
An Introduction to L^AT_EX

MATTHEW DENNY

FRIDAY 26TH SEPTEMBER, 2014

What is L^AT_EX?

L^AT_EX is an document markup language, which is kind of like what goes on under the hood in MS word or OpenOffice when you format one of these documents. The L^AT_EX project website has a nice little intro here: <http://latex-project.org/intro.html>. You can also check out the Wikipedia site but I will be going over this during the workshop.

The Basic Structure of a Document

Before we dive into creating our first document we need some preliminaries. I am going to assume you are using TeXMaker for the purposes of this tutorial and some of the directions are specific to buttons that exist in TeXMaker.

Creating a L^AT_EX folder

Before we actually get started with anything, I strongly suggest you create one folder to hold all of your L^AT_EX documents and source code. A L^AT_EX document is composed of atleast 5 separate files and I find it to be best practice to have all L^AT_EX documents in the same folder. Here is an example, my second assignment for the Advanced Topics in Econometrics class I took last fall:

```
703_HW2.aux
703_HW2.log
703_HW2.out
703_HW2.pdf
703_HW2.tex
```

While there are a bunch of files here, we only care about two: The

```
703_HW2.pdf
```

file is the final output which is what I printed and handed in, and

```
703_HW2.tex
```

is the L^AT_EX source file which I wrote in TeXMaker to actually create my homework assignment. Given that this folder can quickly have a lot of files floating around in it, descriptive titles are a must. One important thing to note is that your document titles cannot contain spaces, so using underscores is a good solution. The nice thing is that once you name your .tex file, all of the accompanying files will take the same name and you don't have to worry about them.

Commenting

Since L^AT_EX is an interpreted language, this allows us to leave notes for ourselves, or to comment out whole sections of our documents depending on our audience. If I decide I want to use a different introductory paragraph in a paper, for example, I will often comment out the old intro and keep it in my L^AT_EX source file. That way, if I even want to go back to the old wording, it is right there for me and all I have to do is un-comment it. Below is an example with L^AT_EX source on the left and output on the right:

```
Here is an example of commenting:
%Percent symbols make any text on
%the same line after them not show up
```

Here is an example of commenting:

If you are using TeXMaker, there are some handy keyboard shortcuts you can use to comment out text. On a Mac keyboard, highlighting text and pressing **Command + T** will comment out the highlighted text and **Command + U** will un comment it. On a Windows/Linux keyboard, this is **Control + T** and **Control + U** respectively. I will use comments throughout this document as a way of reminding you what different commands do.

Your First Document:

We will begin with a very basic document. Our goal here is to create a document that will compile and produce PDF output. On the very top left corner of your screen, click the little white page with a corner folded up and a green plus sign in the top right corner. You should now see a blank document titled: **untitled1** or something like that appear in your editor. Before you do anything else, save this document with an informative title by going to the **File** drop-down menu and selecting **save as**. Here again I suggest you save this in your designated \LaTeX folder (I just call mine \LaTeX and I have it in my DropBox so I can access it from multiple computers). I am going to save my document as: **LaTeX Intro Workshop Example 1**. Now that we have this out of the way, we can copy and paste in the following code, or type it in by hand.

```
\documentclass{article} %tells LaTeX that we want to use the framework of an article as our
%starting point for this document

\title{Our First Document!} % you can specify the title here

\author{Matthew Denny} % author's name

\date{\today} % The date can be \today which automatically updates or can be specified as
% any date you like

\begin{document} % anything that appears on the screen has to be after this command

\maketitle %renders the title, author and date information we ave LaTeX

Hello World! %some text

\end{document} % all documents must contain this statement at the end of the document

Any text that is entered after this point, even if it is un-commented, will not appear in the output.
```

Once you have pasted this code, click on the little arrow next to the words **Quick Build** on the top margin of the TeXMaker window and this will create your first document!

Formatting

There are a number of things I can imagine wanting to do to text, as far as formatting is concerned, and I will walk you through some of the most important aspects of formatting in this section:

Line breaks, spacing and indenting

This is probably the most important thing you need to know to get started:

```
This is a line break \\
We can skip a line by doubling this \\\
We can also skip a line using \bigskip
%which is useful if there is no text on a line
\bigskip

Normally if we skip a line in our code
the first line will be indented\\

\noindent We can use this command to
keep this from happening.\\
We can also add spaces \hspace{.75in} in inches
or in centimeters \hspace{1cm} or in
\hspace{20pt}points\\\\
finally notice that multiple whitespaces
or line breaks
in your source are only interpreted as a
single space by \LaTeX
```

This is a line break
We can skip a line by doubling this

We can also skip a line using

Normally if we skip a line in our code the first line will be indented

We can use this command to keep this from happening.
We can also add spaces in inches
or in centimeters or in points

finally notice that multiple whitespaces or line breaks in your source are only interpreted as a single space by \LaTeX

Changing the appearance of text

We can bold, italicize, change the size and color of text and add emphasis:

```
Lets start with plain text \\
Make it \textbf{bold} \\
Make it \textit{italicized} \\
Make it \color{red} red \color{black} \\
Make it \Huge huge\\
Make it \tiny tiny \normalsize \\
\emph{Emphasize a \emph{particular} word}
\begin{center}
We can also center our text this way
\end{center}
or we can center our text

\centering this way % be sure to include a space
%between lines when using the \centering command
```

Lets start with plain text
Make it **bold**
Make it *italicized*
Make it red
Make it **huge**
Make it tiny
Emphasize a particular word

We can also center our text this way

or we can center our text
this way

One more useful thing we can do with our text is to turn it into a hyper-link to a website. To do this, we need the following package in our header:

```
\usepackage{hyperref}
```

You will find that for much of the functionality we can add to \LaTeX is through the use of add on packages like this one. All of these add on packages are part of your TeX distribution but they are not all loaded at the same time. We can create a hyper-reference as in the following example:

```
\href{http://en.wikibooks.org/wiki/LaTeX}{The \LaTeX\ Wikibook!}
```

This gets rendered as a link: [The \$\text{\LaTeX}\$ Wikibook!](http://en.wikibooks.org/wiki/LaTeX)

Headers and page geometry

To show you how to change the page layout geometry of your document, we need to take a little detour first and explain the idea of document headers. Headers are all of the stuff that comes before your

```
\begin{document}
```

statement in your \LaTeX source code. In our first document example:

```
\documentclass{article}
\title{Our First Document!}
\author{Matthew Denny}
\date{\today}
```

These are all headers. We can add a near infinite amount of flexibility to our \LaTeX documents by modifying this header. Now lets take a look at the block of header code we can add to our simple document in order to change the page layout dimensions if we want to:

```
% Page layout (geometry)
\usepackage{geometry}
%LaTeX naturally believes that the page should start one inch to the right and below
%the top left corner of the page
\setlength\voffset{-1in} %This resets the vertical offset assumption so LaTeX uses the true
%top of the page as a reference point
\setlength\hoffset{-1in} % this does the same thing for the horizontal dimension.

\setlength\topmargin{1in} %sets the top margin
\setlength\oddsidemargin{1in} %sets the left margin
\setlength\textheight{9in} %sets the height of the text area (and implicitly the bottom margin)
\setlength\textwidth{6.5in} %sets the text width (and implicitly the right side margin)
```

```
\setlength\footskip{0.0cm} % sets the distance between text and footnotes
\setlength\headheight{0cm} %sets the height of the doument header
\setlength\headsep{0cm} %sets the distance between header and text
```

Page layout is discussed in much greater detail on this web page, which you should refer to if you want to do any serious tinkering with the page layout: <http://help-csli.stanford.edu/tex/latex-margins.shtml>. The best way to see how this works is to paste a bunch of text into your document, put these statements into your header and start changing the numbers. Remember that the numbers you choose are based on the size of paper you are using (in our case LaTeX should default to 8.5×11 inches).

Sections, subsections, subsubsections and organization

One of the most important aspects of authoring documents in \LaTeX is that the engine tends to *enforce* a certain level of structure on our documents above and beyond what MS word will do for us. \LaTeX is designed to enforce a logically consistent document structure and the use of sections and subsections and even subsubsections is vital in doing this. Here are some examples:

```
\section{The first section}
I can type some text here
\subsection{Now we dive further into our introduction}
Notice that \LaTeX\ automatically numbers our sections
\subsubsection{Getting even more in-depth}
we can say something here.
\subsubsection{furthermore}
we can say something here
\subsection{} %notice we do not need to put any information in section titles
\section{The second section}
\section{} %the third section does not need a title
\subsection{sub section 1} %notice that the numbering automatically goes back to 1
\subsection{}
```

Now while we may really like the auto numbering for our sections but we may prefer a look with just section titles. To do this, we can use the code below.

```
\section*{The first section} % note the star after section, this is how LaTeX knows not to number
I can type some text here
\subsection*{Now we dive further into our introduction}
Notice that \LaTeX\ automatically numbers our sections
\subsubsection*{Getting even more in-depth}
we can say something here.
```

In general, if \LaTeX auto-numbers something, putting a star after the command will probably stop this.

Quotes, Quotations and Footnotes

One thing we tend to like to do quite a bit in our papers is quote other papers. One of the little quirks of \LaTeX is that there are different symbols for an open and close quotes:

```
"this uses the double quotes button twice" \\
‘this uses the tilde button and then the double quotes button’
```

”this uses the double quotes button twice”

“this uses the tilde button and then the single quote button twice”

You will just need to decide which you like better. Now if we would like to include a block quote then we use the following environment:

```
\begin{quote}
Adding colors to your text is supported by the color package. Using this package, you can set the color
of the font of the text, and set the background color of the page. You can use one of the predefined
colors such as white, red, or yellow, or you can define your own named colors. It’s also possible
to color formulas in math-environments. \href{http://en.wikibooks.org/wiki/LaTeX/Colors}
{The \LaTeX\ wikibook, Colors page}
\end{quote}
```

This will render as a really nice indented quote:

Adding colors to your text is supported by the color package. Using this package, you can set the color of the font of the text, and set the background color of the page. You can use one of the predefined colors such as white, red, or yellow, or you can define your own named colors. It's also possible to color formulas in math-environments. [The \$\TeX\$ wikibook, Colors page](#)

Finally we can add footnotes to our documents with some relatively simple code

```
I am saying something\footnote{This is a footnote}, footnotes are pretty cool!
```

I am saying something¹, footnotes are pretty cool!

Bibliography

There are several ways to create a bibliography in \TeX . The best way is to use BibTeX which is described in the resources section at the end of this document. If you use lots of citations there is really no excuse for not learning BibTeX but here is a slightly more straightforward way of creating a bibliography. To do this, we have to include the following statement in our preamble:

```
\bibliographystyle{apalike}
```

A nice little explanation of this statement and its use (along with more complicated stuff) is available here: http://www.colorado.edu/physics/phys4610/phys4610_sp13/bibtex_guide.pdf

This will let us cite books and articles by using the following command:

```
This is a book about TeX that I have not read \cite{knuth84} and here is another
book \cite{anotherbook}
```

We can create a bibliography by using the following commands:

```
\begin{thebibliography}{99}
\bibitem{knuth84} Knuth, D.E. \emph{The TEXbook}, Addison-Wesley, Reading,
Massachusetts, second edition, 1984.
\bibitem{anotherbook} Leslie Lamport \emph{\LaTeX \rm: A Document Preparation System},
Addison-Wesley, 1986.
\end{thebibliography}
```

This is a book about TeX that I have not read [1] and here is another book [2]

References

[1] Knuth, D.E. *The TEXbook*, Addison-Wesley, Reading, Massachusetts, second edition, 1984.

[2] Leslie Lamport *\TeX : A Document Preparation System*, Addison-Wesley, 1986.

The neat thing about bibliographies in \TeX is that they are automatically generated for us and hyper-linked (although not alphabetized) so all you have to do is add entries in the correct order. One important thing to note is that you may have to render the document several times for the bibliography and citations to display correctly. Just click the arrow next to **Quick Build** several times and this should work for you. 5

5

Math Math Math

The elegant simple and conversational way that \TeX handles math is one of its main attractions and one of the primary reasons for its existence as it is used heavily in STEM fields. We will go over a number of things but I will also leave you a number of reference websites and send you some math/symbol PDF guides. It is very important to note that all formal math should be written in a math environment (described below) or your document will not compile. Math environments give \TeX more ability to handle complex equations. To get access to more math symbols, we can add the following line to our preamble:

```
\usepackage{amsmath,amssymb,amsfonts,amsthm,textcomp}
```

You may notice that we can add more than one package in a usepackage statement, we just have to separate them by commas.

¹This is a footnote

The three ways of displaying equations

There are normally three ways in which you might want to display equations: in-line with text, as a numbered (or un-numbered) equation on its own line, or as a block of equations. We will look at all three of these (note that you may not understand the syntax, but we will go over that in the next section).

The simplest way of inserting math into a paragraph is to use the `\$` sign on each side of the equation as follows: for example we might have a function `\$y_A = x_1^2 + 3x_2^4\$` that we want to display inline.

The simplest way of inserting math into a paragraph is to use the `$` sign on each side of the equation as follows: for example we might have a function $y_A = x_1^2 + 3x_2^4$ that we want to display inline.

While this method can be convenient for small equations and those that do not contain fractions, it is not really that elegant and misses one of the best points of \LaTeX : its equation environment. This environment allows us to create numbered equations, but only allows one equation or line per instance, so if you need to simplify something or go onto multiple lines we will need something different. A sample equation is shown below:

The FGLS estimator is:

```
\begin{equation}
\widehat{\beta}_{GLS} = \left[ \sum_n \left( \frac{1}{\sigma_n^2} x_n x_n' \right) \right]^{-1} \left[ \sum_n \left( \frac{1}{\sigma_n^2} x_n y_n \right) \right]
\end{equation}
```

We see that this is somewhat complicated.

The FGLS estimator is:

$$\hat{\beta}_{GLS} = \left[\sum_n \left(\frac{1}{\sigma_n^2} x_n x_n' \right) \right]^{-1} \left[\sum_n \left(\frac{1}{\sigma_n^2} x_n y_n \right) \right] \quad (1)$$

We see that this is somewhat complicated.

While the equation environment is great for single equations, it can't handle blocks of equations. For that we will need either the **align** or **eqnarray** environments. Each of these has its own use. The **align** environment makes a series of equations one under the other look really nice stacked on top of each other. The **eqnarray** environment will let us create multiple equations evenly spaced on a line and in columns:

We can now calculate some summary statistics

```
\begin{align}
s^2 &= y'y - (y'X)(X'X)^{-1}(X'y) \\
R^2 &= 1 - \frac{s^2}{y'y} \\
\widehat{\sigma}^2 &= (y - X'\widehat{\beta})'(y - X'\widehat{\beta}) \\
&= y'y - 2\widehat{\beta}'X'y + \widehat{\beta}'XX'\widehat{\beta}
\end{align}
```

We can now calculate some summary statistics

$$s^2 = y'y - (y'X)(X'X)^{-1}(X'y) \quad (2)$$

$$R^2 = 1 - \frac{s^2}{y'y} \quad (3)$$

$$\hat{\sigma}^2 = (y - X'\hat{\beta})'(y - X'\hat{\beta}) \quad (4)$$

$$= y'y - 2\hat{\beta}'X'y + \hat{\beta}'XX'\hat{\beta} \quad (5)$$

Notice that we use line breaks `\\` to separated our equations. Now while this style may be fine for some settings, we may want to line up all of our equations around the equals sign by using the `&=` sign:

Now these are lined up nicely:s

```
\begin{align}
s^2 &=& y'y - (y'X)(X'X)^{-1}(X'y) \\
R^2 &=& 1 - \frac{s^2}{y'y} \\
\widehat{\sigma}^2 &=& (y - X'\widehat{\beta})'(y - X'\widehat{\beta}) \\
&=& y'y - 2\widehat{\beta}'X'y + \widehat{\beta}'XX'\widehat{\beta}
\end{align}
```

Now these are lined up nicely:

$$s^2 = y'y - (y'X)(X'X)^{-1}(X'y) \quad (6)$$

$$R^2 = 1 - \frac{s^2}{y'y} \quad (7)$$

$$\hat{\sigma}^2 = (y - X'\hat{\beta})'(y - X'\hat{\beta}) \quad (8)$$

$$= y'y - 2\hat{\beta}'X'y + \hat{\beta}'XX'\hat{\beta} \quad (9)$$

We may also find that we want to create an array of equations like this:

```
\begin{eqnarray}
2a + 3b = 7 & & 4b + 5c = 49 \\
7c - 2d = 12 & & 2a + 3c = 464 \\
13 = a + b + c + d
\end{eqnarray}
```

$$2a + 3b = 7 \quad 4b + 5c = 49 \tag{10}$$

$$7c - 2d = 12 \quad 2a + 3c = 464 \tag{11}$$

$$13 = a + d \quad 6 = f \tag{12}$$

Finally we may find that we do not really want any equation numbering. If this is the case, we can use the * to remove numbering in each of these environments. Here is an example with an equation:

```
All that changes is the star:
\begin{equation*}
\widehat{\beta}_{GLS} = \left[ \sum_n \left( \frac{1}{\sigma_n^2} x_n x_n' \right) \right]^{-1}
\left[ \sum_n \left( \frac{1}{\sigma_n^2} x_n y_n \right) \right]
\end{equation*}
```

All that changes is the star:

$$\hat{\beta}_{GLS} = \left[\sum_n \left(\frac{1}{\sigma_n^2} x_n x_n' \right) \right]^{-1} \left[\sum_n \left(\frac{1}{\sigma_n^2} x_n y_n \right) \right]$$

Finally we can choose whether we want our equations to appear on the center of the page, or to the left side (my preference). In our normal environment, equations will display like this:

$$A'B = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix} \tag{13}$$

but if we use the following alteration to our document header:

```
\documentclass[fleqn]{article}
```

Then our equations will align like this:

$$A'B = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix} \tag{14}$$

Please note that there cannot be any spaces between equations as this will give you an error and prevent your code from compiling:

```
\begin{equation}
x + y = 7
\end{equation}
%this will give you an error and will not compile
```

Math symbols and operators

Well after that last section, you have seen quite a lot of math, but now we are actually going to go over some useful operators. There are more symbols and operators than you can imagine in \LaTeX (over 100,000 last time I checked) and I will provide you with a cheat sheet for the most frequently used ones and a several hundred page PDF of less frequently used ones. So now lets look at some basic math operators:

```
$ x_w %subscripts of only one character
Y_{A1} % multiple characters need brackets
x^w %superscripts of only one character
Y^{2x + 1} % and of multiple characters
Y_{A1}^{2x_1 + 5} %sub and superscripts
\frac{1}{2} % this is a fraction
\frac{\frac{2x_4 + 1}{6x_2^3}}{27} % a more
complicated one
\max_{x_1, x_2} (x_1, x_2) %the max function
\min_{x_1, x_2} (x_1, x_2) % the min function
\sum_{i = 1}^{\infty} \frac{6x^2}{n!} % summation
\left[ \frac{2x^2}{32} \right] %big brackets
\max x^2 \text{ subject to } no spaces otherwise
$
```

$$\begin{array}{ccc}
 x_w & Y_{A1} & x^w \\
 Y^{2x+1} & Y_{A1}^{2x_1+5} & \frac{1}{2} & \frac{2x_4+1}{6x_2^3} \\
 \max_{x_1, x_2} (x_1, x_2) & & \min_{x_1, x_2} (x_1, x_2) \\
 \sum_{i=1}^{\infty} \frac{6x^2}{n!} & & \left[\frac{2x^2}{32} \right] \\
 \max x^2 \text{ subject to } \textit{nospacesotherwise}
 \end{array}$$

Please note that the max and min operators will only display properly (with the variables we are max/minimizing over under the max/min operator) if we are using an equation or align environment. We will get something that looks like this:

$$\min_{\beta} Q(\beta) = \left[\sum_i \{z[y - \exp(x'\beta)]\} \right]' \left(\sum_i z_i z_i' \right)^{-1} (\beta) \tag{15}$$

Tables

We are now going to move on to tables. These are another area where \LaTeX really shines as you can produce publication quality tables of arbitrary complexity and elegance. Before we can create any tables, we need to add a statement to our preamble:

```
% we can also make use of the tabularx package for more formatting options
\usepackage{booktabs} % lets us draw lines in tables
```

This package allows us to create beautiful tables, lets look at a simple table:

```
{\centering
\begin{tabular}{l l l }
\toprule
& Left & Right \\
\midrule
Top & 3,3 & 1,5 \\
Bottom & 5,1 & 2,2 \\
\bottomrule
\end{tabular}
\par}
```

	Left	Right
Top	3,3	1,5
Bottom	5,1	2,2

There are several things we need to know about the code above. First, we see that we are using the **tabular** environment. The `{l l l}` tells \LaTeX how to align text in each of the columns. We can specify l for left, c for center and r for right, you will just have to try playing around with this. We use `&` 's to separate columns and `\\` to separate lines. You can make a lot of neat simple table in \LaTeX but lets move on to some more complicated stuff:

```
{\centering
\begin{tabular}{lcccc}
\toprule
\multicolumn{4}{r}{Community} \\
\end{tabular}
```



```

\cmidrule(r){2-5} %lets us draw a line only over certain columns
\multicolumn{5}{r}{Same \hspace{69pt} Different}\\ %lets us put text over multiple columns
\midrule
&\color{blue} Cooperate & \color{red}Defect & \color{blue} Cooperate & \color{red}Defect\\
\color{blue} Cooperate& 3,3 & 3,2 & 3,3 & 1,5\\
Community Update? & No & Yes & Yes & No\\
Break Ties? & No & Yes & No & Yes\\
\midrule %draws a line across the table
\color{red} Defect & 2,3 & 2,2 & 5,1 & 2,2\\
Community Update? & Yes & No & No & No\\
Break Ties? & Yes & No & Yes& No \\
\bottomrule
\end{tabular}
\par}
\bigskip

```

Now we can take a look at the output:

	Community			
	Same		Different	
	Cooperate	Defect	Cooperate	Defect
Cooperate	3,3	3,2	3,3	1,5
Community Update?	No	Yes	Yes	No
Break Ties?	No	Yes	No	Yes
Defect	2,3	2,2	5,1	2,2
Community Update?	Yes	No	No	No
Break Ties?	Yes	No	Yes	No

Importing Tables from Excel

I do not have much experience with importing tables directly from excel into \LaTeX but I have done some hunting and there are three avenues I suggest you follow if you are interested in doing this. The simplest way is to export the table in Excel as a PDF or PNG and then use the include graphics statement described below to add the table to your document. There is also a macro available for Excel (that apparently works most of the time) that will generate \LaTeX code from your excel talbes in excel that you can then paste into your document. This will require some fidgeting on your part but a link to the website is available here: <http://www.ctan.org/tex-archive/support/excel2latex/>

The third option is to convert the .xls file to a .csv and then import this file into \LaTeX . This soultion is detailed in the web page below, along with a number of other options: <http://ggorjan.blogspot.com/2008/09/excel-and-latex.html>

Graphs ,Graphics and Verbatim

\LaTeX has a vast number of advanced packages for creating and integrating graphics into your documents. In this section I will give you a taste one making your own graphs and including graphics.

Example Graph

Below is some example code to create a graph of one of the principal-agent problems in Sam Bowles Microeconomics textbook.

```

\begin{center}
\setlength{\unitlength}{5cm} %this scales the unit size of the entire picture
\begin{picture}(2,2) % here we set the unit dimensions of the picture

\put(0,0){\line(0,1){2}} % this statement can be read as: at the point zero, zero
%(the bottom left corner of our picture) start a line that heads straight up
%(the y coordinate direction) that has a length of two

```

```

\put(0,0){\line(1,0){2}}

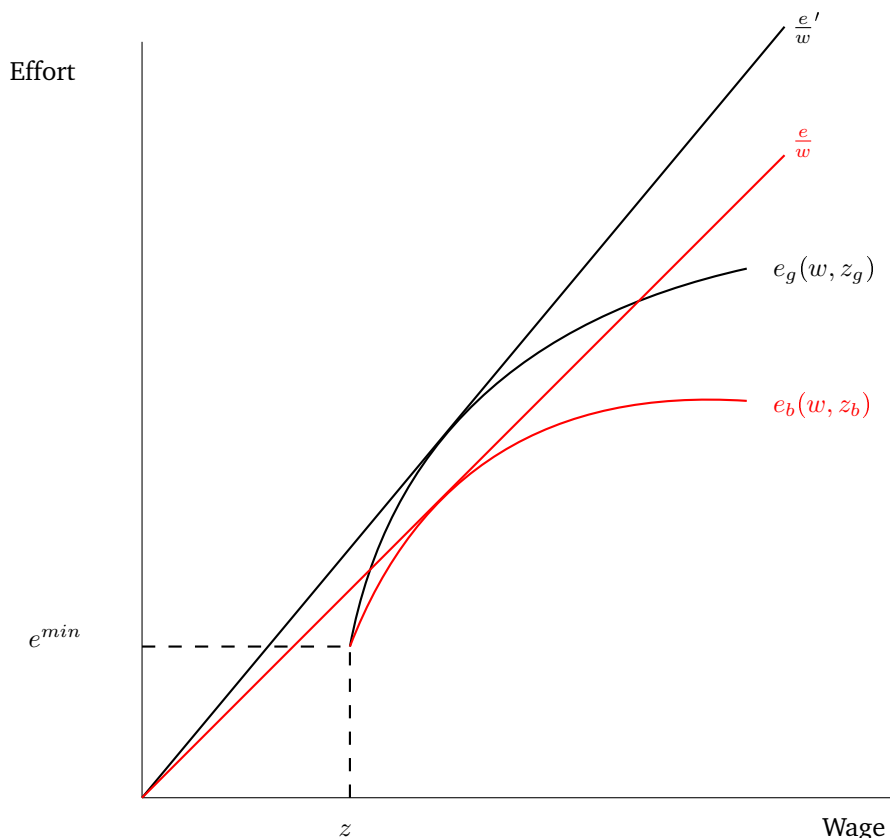
\put(1.8,-.1){Wage} % we can use \put statements to add labels
\put(-.35,1.9){Effort}
\put(1.67,1.38){$e_g(w,z_g)$}
\put(1.72,2.02){$\frac{e}{w}$}
\put(.52,-.1){$z$}
\put(-.3,.4){$e^{\min}$} % we can even add inline math

\thicklines % this makes all lines after it in the picture thicker
\multiput(.55,0)(0, 0.07) {6} {\line(0,1){0.035}} % This is how we create dotted lines.
%The first parentheses tell the lines where to start, the second parentheses tell
%LaTeX how far apart each line segment should be. The bracketts tell LaTeX how many line
%segments there should be. Finally the fourth set of brackets with the \line inside them tell LaTeX
%what the dotted lines should be (what direition they should point and how long )
\multiput(0,.4)(0.07, 0) {8} {\line(1,0){0.035}}
\qbezier(.55,.4)(.7, 1.2)(1.6,1.4) % bezier curve lets us draw curved lines of nearly
% any shape. The first and last set of coordinates specify the endpoints of the line
% and the middle set of coordinates set the shape of the curve

\put(0,0){\line(5,6){1.7}}
\color{red} % we can change the color of all the lines after this statement to red
\put(0,0){\line(1,1){1.7}}
\qbezier(.55,.4)(.82,1.1)(1.6,1.05)
\put(1.67,1.02){$e_b(w,z_b)$}
\put(1.72,1.72){$\frac{e}{w}$}
\end{picture}
\end{center}

```

Now we can look at the output:



Adding graphics

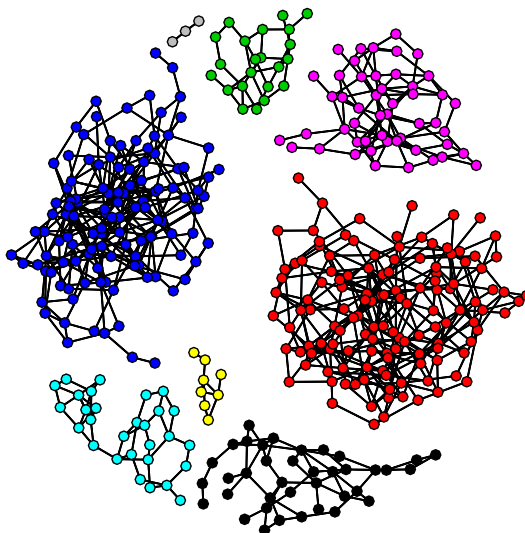
If we want to add externally created graphics to our \LaTeX document we will first need to add this statement to our header:

```
\usepackage{graphicx}
```

The code to actually add a picture is quite simple. We just need to specify the file path (which can be just the file name if we put this file in the same folder as the .tex file we are working on) and a scale factor which tells \LaTeX how big to render the picture:

```
\begin{center}
\includegraphics[scale=0.7]{socialstructureat500000.pdf}
\end{center}
```

Now we can take a look at the output:



Verbatim

The verbatim package allows us to include text exactly as is, and is useful for including Stata output or any code you use in your projects. First we need to add this code to our header:

```
\usepackage{verbatim}
```

The verbatim package is very easy to use as is illustrated in the code segment below:

```
\begin{verbatim}
regress Wage Education Experience Ability MotherEducation FatherEducation NumSibling

      Source | SS   df MS   Number of obs = 17919
-----+-----+----- F( 6, 17912) = 627.69
      Model | 132096.211  6 22016.0352 Prob > F = 0.0000
      Residual | 628256.763 17912 35.0746295 R-squared = 0.1737
-----+-----+----- Adj R-squared = 0.1735
      Total | 760352.974 17918 42.4351476 Root MSE = 5.9224

-----+-----
      Wage | Coef. Std. Err. t P>|t| [95% Conf. Interval]
-----+-----
      Education | .9631935 .027773 34.68 0.000 .9087557 1.017631
      Experience | .4731948 .0110963 42.64 0.000 .4514449 .4949446
      Ability | .765148 .0609231 12.56 0.000 .6457329 .8845631
      MotherEducation | .0233355 .0209318 1.11 0.265 -.0176929 .0643639
      FatherEducation | .0621293 .0165187 3.76 0.000 .029751 .0945076
\end{verbatim}
```

```

NumSibling | .0570295 .0221172 2.58 0.010 .0136776 .1003814
_cons | -5.979799 .4159054 -14.38 0.000 -6.795014 -5.164584

```

\end{verbatim}

This gives us the following output:

```
regress Wage Education Experience Ability MotherEducation FatherEducation NumSibling
```

Source	SS	df	MS			
Model	132096.211	6	22016.0352	Number of obs =	17919	
Residual	628256.763	17912	35.0746295	F(6, 17912) =	627.69	
				Prob > F =	0.0000	
				R-squared =	0.1737	
				Adj R-squared =	0.1735	
Total	760352.974	17918	42.4351476	Root MSE =	5.9224	

Wage	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
Education	.9631935	.027773	34.68	0.000	.9087557	1.017631
Experience	.4731948	.0110963	42.64	0.000	.4514449	.4949446
Ability	.765148	.0609231	12.56	0.000	.6457329	.8845631
MotherEducation	.0233355	.0209318	1.11	0.265	-.0176929	.0643639
FatherEducation	.0621293	.0165187	3.76	0.000	.029751	.0945076
NumSibling	.0570295	.0221172	2.58	0.010	.0136776	.1003814
_cons	-5.979799	.4159054	-14.38	0.000	-6.795014	-5.164584

Stata Output

If we want to go one step further, we can use the following Stata commands:

```

. findit outtex /* Brings you to a page where you can download the package */
. outtex, detail level below title(Dep = 'e(depvar)') /* creates the output */

```

You can find out more about this package here: <http://www.ats.ucla.edu/stat/stata/latex/estimates.htm>

If we run this command, we get the following output in our Stata console:

```

%----- Begin LaTeX code -----%
{
\def\sep{0.5em}
\def\fnsize{\footnotesize}
\def\onepc{${\ast}\ast}$} \def\fivepc{${\ast}\ast}$}
\def\tenpc{${\dag}$}
\def\legend{\multicolumn{3}{l}{\footnotesize{Significance levels
:\hspace{1em} $\dag$ : 10% \hspace{1em}
$\ast$ : 5% \hspace{1em} $\ast\ast$ : 1% \normalsize}}}
\begin{table}[htbp]\centering
\caption{Dep = 'e(depvar)'}
\label{tabresult regress}}
\begin{tabular}{l r @{} l }\hline\hline
\multicolumn{1}{c}
{\textbf{Variable}}
& \textbf{Coefficient} & \& \fnsize{(Std. Err.)} & \& \hline
Education & 0.094&\onepc & \& \fnsize{(0.002)} & \& \& \sep]
Experience & 0.037&\onepc & \& \fnsize{(0.001)} & \& \& \sep]
Intercept & 0.794&\onepc & \& \fnsize{(0.027)} & \& \& \sep]
\hline
\multicolumn{3}{c}{}}
\hline N & \multicolumn{2}{c}{17919}}

```

```

R^{2}$ & \multicolumn{2}{c}{0.158}\\
F $ _{(2,17916)}$ & \multicolumn{2}{c}{1686.584}\\
\hline
\end{tabular}
\end{table}
}

%----- End LaTeX code -----%

```

This gives us the Stata output we are used to seeing in journal articles:

Table 1: Dep = 'e(depvar)'

Variable	Coefficient (Std. Err.)
Education	0.094** (0.002)
Experience	0.037** (0.001)
Intercept	0.794** (0.027)
<hr/>	
N	17919
R ²	0.158
F _(2,17916)	1686.584

Errors!

There are two particularly common errors that everyone runs into while authoring \LaTeX documents. They can be very discouraging and frustrating if you do not know what to look for to fix them. The number one most important piece of advice I can give you for avoiding hard to fix errors is to compile your document frequently. This also has the added benefit of saving your work and can be done by clicking the **Quick Build** arrow button.

Not closing a parentheses, or having an extra bracket

This is perhaps the most common error people make and all of the code below will give you an error message:

```

\begin{equation}
\widehat{\beta}_{GLS} = \left[ \sum_n \left( \frac{1}{\sigma_n^2} x_{nx_n}' \right) \right]^{-1}
\left[ \sum_n \left( \frac{1}{\sigma_n^2} x_{ny_n} \right) \right]
\end{equation}

```

You will get an error that looks something like this:

```

! Paragraph ended before \mathaccentV was complete.
,to be read again>
\par
1.683

```

If we look closely, we can see that there is a missing `}` after the beta term at the beginning of the equation. the useful thing about this error message is that it tells you the line where the error occurred. To have the line numbers show up to the left of your input window, on the **texmaker** menu, then click on **Preferences**. Now that you are in the preferences pane, click on the **Editor** tab on the bottom left of the pane and you will see a check box on the top right of this pane that lets you choose to **Show Line Numbers**. Click this box and you will have a much easier time finding the source of error in your documents.

MS Word/PDF Symbols

One of the advantages of doing your homework on a computer is that you can often copy the question into your document so it is always there for you to look at. This works really well most of the time except when it doesn't. Then you get an error. The most common error causing symbol in text that you might copy into a \LaTeX document is the apostrophe. Fortunately this is easily remedied by retyping the apostrophe. You will get an error that looks like this:

```
! Package inputenc Error: Keyboard character used is undefined
```

Unfortunately I cannot even show you what this looks like because my document will not compile but I can offer the suggestion that if you see a word or symbol that you know is spelled correctly but still has red underlining in the editor, it is probably not recognized by \LaTeX . All you have to do is replace it with the same symbol but typed by you and this should resolve the error.

Resources

Here is a list of websites that are very helpful resources:

1. The \LaTeX wikibook is **THE** best place to go for information on most normal things you might want to do with a document: <http://en.wikibooks.org/wiki/LaTeX>
2. TikZ is an advanced graphics package for those of you who are interested in making graphical models, game trees or any really fancy graphics in \LaTeX : <http://www.texample.net/tikz/resources/>
3. The Tex Users Group (TUG) is a very active message boards site with lots of posts related to doing any crazy thing you could want: <http://tug.org/>
4. BibTeX is perhaps the most useful part of \LaTeX I will not have time to cover in this workshop. It allows for automatic bibliography management, super easy citations and automatic citation generation. It is perhaps the most timesaving part of \LaTeX and you can read more about it here: <http://www.bibtex.org/>
5. I also highly recommend Mendeley Desktop as a great free bibliography management software that integrates really well with \LaTeX and is available here: <http://www.mendeley.com/>
6. This presentation is a great introduction to the Beamer Package which lets you make beautiful presentations in \LaTeX : <http://users.wfu.edu/wickersg/latex/beamer/beamer.pdf>
7. This website has templates and instructions for creating conference quality posters using Beamer and \LaTeX : <http://www.brian-amberg.de/uni/poster/>

Creating Your Own Class

All of the stuff we have been putting in the preamble of our document can easily get to be a bit cumbersome. Fortunately there is a really slick way of dealing with this. We can create our own document class as a separate file and keep all of our package declarations in this file. Then we can access it simply by changing one line in our preamble.

```
\documentclass{article} % this is what we have been using
\documentclass{ISSRLaTeXWorkshop} % this is what we will change it to
```

Once we have done this, we can get rid of all of those declarations in our preamble and have a nice simple beginning to every document. Before we can do this, we have to create our .cls file and save it in the same directory as our .TeX files. Lets take a look at the code below:

```
\ProvidesClass{ISSRLaTeXWorkshop}[10/16/2013 A custom .cls file for writing academic papers
authored by Matthew Denny]
\NeedsTeXFormat{LaTeX2e} % Tells the computer to use LaTeX2e formatting
\LoadClass{article} % this class is a modified version of the standard Article class

%BELOW IS A SAMPLE DOCUMENT WITH PREAMBLE
%\documentclass[fleqn]{MJD} %this should be whatever you call your class
%\title{Example Document}
%\author{Matthew Denny}
%\date{\today}
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%\begin{document}
%\maketitle
%\end{document}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% These packages give you access to a whole bunch of different fonts and alot of
% functionality, if you need to add a package that is required for doing something you want to do,
% put it below

%The etx package need to be the first loaded
\usepackage{etex} % this let use lots of packages

\usepackage[english]{babel} % this gives us access to special characters
\usepackage{amsmath,amssymb,amsfonts,amsthm,textcomp} %math characters
\usepackage{mathabx}
\usepackage{multirow}
\usepackage{verbatim} % for including code
\usepackage{tabularx} % for creating tables with some auto formatting
\usepackage{hhline} %for drawing lines in tables
\usepackage[normalem]{ulem} % lets us do some neat underlining
\usepackage{booktabs} % for lines in tables
\usepackage{array} % lets us create arrays
\usepackage{datetime} % lets us timestamp a docuemnt with the exact time it was compiled
%useful for version control and can be done by including the following line in your document:
% \noindent Time Created: \ampmtime
\usepackage{charter} % uses the charter font
\usepackage{float} % helps LaTeX better position figures

%allows us to remove page numbering
%\usepackage{nopageno}

%makes our documents double spaced
% \doublespacing

% Color package and define colors
\usepackage{color}
%This is another really nice color package with over 60 predefined colors. Check it out in
% the \LaTeX wikibook section on colors
\usepackage[usenames,dvipsnames]{xcolor}

% Hyper Reference Formatting
\usepackage{hyperref}
\hypersetup{colorlinks,breaklinks,
            linkcolor=black,urlcolor=blue,
            anchorcolor=blue,citecolor=black}

% Allows me to import pictures
\usepackage{graphicx}
% Sample code to include a picture (make sure you include the full filepath if the picture is
% not in the same directory as the .TeX file you are working on)
% \includegraphics[scale=0.5]{examplepicture.pdf}\

%page layout
\setlength\voffset{-1in}
\setlength\hoffset{-1in}
\setlength\topmargin{1in}
\setlength\oddsidemargin{1in}

```

```
\setlength\textheight{9in}  
\setlength\textwidth{6.5in}  
\setlength\footskip{0.5cm}  
\setlength\headheight{0cm}  
\setlength\headsep{0cm}
```

Now all we have to do is open a new document, paste this code in, and save our new class file as **ISSRLaTeXWorkshop.cls** in the same folder as all the rest of our .tex files. As you get more comfortable with using \LaTeX you should modify this .cls file, make it your own and even change its name to something easier to remember (just make sure to also change its name in your documentclass statement so you are still calling the right file). You are now ready to embark on what will be a challenging, but immensely rewarding few months of getting comfortable using \LaTeX . You will be rewarded for experimenting and for practicing Googling for solutions to the problems you encounter. There is an answer out there to pretty much any question you could have, you just have to find it. I hope someday you all will end up teaching a \LaTeX workshop. If you do, feel free to email me and I will send you the .tex code for this document.