The `R` Reading Group's Introduction to LaTeX

Isabell Hubert

July 13, 2015

1 LaTeX is...

- pronounced /ˈlɑːtɛx/ or /ˈleɪtɛx/. At least this was how it was intended - Donald Knuth, the creator of TeX, said in the first chapter of his TeXbook: "When you say it correctly to your computer, the terminal may become slightly moist." Apparently /ˈlɑːtɛk/ and /ˈleɪtɛk/ are also acceptable, what with the extended use of LaTeX in the English-speaking world. Just don't make it /ˈleɪtɛks/ (the rubbery stuff. This can be confusing).

- the most widely used format of TeX, which is the underlying language. LaTeX adds a couple of very useful functions, like sections and their numbering, referencing, and document classes.

- **not** like Word, Pages, etc. You provide the syntactic markup, and TeX takes care of the rest, namely layouting and design. This makes for a uniform, beautiful design that works according to professional typesetting standards (and doesn't have us as non-designers/non-typography experts meddle with design or typography). If you're into web design, think about it this way: You provide the HTML syntax, TeXdoes the CSS'ing.

LaTeXs default output is usually pdf, but you can specify other print formats too. I have only ever used pdf though.

1.1 TeX is especially useful...

- for mathematics and formulas, or other special characters (like IPA in our case - you won't get the weird font issues like in Word, where everything is one font and then IPA characters magically are not, as the font you wanted does not support this range of special characters.)

- for cross-referencing, footnotes, tables, figures (you can get more elaborate layouts, e.g. for the distribution of figures, than Word would allow for).

- as it actually gets numbering right... for some reason Word was never able to do this properly for me, especially for nested lists.

- for multi-column layouts.

- for when journals etc. provide you with templates - they are less likely to muck up than e.g. Word templates, as you cannot accidentally or otherwise interfere with the layouting (you can, if you really want to, but then we'd assume you know what you're doing anyways).

- as it can be integrated with `R` so tables and figures are continuously updated, which is where `knittr` comes in (this is the part I do not know how to do yet, but I hear it's awesome!)

2 How to Install LaTeX

For you to use LaTeX comfortably, you'll need two things: First, a *TeX installation*, and second, a *LaTeX editor*.

For your *TeX installation*, you can get LaTeX on its own, or even just TeX, which is the underlying language; most people opt for a LaTeX *distribution*, which is more user-friendly in installation and usage, includes a package manager, and is regularly updated. The two most popular ones for Windows are TeXlive and MikTeX; I like and use MikTeX, as TeXlive somehow did not work on my computer. It does work for most other people though, so give it a try if you like. You can simply install either of them using the installer from the corresponding website. The TeX installation can theoretically be used from the shell or the terminal, but this is not the best user experience, especially if you are just getting started with LaTeX. This is where the editor comes in. This will be the "frontend" for you to comfortably interact with the TeX install through a user interface. Think of the TeX installation as `R`, and the editor as RStudio. That is basically what is happening here.

If you require/prefer a different LaTeX distribution, e.g. for a different operating system or as portable software, look here.

Then, as a second part, you'll need an *editor*. You can make your choice as to which editor you would like to use among the many that are available for various operating systems. Make sure to check that the editor you want works with the TeX installation you opted for. I use TeXmaker, but you can even use Notepad++ or Sublime, which you might already have installed. TeXworks is another popular editor, but you may find that an entirely different editor might fit your needs and preferences better. If you opt for a less common one, be aware though that there won't be that many discussions on e.g. StackExchange to help you out in times of need. Most editors behave very similarly though.

This seems like a lot of work (and it is - especially compared to installing Word or running pre-installed Pages), but you'll only have to do this once, and it'll be worth it!

3 Helpful LaTeX Resources

Now that we're done installing LaTeX, I'd like to introduce its potentially best resource: the Wikibook, without which I probably would not have been able to use LaTeX for basically everything. Use it. Seriously. Especially at the beginning, it will answer all your questions.

The potentially second best resource is the *Not So Short Introduction to* LaTeX. I found this one great for reading, just as a general introduction. It is also great to look things up though.

The third majorly helpful source is the TeX section on StackExchange, which is fantastic for specific questions about LaTeX as such, its packages, or how to accomplish specific things. Chances are, you are not the first one to have the question you are having!

I used those three resources extensively the first few times I typeset a document in LaTeX (by "extensively," I mean that I used it about 30 times an hour.) You are always learning something new, and you'll feel like there's no end to it, but very soon you'll notice a sharp decline in how often you have to look stuff up. You'll begin to notice this after you have created your first one to three documents. I still look up stuff from time to time, as one of course gets bolder and wants to try new stuff, but at that point LaTeX isn't even that scary anymore anyways. :)

4 Any basic LaTeX document...

looks like this, or a variant of this - just copy this code into an empty document to get started:

```
\documentclass{article}
\title{The Title of My Fancy Document}
\author{Gnome Chomsky}
\date{July 2015}
```

```
\begin{document}
   \maketitle
   Hello world! This is where I can put my body text.
\end{document}
```

If this reminds you of:

```
<!DOCTYPE html>
<html>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

this is because TEX and HTML are both markup languages! They require a basic structure to work, which is the one we have just seen. Now that we have this very basic structure in place, we can render our first document. Click the button (or use the shortcut - usually something like F1 or F2) in your editor to produce some sort of output, ideally pdf.

Our document looks pretty basic and bland. Many other LATEX documents you see are a lot more elaborate, which is (partly) the work of document classes: There's a number of standardized ones; pick the one closest to what you want to achieve. For most papers, `article` is probably fine. Each of those classes have options, like page format, columns, etc. The standard set of options works for all of the document classes; some are restricted to only a couple of classes. You can refer to the Wikibook to see what's possible. If you have to change the document class later, that's also not the end of the world, and very easy to achieve!

A basic document class with a couple of options looks like this:

```
\documentclass[11pt,letterpaper,twoside]{article}
```

We can incorporate this into our basic document to make it less basic:

```
\documentclass[11pt,letterpaper,twoside]{article}
\title{The Title of My Fancy Document}
\author{Gnome Chomsky}
\date{July 2015}
\begin{document}
   \maketitle
   Hello world! This is where I can put my body text.
\end{document}
```

To accomplish things that are not covered by those standard classes and options, it is very likely that there is already a package in existence that you can use to accomplish precisely what you want. Google and StackExchange can help you find them!

When you use a package name for the first time, your distro will automatically install the package. (You won't have to go through a separate installation and loading/library sequence like the one in `R`. Also, the `/usepackage` command simply stays in the document, so you won't have to add it again and again whenever you start up TEX.) Like in RStudio, you'll get an error if something goes wrong with package installation (I don't think I have ever gotten a package error, apart from the times where I mistyped the package name...). Installing and using a package works like this:

```
\usepackage[option1,option2,option3]{package_name}
```

We can include this in our document (put packages *before* the `/maketitle` etc., but *after* the class declaration):

```
\documentclass[11pt,letterpaper,twoside]{article}
\usepackage{tipa}

\title{The Title of My Fancy Document}
\author{Gnome Chomsky}
\date{July 2015}
\begin{document}
   \maketitle
   Hello world! This is where I can put my body text.
\end{document}
```

Congratulations! You have just downloaded and installed the text IPA package, with which you can type elaborate IPA. Now for the “meat” of the document, the actual content. It is typically sectioned, like this:

```
\section{Introduction} I researched this because...
\subsection{Literature Review} Look how much I read!!
```

The stuff in curly brackets will be output as numbered headings. Check the document structure Wikibook page to see which sections are supported by which document class.

There we go! You just created your first LaTeX document. My advice: Experiment with everything (apart from layouting, see the word of caution below; it takes exceptional will and determination to change this though. You cannot do this accidentally.) You'll learn the most when you actually sit down and want to use LaTeX to accomplish something specific. You'll gradually find out how everything works, especially with the help of the resources listed above, and by asking others who use LaTeX too.

5 Some More Helpful Stuff to get you going:

- To make a new paragraph within a section, just insert a blank line between two blocks of text.
- Look up LaTeX bibliography management if you are planning to use a bibliography and referencing in your documents. There is a number of bibliography managers available, like `BibTeX, BibLaTeX, and NatBib`; they all have pros and cons, so settling for one of them is mostly a personal choice.
- Comments begin with % and affect one line.
- This leads us to a great example of package use: Maybe you want to comment out a large amount of lines, and you're tired of putting a % at the beginning of every line. Texmaker provides a keyboard shortcut for this, where you can just highlight the block of text you want to comment out and hit CTRL+SHIFT+t, but maybe you are using a different editor that doesn't offer this option. You probably guessed it - there's a package for it! you can `/usepackage{comment}`, and then just `/comment[everything you don't want to appear on the page, like this bit of text.]`

6 A Word of Caution

Do not try and change LaTeX layouting. TeX is extremely good at layouting on its own; this is where and why it shines compared to WYSIWYG word editors. Specifying/changing layouting manually is extremely complicated and does not follow regular LaTeX syntax. Mostly, if something doesn't look the way you want it to look, changes can be accomplished through (in order of what you should try first) (a) a better (= more meaningful) syntax, (b) changing a document class property, or (c) using a package. Change layouting ***only if you know what you are doing***, as it overrides the internal layouting that TEX does for you, and can hence severely interfere with the rendering and it can definitely mess everything up.

7 That is all.

Now go out and create things! Don't forget to make extensive use of the resources. They don't cost you anything. Or, just ask someone who is using LATEX as well!

Good Luck!!