\LaTeX: The Bare Minimum

Paul E. Johnson\textsuperscript{1, 2}

\textsuperscript{1}Department of Political Science

\textsuperscript{2}Center for Research Methods and Data Analysis, University of Kansas

2016
Outline

1. Why?

2. Structure of a \LaTeX\ Document
Outline

1. Why?

2. Structure of a \LaTeX\ Document
MS Word is Finger Painting

- It is easier to write a letter to Mom with Word, more difficult to produce a systematic, uniformly formatted documents.
- Too easy to accidentally “reformat” particular pieces in inconsistent ways.
- Pasting imports inconsistent, hidden style & structure
- Equations are tedious, unpredictable when exported
- Non-reproducible documents.
We’ve Lost the Separation of Content and Structure

- Word, and Word Perfect, were not always so GUI.
  - Text was created and marked by its style, and it stayed that way
  - Reformatting was done by revising the style sheet
    * Example: if you want to change all italicized words to bold italics, change the style, not the document
- The separation of “content” from “format” was possible, as late as 1992 (or so).
The \TeX\ Idea

- Donald Knuth, Stanford professor, developed \TeX
- Stated objective: let authors focus on the content of their words and equations
- Publisher standards for margins, indentation, table placement, etc, were wrapped up in “Style” or “Class” packages.
- \LaTeX\ is the newer edition of \TeX\ (both name files *.tex)
Brief Historical Detour

Original workflow was \LaTeX \rightarrow \text{DVI} \rightarrow \text{Postscript}

1. DVI: “device independent format”
2. A program named “latex” converted “*.tex” \rightarrow “*.dvi”
3. One would view the dvi, much the same as one views PS or PDF today
4. Follow up programs convert: DVI \rightarrow \text{PS}

- Today, English speakers more likely use “pdflatex” \LaTeX \rightarrow \text{PDF}
- New development focus in on incorporation of international character sets (Xe\TeX and Lua\TeX)
Why?

Different from MS Word?

- The *.tex document is plain text (has no hidden fields, markup)
- Blank lines separate paragraphs, etc
- Can edit with any “flat text editor” program (Emacs, TexMaker or TexWorks or TexShop or TexStudio . . .)
- Does not “absorb” graphics to make on giant *.tex file. Rather, the *.tex file refers to other files.
- Authors “compile” the document into PDF or HTML or ...
- Word can be used more systematically, but most users never bother to learn how
Most \LaTeX\ Projects Begin with a Template

- As we will see, a \LaTeX\ document has some “boilerplate” that is required.
- Because that boilerplate is difficult to produce and use, most users find example documents that meet their requirements and they revise from there.
- The KU Dissertation Thesis Template project is in that vein, providing a working document.
A \LaTeX{} Distribution

- Distribution is a big\textit{ish} collection of programs and format files
- Consider "Mik\TeX", a large, free distribution of \LaTeX{} software for MS Windows

- Look under Mik\TeX{}’s install, eg “C:\Program Files(x86)\Mik\TeX”
- Folder Mik\TeX/miktex/bin: executables (exe files)
  - Processors: latex, pdftex, dvips, xetex, tex4ht, oolatex
  - Viewers: yap (for dvi and ps)
  - Editors: \TeX{}works

- Folder “tex” is collection of packages.
- Look under tex/latex, one folder per addon package
Extensible: The Good and the Bad

- CTAN: Comprehensive \TeX{} Archive Network: 1000s of packages
  - There is no “corporate regulation” of the \TeX{} “thing”. New compilers, packages, scripts, pop up all the time.

- The Good:
  - Packages for many specific purposes
  - Scholars/Universities/companies/journals can create customized document styles
  - Example: Beamer \TeX{} framework (this document)

- The Bad:
  - Some packages don’t work, are not up-to-date
  - User documents are “broken” by slapping together contradictory packages.
  - Some obvious features unavailable (nobody pays package developer).
Outline

1 Why?

2 Structure of a \LaTeX{} Document
Simplest Possible LaTeX Document

\documentclass{article}
\begin{document}
\maketitle
Here's the smallest LaTeX document I can provide.

Type any crap you want here. Use blank lines to separate paragraphs.
Test some math $\pi R^2$
\end{document}
Macros, Environments, etc

- Comments prefixed by %
- A \LaTeX{} Macro: backslash-argument{content}: \texttt{\author{Paul Johnson}}
- An environment is text bracketed by “begin” and “end” statements

\begin{frame}
\frametitle{Macros, Environments, etc}
\begin{itemize}
\item Comments prefixed by \%
\item A \LaTeX{} Macro: backslash-argument{content}: \texttt{\author{Paul Johnson}}
\item An environment is text bracketed by ‘‘begin’’ and ‘‘end’’ statements
\end{itemize}
\end{frame}
Save That, Compile it

- In the terminal, the user runs “pdflatex example.tex”
  - latex \rightarrow pdf
- Looks like this, if you can see the input & output.

```bash
$ pdflatex example.tex
This is pdfTeX, Version 3.14159265–2.6–1.40.15 (TeX Live 2014/Debian) (preloaded format=pdflatex)
  restricted \write18 enabled.
  entering extended mode
(./example.tex
LaTeX2e <2014/05/01>
Babel <3.9k> and hyphenation patterns for 4 languages loaded.
(/usr/share/texlive/texmf-dist/tex/latex/base/article.cls
Document Class: article 2007/10/19 v1.4h Standard LaTeX document class
(/usr/share/texlive/texmf-dist/tex/latex/base/size10.clo))
No file example.aux.
```
Save That, Compile it ...

```
\texttt{example/texmf-dist/fonts/type1/public/amsfonts/cm/cmmi7.pfb}<\texttt{/usr/share/texlive/}
\texttt{texmf-dist/fonts/type1/public/amsfonts/cm/cmr10.pfb}<\texttt{/usr/share/texlive/texmf-d}
\texttt{ist/fonts/type1/public/amsfonts/cm/cmr12.pfb}<\texttt{/usr/share/texlive/texmf-dist/fonts/type}
\texttt{1/public/amsfonts/cm/cmr7.pfb>}
```

Output written on example.pdf (1 page, 60271 bytes).
Transcript written on example.log.

- Running pdflatex produces several intermediate files:

- A more complicated document may require repeated runs of “pdflatex” and “bibtex” to make all of the separate pieces work together.

- To avoid manually running those separate bits, many people use a convenience scripts like “texi2pdf”
Editors to Facilitate LaTeX Work

- TexShop, TextMate for Macintosh
- Multiplatform General Purpose Editors
  - Emacs (The editor of the gods) with “AucTeX” mode
  - Eclipse (a programming IDE)
- LaTeX Specific
  - TexMaker (I like that one)
  - TexStudio
- Windows
  - TEXWorks (delivered with MikTeX)

Gotchas:
- Assumes user has medium/deep understanding of computer
- Editing: Lots of “boilerplate” details
- Preamble has \usepackage{} statement for each package
  - Each macro or environment comes from some package
  - Users must learn how to install packages (hassle...)
Software to Facilitate Producing \LaTeX\ Documents

- **\texttt{LYX}** (Open Source, Multiplatform): can export to \LaTeX
  - a “document processor” with some point-and-click features
  - Allows to write “real \LaTeX” as well inside \texttt{LYX} document
  - Version 2 introduced the “on the fly” spell-checking
- **Scientific Word** (Commercial, MS Windows)- A MS Word look-alike that can create \LaTeX\ documents
- **\TeX\Macs** (Open Source) Similar in concept to \texttt{LYX}, developed by a smaller team of programmers

Generally, these provide
- Document “templates”, pre-formatted examples that work
- Facilitators for entry of formulae and special formatting
- I often use \texttt{LYX}, and export documents to \LaTeX\ format.
When Do I Edit with Emacs, not LyX?

- Some document types—multiple choice exams—have specialized \LaTeX classes for which LyX has no “customization” or “layout”
- My co-author is a \LaTeX writer who has invested years to learn how that works and refuses to try LyX
- LyX has a bug that I can’t work around.
- LyX upgrades and I don’t like their “enhancements” as much as raw \LaTeX.
Raw TeX Exercise: Compile My Terminal-1 lecture

- Edit and Compile a \LaTeX file. In my Guides repository, look for the folder Computing_HOWTO/IntroTerminal-1. Find the file “terminal-1.tex”.
  - Make a directory in your computer
  - Download terminal-1.tex and beamerthemeKU.sty in there.
  - You also need to copy the sub-directory “importfigs”. Those graphics were recently added to beautify this.

- Figure out how to open and compile the document.
  - Open a terminal, run “pdflatex terminal-1.tex”, for example.

- Because that file has a table of contents, it is necessary to run pdflatex twice
  - If your computer has a copy of the program “texi2pdf”, use that instead, it will run pdflatex as many times as necessary.
What to do next? Followup Presentations Needed

- This will become the organizing location of \LaTeX{} support documents. http://crmda.ku.edu/guides/latex-help

- In the past, and probably for a while, most of my \LaTeX{} support material has been hosted at http://pj.freefaculty.org/latex

- Notes “LyX for Beginners”
  http://pj.freefaculty.org/guides/Computing-HOWTO/LaTeXAndLyx/LyX-Beginner

- Embarrassingly, I forgot I had written those notes and agreed to give a LyX presentation, so I made a new, possibly better version of the same (with a much more enticing title) “How to Cheat on Your \LaTeX{} Homework”. This will be moved into the CRMDA guides soon.
  http://pj.freefaculty.org/guides/Computing-HOWTO/LaTeXAndLyx/LyX-for_\LaTeX{}_homework
What to do next? Followup Presentations Needed ...

- **LyX-Intermediate!**. You can monitor our progress here:
  http://pj.freefaculty.org/guides/Computing-HOWTO/LatexAndLyx/LyX-Intermediate

- **KU Thesis class & example document**

- **Developing your own LyX Template**
  http://pj.freefaculty.org/guides/Computing-HOWTO/LatexAndLyx/LyX-article-template

- **For “reproducible research” by the use of Sweave? Maybe knitr**
  http://pj.freefaculty.org/guides/Computing-HOWTO/LatexAndLyx/LyX-sweave-tutorial
What to do next? Followup Presentations Needed ...

- KUant guide templates
  http://pj.freefaculty.org/guides/Computing-HOWTO/LatexAndLyx/KUant_template
  http://pj.freefaculty.org/guides/Computing-HOWTO/LatexAndLyx/KUant_template_sweave