subtable[2013]{
```

```
\begin{tabular}{lcc}
& Men & Women\\ \hline
Master's & 4 & 2 \\
No Master's & 2 & 6 \\ \hline
\end{tabular}\label{tab:subtab1}}
\quad \quad \subtable[2014]{
\begin{tabular}{lcc}
& Men & Women\\ \hline
Master's & 5 & 3 \\
No Master's & 5 & 3 \\ \hline
\end{tabular}\label{tab:subtab2}}}}
\end{table}
```

Table 3: Entering Politics Graduate Students in 2013 and 2014

(a) 2013

	Men	Women
Master's	4	2
No Master's	2	6

(b) 2014

	Men	Women
Master's	5	3
No Master's	5	3

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 TeXStudio, LaTable, and Table Editor

First, if you're using TeXStudio, you can navigate to **Wizards** in the pull-down menu, which provides several different quick start functions to choose from, then select **Quick Tabular**; see a screenshot of the interface in Figure 2. See the following subsections for other easy table options.

Figure 2: TeXStudio Quick Tabular

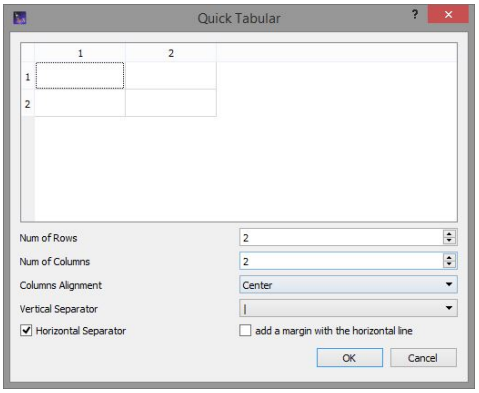
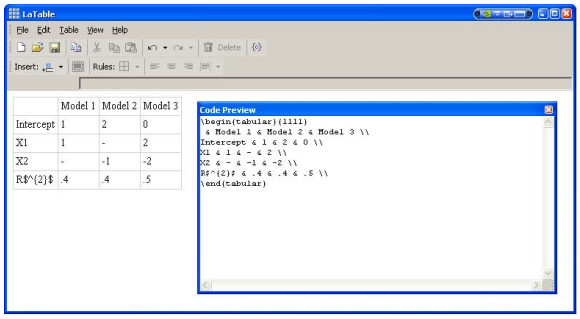


Figure 3: Screenshot of LaTable Interface



In addition, 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; see Figure 3 and Table 4.

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

Table Editor is another such program with similar functionality.

4.3.2 Models from R into LaTeX

In R, you could do the following to produce Table 5:

```
> 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 table 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, you could do the following to first store the results from three different regression models and then format a "tex-style" table:

```
use "/Users/Chelsea/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

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("" "" "")
```

This will put Table 6 in the file `statamods.tex` and will print it to the screen (from which you could just copy and paste into your LaTeX file).

Table 6: Example table 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)

Or, you could do the following in LaTeX (after creating the `statamods.tex` file using Stata):

```
\begin{table}[h!]
\caption{Example table 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 (note the `s`, not `x`).

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}`.

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

  `width` and `height` scale the width and height to the specified values (e.g., [`height=2in`])

  `scale` scales the image by the given amount (e.g., [`scale=0.5`])

  `keepaspectratio` preserves the aspect ratio of the original figure

  `angle` angle of figure rotation (in degrees counterclockwise)

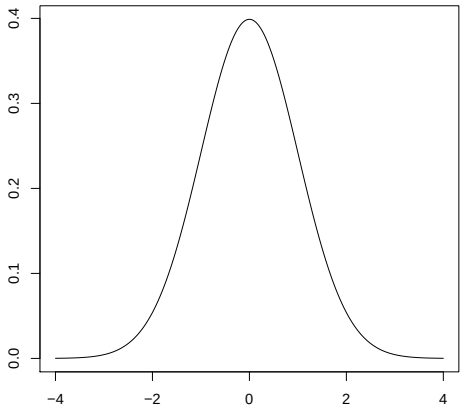
- It is easiest to put graphs in the directory where your document is. That is where LaTeX is going to look for them by default. However, there is a command you can put in the preamble: `\graphicspath{{dir1}{dir2}{dir3}...}`. However,

  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!]
\centering
\caption{Normal Density over [-4,4], $\mu$=0,  $\sigma$=1}
\includegraphics[width=2.75in,keepaspectratio]{norm1.pdf}
\label{fig:onefig}
\end{figure}
```

Figure 4: Normal Density over [-4,4], μ=0, σ=1



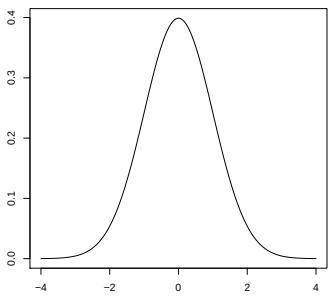
I can reference Figure 4 with `Figure~\ref{fig:onefig}`, which I could not do without the figure environment (nor could I specify a caption) or the label `fig:onefig`.

Note my naming conventions for float environments like tables and figures: all table labels start with `tab`, all figure labels start with `fig`, and you'll see shortly that all my equation labels start with `eq` (similarly, all of my section labels begin with `sec`; although these are not floating environments, in-document naming conventions remain the same).

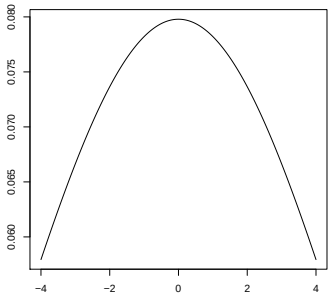
5.3 Including Multiple Figures with `subfigure`

```
\begin{figure}[h!]
\caption{Three Normal Densities over [-4,4] with $\mu$=0}
\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}
```

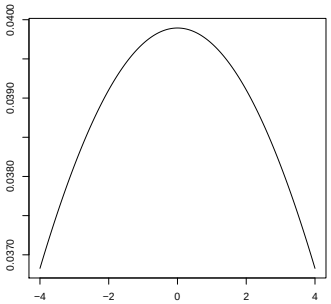
Figure 5: Three Normal Densities over [-4,4] with μ=0



(a) σ=1



(b) σ=5



(c) σ=10

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

6 Equations

One of the biggest advantages 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, it is much faster to type than use point-and-click menus. There are two math 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:

```
\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}
```

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

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

For example:

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

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

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

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

will produce:

$$x + 11 = 3 \tag{3}$$
$$x = 3 - 11 \tag{4}$$
$$x = -8 \tag{5}$$

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 noted above. For example, the matrix algebra generating regression coefficients is $(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$, which was obtained using the code: `$\left(\bm{X}^{\prime}\bm{X}\right)^{-1}\bm{X}^{\prime}\bm{Y}$`.

7 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 LaTeX will 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, LaTeX will quit and allow you to fix the error. In TeXStudio, see the **Messages / Log File** window at the bottom.

One of the most common ways to generate an error is to not close a delimiter (that LaTeX is 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 LaTeX thought it should find. When you don't understand an error message or you don't know how to get LaTeX to do what you want it to do, use the google!

8 Distribution, Text Editors, and Ancillary Stuff

8.1 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.

8.2 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've also heard good things about 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 *strongly* suggest TeXStudio for beginners because it offers good syntax highlighting, quick templates, helpful references, and some other nice interactivity; 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.

8.3 Ancillary Stuff

Windows users might also want to to install Ghostscript and Ghostview, which are programs that facilitate working with postscript documents. Mac users will likely already have facility to work with these files.

9 Things We Didn't Cover

There are a number of things I didn't cover much (or at all...); however, if you have any questions about these or anything else in this presentation, please don't hesitate to ask!

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

10 Helpful References

- Mittelbach and Goossens (2004): The LaTeX Companion
- Oetiker (last updated July 2015): The Not So Short Intro to LaTeX
- Wikibooks: LaTeX