LaTeX Workflow Demonstration

Michael G. DeCrescenzo*

Updated September 18, 2018

This document describes how to incorporate LaTeX into a social science workflow. It covers bibliographies (using BibLaTeX) and integration with statistical software (tables, figures, and "saved quantities"). The bibliography is managed by a companion `workflow.bib` file. The software integration component focuses primarily on `R`, but the same concepts generally apply to Stata. All tables and figures are created algorithmically to maximize reproducibility throughout the workflow. All of the technological capabilities demonstrated in this document are possible using RMarkdown, with a demonstration contained in a separate folder in the online repository.[1] Slideshows using `Beamer` are also discussed in a separate folder.

1 Introduction: The value of LaTeX

LaTeX is designed to emphasize content control over aesthetic micromanagement. This document exemplifies how far "content control" goes. It covers much more than just managing *variables* about the document itself, such as the title, author name, margin sizes, and so on. It extends to more advanced topics that can be automated: bibliographies, tables, figures, and other aspects of your empirical results.

Many LaTeX hop off the train before learning these skills. They know how to create LaTeX documents, but they don't fully learn how to manage bibliographies or integrate their statistical output with their `.tex` files. These people are selling themselves short! Dig into the source code of this document to learn more.

2 Bibliographies

LaTeX lets you automate the formatting of (a) in-text citations and (b) their corresponding bibliography entries in the back of your paper.

*Ph.D. Candidate, Political Science, University of Wisconsin–Madison.

1. https://github.com/mikedecr/latex-workshop-2018

How? First, create a `.bib` file. A `bib` file contains information about works you would like to cite—the author, date, title, journal, volume and issue number, and so on. An example `bib` file is included with this document, so you can see how the syntax works.

Each entry in the `bib` file has a *cite key*, which is a short string of unique text that serves as a label for each entry. When you are writing your document, you refer to the cite key to import information about the referenced work.

2.1 Bibliography systems

You may read about different bibliography systems and packages. To learn more, follow the link in this footnote.[2] Here's a short version:

- There are two common back-end systems you could choose from: `bibtex` and `biber`. These systems interface between your `bib` file and your `tex` file. We will use `biber` because it is more flexible, powerful, and actively developed.
- There are two front-end bibliography packages you could choose from: `natbib` and `biblatex`. They are (loosely speaking) companions to `bibtex` and `biber`, respectively. They control the commands you use to summon and customize the appearance of your citations.

We will use a combination of `biber` and `biblatex`, because they're a bit simpler. Luckily, changing between systems is not too challenging, should you find that you need to modify your citation system.[3] The following code sits in the preamble:

```
\usepackage[authordate, backend = biber]{biblatex-chicago}
\addbibresource{bib/workflow-bib.bib}
```

The first line calls the `biblatex` package using Chicago style (very similar to APSA style), and the second line defines the pathway to your `.bib` file. In the body of your document, insert the bibliography section using `\printbibliography`.

2.2 Citation commands

When you read a paper, you encounter citations of various formats. This section describes how to create those citation formats yourself, using BibLaTeX syntax.

2. https://tex.stackexchange.com/a/25702

3. You can view natbib/bibtex instructions in last year's workshop: https://github.com/mikedecr/latex-workshop/blob/master/slides-2-bibs/latex-workshop-bib-slides.pdf. Thomas Leeper has an APSA-formatted bibliography style file for use with natbib here: https://github.com/leeper/apsa-leeper.bst/blob/master/apsa-leeper.bst

Textual citations "Text" citations are citations where we refer directly to an article in the text. For example, not every statistical estimate is statistically significant, but Gelman and Stern (2006) point out that statistically significant and insignificant estimates are sometimes similar to one another. This is done using `\textcite{citekey}`:

```
not every statistical estimate is statistically significant,
but \textcite{Gelman2006significance} point out...
```

Parenthetical citations The difference between "significant" and "insignificant" is not itself significant (Gelman and Stern 2006). This is done using `\parencite{citekey}`.

"Plain" citations This citation is inserted with no parentheses whatsoever. As it happens, this is most useful when you are *already within* parentheses. For example, not all of our estimates are significant, but the insignificant estimates are of a similar magnitude to the significant estimates (see e.g. Gelman and Stern 2006). Do this using `\cite{citekey}`.

Footnote citations If you are an international relations scholar, you are more likely to deal with the journal *International Organization*, which famously uses footnote citations. You can make these with `\footcite{citekey}`. Remember, footnotes go *after* punctuation. Some people said something about statistical significance once.[4]

```
Some people said something about statistical
significance once.\footcite{Gelman2006significance}
```

Invisible citations If you want to include a bibliography entry, but you have no reason to include an in-text citation, you can use `\nocite{citekey}`. The entry appears in the bibliography, but nothing else appears in the text. (You will notice a Gelman piece about multilevel modeling in the bibliography.) This is often handy for citing software or data that you rely on.

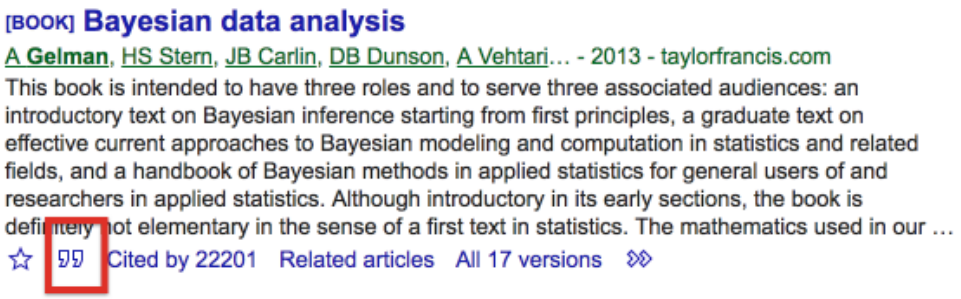
2.3 Tools for easier bibliographies

How do you get bibliographic information into your `.bib` file?

4. Gelman and Stern 2006.

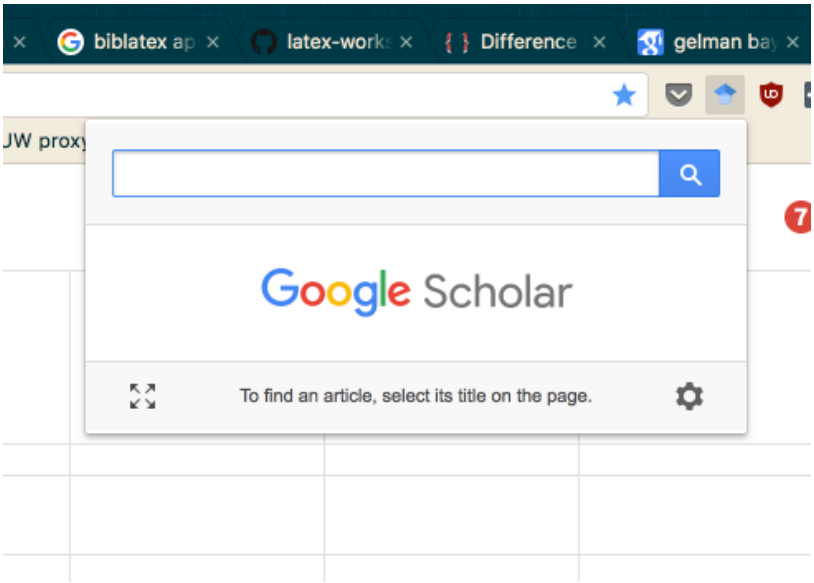
Hand-typing Enter bibliographic information by hand *if you have to*, but there are probably tools that make the process easier.

Google Scholar Google Scholar has a citation button that lets you access citations in multiple formats. Just click the ” button and ask for BibTEX format...



...and it will give you some text that you can copy into your `.bib` file.

Google Scholar Button Google offers a web extension called the *Google Scholar Button*, which gives you a menu-bar button to search and collect info using Google Scholar.



`doi2bib` The idea of `doi2bib.org` is simple. Paste a DOI number, get a `.bib` entry.

Mendeley, Zotero, BibDesk... There are a handful of reference-manager apps. You can get these apps to export `.bib` entries for the works in your library.

Various R *packages* There are some R packages out there that provide package utilities, such as importing all entries from a .bib file into a data frame for customization within R.[5]

2.4 Bibliography tips

One tip I have for using LaTeX for bibliographies is to keep a global .bib file somewhere on your computer. I keep mine in my synced Dropbox folder so it is always backed up online. You could keep yours in a git repository with a remote backup. Whichever works. Keeping a global .bib file makes it easy to include references in whatever you're working on.

However. However, however, however. If you want your project to be *reproducible*, you should eventually have a local, project-specific .bib file by the time you are distributing your project anywhere (to coauthors, to journals, etc). This ensures that your project repository is always reproducible as a folder unto itself.

There is other great advice on the web.[6]

3 Tables

As last week's handout document showed, tables are weird in LaTeX. They make syntactic sense, but the syntax itself is complicated and prone to user error. The beauty of LaTeX, however, is that its *content control* capabilities obviate the necessity to create tables by hand in almost every instance. You can have R or Stata do that for you.

The online repository for this workshop contains a folder of R code. In it, you can find a workflow.R file to accompany this document. It discusses how to create LaTeX tables algorithmically. It works generally like this:

1. Do some analysis in R or Stata.
2. Create a table.
3. Convert the table to .tex code; save the tex table in a separate .tex file.[7]
4. In your main tex file, import the tex file for the table.[8]

We will see this in action using some Pokémon data.[9]

5. https://cran.r-project.org/web/packages/bib2df/vignettes/bib2df.html
6. https://serialmentor.com/blog/2015/10/2/Bibtex
7. The R script uses a few different packages for exporting tables. Stata code generally relies on the outtex package for exporting tables as LaTeX code.
8. Yes, you can import tex documents into other texdocuments. This can actually be very handy for managing large projects such as books or, ahem, dissertations.
9. *For the record*, I am not a Pokémon fanatic, but this is a fun dataset.

3.1 An `xtable` *example*

My preferred package for outputting summary statistics from `R` to LaTeX is the `xtable` package. It straightforwardly converts data frames into nicely formatted tables with sensible options.

The `R` file processes the data, creates a table in `tex` code, and saves the table in the file `pokemon-stats.tex` (within the `tables` subfolder). To import the table, I write...

```
\input{tables/pokemon-stats}
```

...which tells LaTeX to import any code that lives in the `tables/pokemon-stats.tex` file. It happens to contain a table, so a table is produced. Because the table is contained in a `float` environment, it may not appear right here in the PDF, but it will be somewhere sensible. I can refer to it with the label I created for it (in `R`) as Table 1.

Generation	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed
1	66	77	71	72	69	73
2	71	72	73	66	74	62
3	67	82	74	76	71	67
4	73	83	78	76	77	71
5	72	82	72	72	69	68
6	68	76	77	74	75	66

Table 1: Mean Pokémon Statistics

The benefits of this system:

- Except in very strange circumstances, there will be no need to hand-type your own LaTeX tables again.
- If for any reason your results should change, the table will automatically be updated (in the `R` code), which will also update the `pokemon-stats.tex` file. As a result, the results in your LaTeX document *will always be up-to-date*. I cannot overstate how much of a big deal this is for your workflow.

Even if you want to create a table of textual information (no data), you can still use `R` to create it. The example in Table 2 uses a dataset containing only text variables to create a table of textual information. Again, I never touched LaTeX to produce this table.

Pokémon	Type	Weakness	Gary Picks
Bulbasaur	Grass	Fire	Charmander
Squirtle	Fire	Grass	Bulbasaur
Charmander	Water	Water	Squirtle

Table 2: Choices for your first Pokémon

3.2 Regression tables using `texreg`

When exporting regression tables from `R`, I prefer the `texreg` package.[10] It can process a broad range of model types and has sensible options for regression output.

In the code, we estimate two models to predict a Pokémon's HP ("Hit points", or *health*) as a function of their Defense and Attack points. It seems sensible to me that Pokémon with low HP should be somewhat compensated with higher Defense capabilities. The regressions presented in Table 3 so far contradict that intuition.

Table 3: Regressions predicting Pokémon HP

	Restricted Model	Full Model
(Intercept)	54.77***	40.77***
	(2.26)	(2.46)
Defense	0.20***	0.06
	(0.03)	(0.03)
Attack		0.31***
		(0.03)
R^2	0.06	0.18
Adj. R^2	0.06	0.18
Num. obs.	800	800
RMSE	24.81	23.12

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

3.3 Other table tips

If you want help creating LaTeX and don't want to fuss with statistical code, here's what you can do.

10. The `stargazer` package is another popular choice for regression tables.

- I'll start by being pedantic: you should probably just learn to get comfortable with the statistical code. It's better for reproducibility and for reducing human error.
- https://www.tablesgenerator.com/

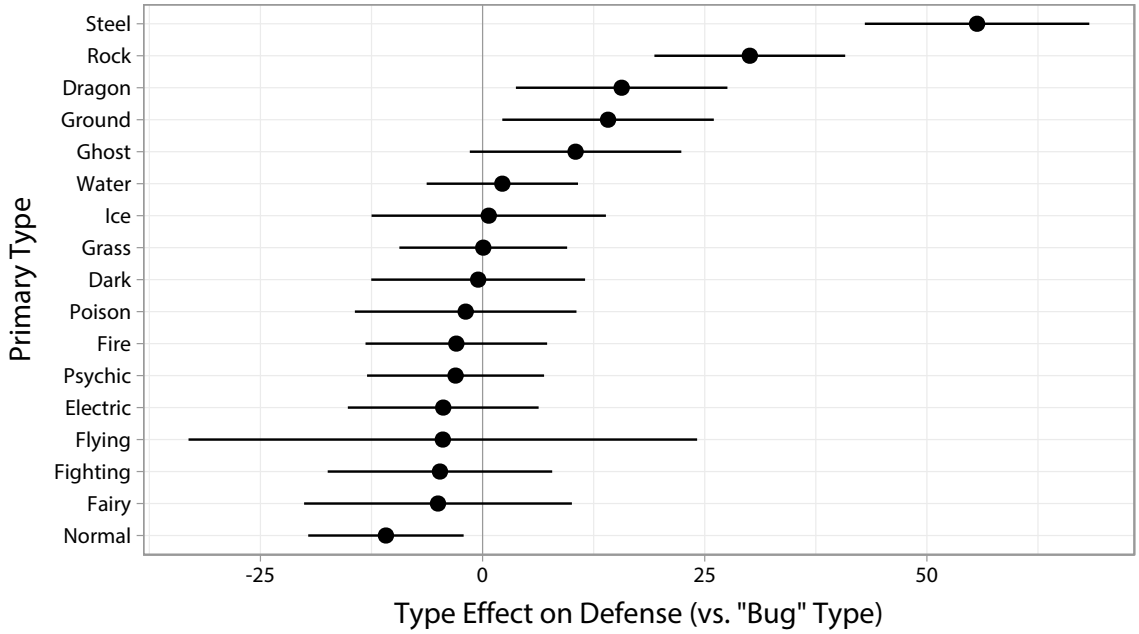
4 FIGURES

Figures follow a similar logic and rationale as tables. It's best to create them algorithmically in order to maximize the reproducibility of your project. Moreover, calling a figure file using LaTeX ensures that your paper always has up-to-date results.

It's best to export graphics as `.pdf` files, which are "vector" graphics and thus are infinitely scalable without pixelation.

Figure 1 shows the effect of Pokémon type on their defense points ("Type" is treated as a set of binary indicators with "Bug" as the omitted category).

Figure 1: Effect of Pokémon "Type" on Defense Points (multilevel model)



5 EXPORTED QUANTITIES

Let's say that I wanted to refer to some exact quantities from my R results. I could type numbers in LaTeX by hand, but that process is tedious and imprecise. Instead, I use R to save quantities directly into little .tex files, and then I pull those files into my paper using \input{}.

For example, the only type that has a defense point value that is significantly *lower* than "Bug" type is a "Normal" type Pokémon, with an effect of -10.9 points ($p = 0.015$).

```
with an effect of $\input{refs/normal-coef}$ points
($p = \input{refs/normal-p}$)
```

The largest difference compared to "Bug" type is "Steel" type, which has an average of 55.6 more defense points.

```
which has an average of $\input{refs/steel-coef}$
more defense points
```

6 CODA: EDITING SOFTWARE

An under-appreciated detail in your workflow is your code editing software. Different softwares offer different benefits—they won't all make your life easier in the same way.

An important detail for working with LaTeX is *auto-completion*—of your cite keys, file names, table and figure labels, and so on. Software like TeXMaker provides these supports.

My own setup (Sublime Text with the LaTeXTools package) has additional capabilities with syntax highlighting, code snippets, and keybindings. As you grow more advanced in your use of LaTeX and other coding languages, your preferences over these editor features may become more specific and refined. As always, I am happy to assist and advise on these choices.

I would enthusiastically recommend RStudio[11] to newcomers. Although Rstudio is designed as an R editor/IDE, it has capabilities for handling LaTeX, Rmarkdown, and other languages that are valuable for social and data sciences.

11. https://www.rstudio.com/

REFERENCES

Gelman, Andrew. 2006. "Multilevel (Hierarchical) Modeling: What It Can and Cannot Do." *Technometrics* 48, no. 3 (August): 432–435. doi:10.1198/004017005000000661. https://doi.org/10.1198/004017005000000661.

Gelman, Andrew, and Hal Stern. 2006. "The Difference Between "Significant" and "Not Significant" is not Itself Statistically Significant." *The American Statistician* 60, no. 4 (November): 328–331. doi:10.1198/000313006x152649. https://doi.org/10.1198/000313006x152649.