Typesetting in 90 Minutes

Andrew Caird

<2013-07-09 Tue>

On Wednesday, August 7, 2013 I'm going to give a quick introduction to LaTeX. I'll be in the Johnson Rooms in the Lurie Building from 12:00p–1:30pm.

I'll be updating this blog entry in the coming weeks; please check back for updates, instructions, and notes.

LaTeX in 90 minutes

In 90 minutes in the Johnson Rooms we'll go over how to install LaTeX on computers running Microsoft Windows, Apple OS X, and Ubuntu Linux, then how to write a simple document and produce a PDF file, how to add a title, author, table of contents, equations, and graphics. The last third of the time will be spent answering specific questions about LaTeX.

If you have experience with LaTeX, the early part of the talk isn't likely to be useful to you, but the later parts might be.

You should bring a laptop, as this will be a hands-on discussion.

The Agenda for the Class

12:00p–12:30p sorting out the installation of LaTeX and its supporting programs on your laptops, and getting it installed if you haven't done so already.

12:30p–1:00p writing a first simple document and producing a PDF file, then augmenting the document with a title, author, equations, figures, tables and tables of contents, figures, and tables.

1:00p–1:30p answering particular questions about LaTeX and your use of it

Preparation for the Class

Before coming to class, please install LaTeX on your laptop[1] and download a copy of *The Not So Short Introduction to LaTeX* and bring an electronic or printed copy.

[1] This is also described in Appendix A of *The Not So Short Introduction to LATEX*

Installing LaTeX on Microsoft Windows

The LaTeX installation for Windows is called MikTeX and can be downloaded from `http://www.miktex.org/about`.

The first few FAQ entries at `http://docs.miktex.org/faq/faq.html` may be helpful in getting MikTeX installed.

Installing LaTeX on Apple OS X

The LaTeX installation for Mac OS X is called MacTeX and can be downloaded from `http://tug.org/mactex/`.

That same web page (`http://tug.org/mactex/`) links to other helpful documents about maintaining the MacTeX installation.

Installing LaTeX on Ubuntu Linux

LaTeX can be installed on Ubuntu through the software center or via the command line by typing `sudo apt-get install texlive` at a shell prompt.

Writing a Simple Document

To test your LaTeX installation, you should create a simple document[2].

Contents of a simple LaTeX document

The contents of a very simple LaTeX document are:

```
\documentclass{article}
\begin{document}
This is my document.  There isn't much to it.

This is the second paragraph of my document.  This
paragraph is longer than the first paragraph because
I kept typing words in this paragraph, and didn't
type as many words in the preceding paragraph.

This is my third paragraph, the shortest yet.

\end{document}
```

The first line, `\documentclass{article}`, is the most common type of document you'll write in LaTeX—the other common options are `letter` for letter that you would mail to someone and `book`, which adds the option of chapters to your document. For anything beyond `article` you will probably want to consider packages other than `letter` and =book=[3].

The second and last lines, `\begin{document}` and `\end{document}` bound your document—you always have to have these two lines and nearly all of your content will be between those two lines.

[2] See Chapter 1 of *The Not So Short Introduction to LaTeX* (*TNSSItL*) for more detail

[3] See sections 1.3–1.4 of *TNSSItL*

Between the `\begin` and `\end` lines is the content for your document. In this example it is three paragraphs. A paragraph is created by leaving a blank line. You don't have to wrap the lines yourself, they will be processed by LaTeX when you process your document.

Typing the document

A LaTeX document is simply a text file—this is one of the beauties of LaTeX: you'll always be able to read the text file.

If you are familiar with a text editor for programming, such as `emacs` or `vi`, you can use that editor to write LaTeX documents. If you prefer a more integrated environment for typing your documents, MacTeX comes with TeXworks and TeXshop, MikTeX comes with TeXworks, and on Ubuntu there are TeXmaker, Kile, Gummi, and many others.

After opening your editor, you should be able to paste in the sample document above, or start writing your own short document between the `\begin{document}` and `\end{document}` lines.

Save the file as `sample.tex`.

Processing the document

When you are done writing, you can process the document to a PDF file and see what you've created[4].

If you have a `.tex` file, you can process it from the command line in the Windows `command` shell, the Mac `Terminal`, or a Linux `xterm` by typing:

```
pdflatex sample.tex
```

and you should see several lines of output, the first few and last few should be something like:

```
This is pdfTeX, Version 3.1415926-2.4-1.40.13 (TeX Live 2012)
 restricted \write18 enabled.
entering extended mode
(./sample.tex
   [... many more lines ...]
LaTeX2e <2011/06/27>
Output written on sample.pdf (1 page, 16198 bytes).
Transcript written on sample.log.
```

The LaTeX editors will have a menu item or button that will do the processing for you.

If there is an error, it isn't always easy to decode, but some of the editors will try to set the cursor at the point of the error. Otherwise,

[4] See section 1.5 of *TNSSItL*, with the exception that we will always use `pdflatex` instead of the `latex`, `xdvi`, and `dvips` commands in *TNSSItL*; for more on `pdflatex`, see Section 4.7 of *TNSSItL*

carefully read the error and check your `.tex` file for missing brackets, a `\begin` without its matching `\end`, and other typos.

Once you are able to successfully process `sample.tex` into `sample.pdf` and open `sample.pdf` and see it, we can start adding to the document.

Adding to the document

Once we have a simple document, we can start adding more interesting parts to it.

Title and Author

Adding a title and author or authors to your document is straightforward[5].

Between the `\documentclass` and `\begin{document}` lines, you should add:

```
\title{My Sample Document}
\author{Type Your Name Here}
```

and immediately after the `\begin{document}` line add the line:

```
\maketitle
```

After processing the updated `.tex` file, you will see a title, author, and today's date added to the top of your PDF file.

Sections and subsections

LaTeX supports numbering sections, subsections, and sub-subsections by simply adding a line that reads `\section{My Section Title}` (or `\subsection{My subsection Title}` or `\subsubsection{My subsubsection Title}`) before the section[6].

Editing `sample.tex` and adding `\section` before the first and third paragraphs and `\subsection` before the second paragraph brings us to a source document that looks like:

```
\documentclass{article}
\title{My Sample Document}
\author{Type Your Name Here}
\begin{document}
\maketitle

\section{My first section}
This is my document.  There isn't much to it.
```

[5] This is also described briefly in Section 1.5 of *TNSSItL*

[6] See Section 2.7 of *TNSSItL*

```
\subsection{A subsection for fun}
This is the second paragraph of my document.  This
paragraph is longer than the first paragraph because
I kept typing words in this paragraph, and didn’t
type as many words in the preceding paragraph.

\section{My very short third section}
This is my third paragraph, the shortest yet.

\end{document}
```

Processing that file, assuming I didn’t make any errors and you didn’t introduce any in its transcription, should result in a document with a title, author, date, and two sections, one with a subsection.

Equations and Tables

*Equations There are two types of basic equations in LaTeX: equations that are part of the text, in-line math, and equations that are offset from the text and numbered[7].

Equations are written in text and the keyboard math symbols. Pythagoras might have written:

```
\begin{equation}
a^2 + b^2 = c^2
\end{equation}
```

to get

a$^2 + b^2 = c^2$ (1)

Another example is the equation for the area of a circle:

```
A = \pi r^2
```

to get

A = πr^2 (2)

*Tables Tables in LaTeX comprise five items that describe a table:

- a `\begin{table}` and matching `\end{table}` surrounding the tabular environment[8]
- a `\begin{tabular}{<tableadvice>}` and matching `\end{tabular}` around the table data
- the `<tableadvice>` which describes the alignment and vertical lines in the table using `|` to describe vertical lines and letters (`l`,=c=,=r=) to describe alignment

[7] See Chapter 3 of *TNSSItL* for much more detail

[8] See Sections 2.1.2 and 2.11.6 of *TNSSItL*

- the `&` character to define columns `\\` to define rows, and `\hline` to draw horizontal lines
- the caption for the table, using `\caption{My Caption}` after the `tabular` environment and before the end of the `table` environment.

Taking the **Degrees Granted for Academic Year 2011-12** table from `http://www.engin.umich.edu/college/about/facts` we could represent that table in LaTeX as:

```
\begin{table}
\begin{tabular}{l|r|r|r}
Degrees Granted for Academic Year 2011-12 & Bachelors & Masters & Doctoral \\ \hline
Degrees Granted                           &    1,348  &  1,093  &    258   \\
\% Women                                  &      21\% &    21\% &    21\%  \\
\% URM                                    &       7\% &     8\% &    14\%  \\ \hline
\end{tabular}
\caption{Engineering Degrees Granted for Academic Year 2011-12}
\end{table}
```

- The `|` characters in the `tabular` line instruct LaTeX to put vertical lines between the columns, but not on the left or right ends.
- The `\hline` commands at the ends of the lines instruct LaTeX to put a horizontal line below the current line.
- The `\%` is required because `%` is a special character in LaTeX that denotes a comment from it to the end of the line; that's very useful in many cases, but not when you mean percentage.

This all produces a table that looks like:

Degrees Granted for Academic Year 2011-12	Bachelors	Masters	Doctoral
Degrees Granted	1,348	1,093	258
% Women	21%	21%	21%
% URM	7%	8%	14%

Figures from external files

While it is possible to create images in LaTeX directly using it's native but limited `picture` environment or the somewhat complicated TikZ package[9], it is most common to create images in another package — gnuplot or R for plots, Adobe Illustrator or another drawing package for diagrams.

For this example, we'll make a plot using the statistical package R and write the plot to a file called `mpg-weight.png`. The R code for this is:

[9] See Chapter 5 of *TNSSItL* for more on the `picture` environment and TikZ

```
library(ggplot2)
mtcars$gear <- factor(mtcars$gear,levels=c(3,4,5), labels=c("3gears","4gears","5gears"))
mtcars$am   <- factor(mtcars$am,levels=c(0,1), labels=c("Automatic","Manual"))
mtcars$cyl  <- factor(mtcars$cyl,levels=c(4,6,8), labels=c("4cyl","6cyl","8cyl"))

qplot(wt, mpg, data=mtcars, geom=c("point", "smooth"),
     method="lm", formula=y~x, color=cyl,
     main="Regression of MPG on Weight",
     xlab="Weight", ylab="Miles per Gallon")
```

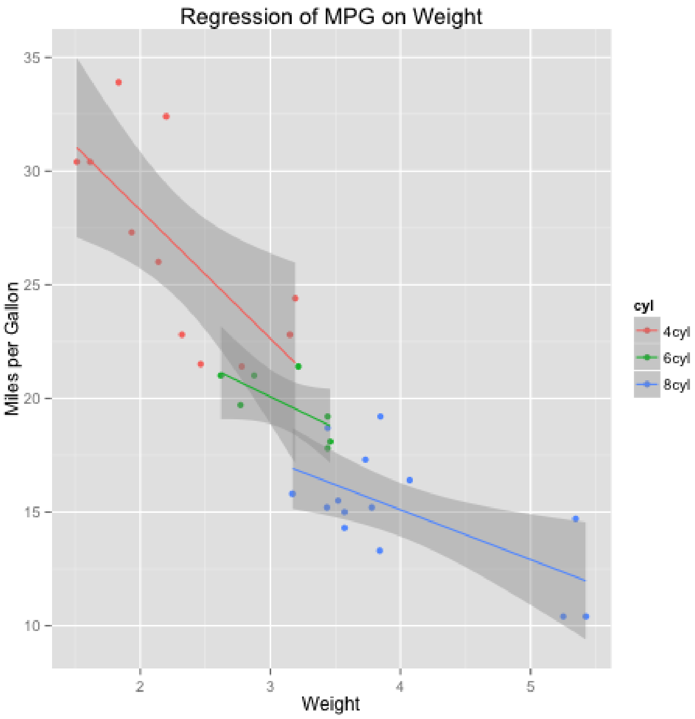

Figure 1: Weight vs. MPG

This file is included in the LaTeX output by adding the line:

```
\usepackage{graphicx}
```

immediately after the `\documentclass{article}` line in your `.tex` file and then including the lines[10]:

```
\begin{figure}
\includegraphics[width=.9\linewidth]{mpg-weight.png}
\caption{Weight vs. MPG}
\end{figure}
```

The full path to `mpg-weight.png` should be specified in the `\includegraphics` line; in this example LaTeX will assume that the image file is in the same directory as the `.tex` file.

[10] See Section 4.1 of *TNSSItL*

You can download this image for using when processing `sample.tex` by right-clicking on it and saving it to the same folder as `sample.tex`.

The `graphicx` package combined with `pdflatex` can read PDF, PNG, JPEG, and MetaPost graphic formats. Other formats should be converted to one of these for inclusion. If you can export directly to PDF, that is ideal.

Tables of Contents, Tables, and Figures

Adding tables of contents, tables, and figures to the front matter of your document is straight-forward.

To add a table of contents, add the line `\tableofcontents` after your `\maketitle` line to generate a table of contents. Because LaTeX is a single-pass processor, it stores some of its information in auxiliary files; this means that you may have to run `pdflatex` more than once to get all of the references resolved[11].

To add a list of figures, insert the line `\listoffigures` after the `\maketitle` or `\tableofcontents` lines in your `.tex` file.

To include a listing of tables at the front of your document, there is the `\listoftables` command that is analogous to the `\listoffigures` command.

Our current `sample.tex` file now looks like:

```
\documentclass{article}
\usepackage{graphicx}
\title{My Sample Document}
\author{Type Your Name Here}
\begin{document}
\maketitle
\tableofcontents
\listoftables
\listoffigures
\section{My first section}
This is my document.  There isn’t much to it.

\subsection{A subsection for fun}
This is the second paragraph of my document.  This
paragraph is longer than the first paragraph because
I kept typing words in this paragraph, and didn’t
type as many words in the preceding paragraph.

\section{My very short third section}
This is my third paragraph, the shortest yet.
```

[11] See Sections 2.7 and 2.12 of *TNSSItL*

```
However, this section has a table and a figure.

 \begin{table}
 \begin{tabular}{l|r|r|r}
 Degrees Granted for Academic Year 2011-12 & Bachelors & Masters & Doctoral \\ \hline
 Degrees Granted                           &    1,348  &  1,093  &    258   \\
 \% Women                                  &      21\% &    21\% &    21\%  \\
 \% URM                                    &       7\% &     8\% &    14\%  \\ \hline
 \end{tabular}
 \caption{Engineering Degrees Granted for Academic Year 2011-12}
 \end{table}

\begin{figure}
\includegraphics[width=.9\linewidth]{mpg-weight.png}
\caption{Weight vs. MPG}
\end{figure}

\end{document}
```

This file demonstrates sections, tables, figures, and front-matter.

Lists, Bibliographies, and everything else

Lists

Lists in LaTeX are environments similar to other environments we've seen, like `\begin{figure}` ... `\end{figure}`; the list environment for a bulleted list is `itemize` and for a numbered list is =enumerate=[12].

Lists can be embedded for sub-items. Example LaTeX that shows this is:

```
\begin{enumerate}
  \item The first item in my list is this sentence; it's a
        pretty long sentences that has several properties:
  \begin{itemize}
    \item it has 18 words
    \item it has a semi-colon
    \item it has one apostrophed word
  \end{itemize}
  \item This is the second item in my list
  \item This is the end of my incredibly boring list
\end{enumerate}
```

1. The first item in my list is this sentence; it's a pretty long sentences that has several properties:

    - it has 18 words

[12] See Section 2.11.1 of *TNSSItL*

  - it has a semi-colon
  - it has one apostrophed word

2. This is the second item in my list
3. This is the end of my incredibly boring list

Bibliographies

LaTeX can support simple bibliographies within a document using the `\cite` command in the document for citations and the `\bibitem` command in the `thebibliography` environment[13].

For larger projects or areas of study, using BibTeX to maintain a bibliographic database and integrate with the `\cite` commands in several documents.

Debugging LaTeX documents

Writing LaTeX documents is similar to writing a computer program, with all of the debugging issues that go with computer programming.

The error messages produced by LaTeX can be difficult to decode; my preferred method of debugging is to localize the error without worrying too much about the error message by processing the document often, and especially after making significant additions.

If processing the document does result in errors that I cannot understand from the error, I then start by using the `%` to comment out parts of the LaTeX file and processing it until the error is gone, and looking at the commented section for errors.

Other resources

LaTeX has been around for decades and has a lot of documentation and support.

Websites

The best web site for LateX support is `Google`, followed by StackOverflow. LaTeX-specific web sites are the website for the LaTeX Project, the TeX Users Group, and the LaTeX WikiBooks site. For additional packages for LaTeX, the Comprehensive TeX Archive Network is the definitive source; the umthesis class is available here, for example[14].

Books

There are 82 books in the search results for `latex typesetting` at Amazon. The canonical books, in my opinion, are:

- *LaTeX: A Document Preparation System (2nd Edition)* by Leslie Lamport

[13] See Section 4.2 of *TNSSItL*

[14] For an alternative, `emacs` based, input format, take a look at Org-Mode; it can export HTML and LaTeX

- *The LaTeX Graphics Companion* by Gossens, Mittelbach, Rahtz, and Voss
- *The LaTeX Companion (2nd edition)* by Mittelbach, Gossens, Braams, Carlisle, and Rowley

there are also a lot of other excellent books, many of which are available in the U-M libraries.

Resources at the University of Michigan

Emailing caen@umich.edu will come to me and others at CAEN and we'll do our best to help you.