LaTeX Fundamentals: Part 2

Isabelle Cohen
DLab, Fall 2018

Contents

1 Welcome to LaTeX! 3

2 Tables 3

3 Footnotes and references 7

4 Stata/R to LaTeX 8
4.1 Figures 8
4.2 Summary statistics: Stata 8
4.3 Tables: Stata 10
4.4 Summary statistics and tables: R 11

5 Resources for further study 13

A Useful packages 14

B Useful symbols 15

C Templates 15

1 Welcome to LaTeX!

LaTeX is a document creation software which uses plain text with what are called "mark-up tagging conventions." We are in Section 1.

LaTeX has all kinds of useful features:

1. It right and left-justifies your text, in a way that looks much better than Microsoft Word
2. It's great for writing lengthy equations
3. It makes it easy to integrate your bibliography into your document, using what's called BibTex
4. It's easy to export statistical analysis from Stata or R directly into a convenient LaTeX format

Overall, while LaTeX might not be the easiest program to start using, it can help make your life easier, and your work look more professional.

The basic way that LaTeX works is that you write code and text together in a writing environment (like TexStudio, but it can also be simpler), then compile your code. The compiling reads the code into the text, and produces a formatted document, typically a .pdf (but we can also make a .ps or .dvi file). The text editor and the distribution (what does the compiling) are often bundled together, like in TexStudio.

In this workshop, we look at how to make tables in LaTeX, how to incorporate footnotes and references into your document, how to automate exporting from Stata or R to LaTeX, and (in a separate document) how to use Stata to make presentations.

2 Tables

LaTeX has great functionality for tables. The simplest environment for a table is called tabular. We can open it below, and build our table cell by cell:

cell1	**cell2**	*cell3*
cell4	***cell5***	cell6
cell7	cell8	cell9

Tables use & to differentiate between cells within a line, and \\ to differentiate between lines.

Another useful environment is the center environment. We can embed a tabular environment within a center environment to create a centered table, like below. We'll also add some vertical and horizontal lines between our cells:

cell1 with some extra text	cell2	cell3
cell4	cell5	cell6
cell7	cell8	cell9

Why are there three cs after we open the tabular command? This tells tabular that the table will have at most three cells per row, and that the text in these cells will be center-justified. If we replaced those cs with ls, we'd get left-justified text.

Notice how the extra text in cell1 stretched out the row? That's the default in tabular, where the column is as wide as it needs to be to accommodate all the text. If we don't like that, we can instead create fixed-width cells. For this, we'll need the array package that we installed earlier.

cell1 lots of text in this cell!	cell2	cell3
cell1 lots of text in this cell too	cell5	cell6
cell7	cell8	cell9

Now, rather than extra text stretching out the cell, the cell is fixed width, and the text wraps to the next line. The m in the tabular environment above means the height is set to the middle of the cell; we could also use p, for top, or b, for bottom.

Something we frequently want to do is combine cells in just one row - like to create a titles for some columns. This is easy to do with the multicolumn command.

	Cell title	
cell1 lots of text in this cell!	cell2	cell3
cell4	cell5	cell6

Notice how our box broke on the right of our multicell? That's because we need to specify within the multicolumn command that there should be a line on the right, like below:

	Cell title	
cell1 lots of text in this cell!	cell2	cell3
cell4	cell5	cell6

There are all kinds of other things we might want for our table, like positioning the table or adding a title. These are easier when we embed our tabular environment inside a table environment. Table and tabular have different functionalities; tabular is great for simple tables, but to add more complex features we often also need the table environment.

Once we use table, we don't need the center environment anymore; instead, inside the table, we can use centering, which will center the table for us. For tables, we can also choose from h (here), t((top), b (bottom) and P (for special page). If we want, we can also use a ! to strengthen the request, and encourage LaTeX to override its default.

Here's a table with the tabular environment embedded in the table environment. Now we can specify that the table's location with h!, and add a caption and label. The label isn't printed, but lets us refer to the table as Table 1.

Col1	Col2	Col2	Col3
1	6	87837	787
2	7	78	5415
3	545	778	7507
4	545	18744	7560
5	88	788	6344

Table 1: This table has a caption

The booktabs package also opens up a world of professional looking customization for our tables. The author of booktabs recommends only using horizontal lines, never vertical ones; this is consistent with what you'll actually see in published tables.

The following table is from the booktabs documentation, available here.

Item		
Animal	Description	Price ($)
Gnat	per gram	13.65
	each	0.01
Gnu	stuffed	92.50
Emu	stuffed	33.33
Armadillo	frozen	8.99

The new commands we used inside tabular are toprule (for the line at the top of the table), cmidrule (for the partial line under item), midrule (for the line until the column titles) and bottomrule (for the line at the bottom of the table). While not hugely different, these lines tend to look a little better than hline (what we used before). Booktabs also includes the command addlinespace, which is how we add a line of blank space within the table (as we do between Emu and Armadillo above).

We can also put captions at the top of table, as we do for Table 3, and add notes to the bottom of the table using multicolumn and footnotesize (one of the commands to control text size).

Table 2: Animal Prices

Item		
Animal	Description	Price ($)
Gnat	per gram	13.65
	each	0.01
Gnu	stuffed	92.50
Emu	stuffed	33.33
Armadillo	frozen	8.99

We can include helpful notes here.

You don't even need to copy and paste a table into LaTeX. If you have a separate file - perhaps generated via Stata or R - you can just call it into your document with the input command, as below:

Table 3: Animal Prices

Item		
Animal	Description	Price ($)
Gnat	per gram	13.65
	each	0.01
Gnu	stuffed	92.50
Emu	stuffed	33.33
Armadillo	frozen	8.99

We can include helpful notes here.

3 Footnotes and references

Footnotes are easy to include and reference in LaTeX.[1]. We can use the label we just defined to easily refer to footnote 1.

References are a little more complicated, but the LaTeX system works very, very well. With a little work, you can set up all the citations you need, and access them easily. Let's say you wanted to cite a journal article. Here's what the citation looks like typed out:

Arrow, Kenneth J., Leonid Hurwicz, and Hirofumi Uzawa (1961), "Constraint qualifications in maximization problems." *Naval Research Logistics Quarterly*, 8, 175191.

We could handle all our citations manually - but we can use Bibtex to do better. Bibtex is actually its own software, but generally used with LaTeX to make your life easier. There are a few things you need to do to use Bibtex:

- Set up a bibtex file where all your citation info is stored
- References to any citations where necessary/appropriate
- Pick a bibliography style and call it in your LaTeX file
- Generate your bibliography

Here's what the entry would look like in the .bib file:

```
@article{arrow1961,
Author= {Kenneth J Arrow and Leonid Hurwicz and Hirofumi Uzawa},
Journal = {Naval Research Logistics Quarterly},
Pages = {175-191},
Title = {Constraint qualifications in maximization problems},
Volume = {8},
Year = {1961}}
```

File types include book, article, report, conference proceedings (inproceedings), and even a generic misc format. Each object in the citation - Author, Journal, Pages, Title, Volume and Year in this case - is formatted the same way. The { } before and after define the words, with a comma to separate lines. When we reference the citation inside the document, we'd get [Arrow et al., 1961].

Notice that using this requires a few other things as well, as below. We're using the simple plainnat style from the natbib package.[2]

To use the bibliography, we'll need to:

1. Specify a bibliography style and references list
2. Run the LaTeX code (it will have errors - don't worry about them)

[1] See?

[2] For more on natbib, see here.

3. Run the bibliography from the tex document

4. Run the code again, twice; this integrates our references.

Our reference list will show up where we call the bibliography.

References

K. J. Arrow, L. Hurwicz, and H. Uzawa. Constraint qualifications in maximization problems. *Naval Research Logistics Quarterly*, 8:175–191, 1961.

4 Stata/R to LaTeX

It's very easy to export high-quality tables from Stata or R to LaTeX. We'll discuss figures in both softwares, then how to export summary statistics and regression results from either.

4.1 Figures

In either software, once you export a figure into a non-native format (such as a .jpg or .png), you can use the includegraphics command to add it to your tex document. If you want, Stata also has the user-written graph2tex command, which converts the most recently produced figure into an .eps file - which works well in Stata - and displays sample code for incorporating it in your Review window.

4.2 Summary statistics: Stata

For Stata, the user-written command sutex produces excellent summary statistics tables. To install, just type

```
ssc install sutex
```

Once installed, you can type – sutex [varlist] – where varlist is a list of all the variables you want. If you leave it blank, the command will include every variable in the dataset. You can specify options like label (to include variable labels instead of names), minmax to include minimums and maximums, and a title. You can copy and paste the results from the Review window into LaTeXor save them using the file(filename) option.

Table 4: Summary statistics

Variable	Mean	Std. Dev.	N
Make and Model			0
Price	6165.257	2949.496	74
Mileage (mpg)	21.297	5.786	74
Repair Record 1978	3.406	0.99	69
Headroom (in.)	2.993	0.846	74
Trunk space (cu. ft.)	13.757	4.277	74
Weight (lbs.)	3019.459	777.194	74
Length (in.)	187.932	22.266	74
Turn Circle (ft.)	39.649	4.399	74
Displacement (cu. in.)	197.297	91.837	74
Gear Ratio	3.015	0.456	74
Car type	0.297	0.46	74

Table 5: Summary Statistics

Variable	Mean	Std. Dev.	Min.	Max.	N
age in current year	39.153	3.06	34	46	2246
married	0.642	0.48	0	1	2246
current grade completed	13.099	2.521	0	18	2244
union worker	0.245	0.43	0	1	1878
hourly wage	7.767	5.756	1.005	40.747	2246
usual hours worked	37.218	10.509	1	80	2242

4.3 Tables: Stata

The two main commands in Stata to produce publication-quality tables in LaTeX are outreg2 and estout. To install estout, just type

```
ssc install estout
```

When you run regressions, you can store the results using the estimates store (or eststo) command, as a prefix or after the regression is run. Once you've run all the regressions you want, you can export them into a LaTeX table.

Table 6: Wage Correlates

	(1)	(2)	(3)
current grade completed	0.743***	0.694***	0.744***
	(0.046)	(0.034)	(0.046)
union worker		1.093***	
		(0.201)	
married			-0.540**
			(0.240)
N. of obs.	2244	1876	2244
Control Mean	7.767	7.767	7.767

Standard errors in parentheses

* $p < 0.10$, ** $p < 0.05$, *** $p < 0.01$

The Stata code that generated it:

```
reg wage grade
eststo, title(Reg1)
summarize wage
estadd scalar ymean = r(mean)
reg wage grade union
eststo, title(Reg2)
summarize wage
estadd scalar ymean = r(mean)
reg wage grade married
eststo, title(Reg3)
summarize wage
estadd scalar ymean = r(mean)

esttab using "sample_table1.tex", title("Wage Correlates") ///
varwidth(32) b(%6.3f) label se nomtitles replace booktabs ///
drop(_cons) ///
stats(N ymean, fmt(0 3) labels ("N. of obs." "Control Mean")) ///
star (* 0.10 ** 0.05 *** 0.01) width(1.0\hsize) compress
eststo clear
```

Let's walk through this line by line. We run a regression, then we use the eststo command to store the results. On the next line, we run a summary stats command, and store the output in the scalar ymean, using the estadd command. That combination is repeated three times, and takes up the first twelve lines.

Once we have the regression we want, we use the esttab command.

- We need to name the file we want to save (after using)
- we title the table
- We specify that no line is wider than 32 characters
- We set the coefficient formatting (no more than three decimals)
- We tell Stata to use variable labels, standard errors (rather than t-stats), no model titles
- We tell Stata to overwrite any existing files with the same name
- We use the booktabs package to get top and bottom rule lines
- We drop the constant; include the sample size with no decimal places, the mean with three, and label them; and specify significance thresholds
- Finally, we set the width of the entire table and reduce the amount of horizontal space with compress

There are many, many more ways to customize your tables. For more, type the following into Stata after esttab is installed:

```
help esttab
```

4.4 Summary statistics and tables: R

In R, the *stargazer* package will be your best friend for producing high-quality tables.

To use stargazer, we would first make sure we have it installed on our computer, then add it to our library:

```
install.packages("stargazer")
library(stargazer)
```

For a simple table of summary statistics, we just type stargazer() and the name of our dataframe.

```
stargazer(birthwt)
```

which produces:

For regression analysis, we run the regression and store the results, then use stargazer to export them:

Table 7

Statistic	N	Mean	St. Dev.	Min	Max
low	189	0.312	0.465	0	1
age	189	23.238	5.299	14	45
lwt	189	129.815	30.579	80	250
race	189	1.847	0.918	1	3
smoke	189	0.392	0.489	0	1
ptl	189	0.196	0.493	0	3
ht	189	0.063	0.244	0	1
ui	189	0.148	0.356	0	1
ftv	189	0.794	1.059	0	6
bwt	189	2,944.587	729.214	709	4,990

```
reg1 <- lm(bwt ~ age + smoke, data=birthwt)

#Stargazer table, for LaTeX
stargazer(reg1,out="sample_table2.tex",
title="Birthweight on Mother's Age and Smoking Habits",align=TRUE,
omit.stat=c("LL","ser","f","adj.rsq"),no.space=TRUE)
```

This generates the following table, which prints in our display and is saved as a file:

Table 8: Birthweight on Mother's Age and Smoking Habits

	Dependent variable:
	bwt
age	11.290
	(9.881)
smoke	-278.356^{**}
	(106.987)
Constant	$2,791.224^{***}$
	(240.950)
Observations	189
R^2	0.043
Note:	*p<0.1; **p<0.05; ***p<0.01

As before, there are many, many more options. We've used just a few:

- out, to name where we'll put the file
- title, to specify our title

- align=TRUE to align columns along decimals
- omit.stat to drop some possible regression summary statistics
- no.space=TRUE to remove all empty lines

This simple tables take just minutes to create, but look great! Once you find formatting options that you like, you can just copy and paste them between different sets of analysis to make the tables even faster, with barely seconds longer than it takes to run your regressions.

5 Resources for further study

There are a ton of "getting started in LaTeX" resources:

- Getting to Grips with LaTeX
- The Not So Short Introduction to LaTeX 2ϵ
- CTAN's Starting Out in LaTeX Guide

A Useful packages

There are a number of useful packages in LaTeX. Here are the packages we use in the Fundamentals series:

- soul: highlighting
- appendix with

  $$toc, page$$

  : lets us make an appendix
- enumitem: this package lets us customize list types
- amsmath: this package will let us use the align environment, and generally make our equations prettier
- hyperref: lets you embed hyperlinks into your document
- booktabs: one of the best packages for table formatting
- array: lets us create fixed width cells
- graphicx: a great package for adding figueres
- caption: for making subcaptions in figures
- subcaption: for making subcaptions in figures
- dcolumn: required for stargazer
- natbib: for making our citations nice

We can't cover all of them today, so here are a few others that you might be interested in:

- tikzpicture: lets you draw lines and geometric shapes within a TeX document. See this link
- fancyhdr: lets you add custom headers and footers into your LaTeX document. See this link
- pdfpages: lets you include pdfs into your TeX document. See this link
- tabularx: lets you create tables that are the same width as the text. There are subtle differences between how tabular, tabular*, and tabularx handle spacing within tables; this stackexchange has some helpful examples.
- xspace: creates spaces after commands (like LaTeX); you can use xspace in creating commands to insert that extra space
- bibtex and biblatex: both are alternatives to natbib which contain other options for formatting. This page on bibliography management in LaTeX is a great resource.
- geometry: lets you customize header, footer and margin width in your document. See this link

B Useful symbols

All predefined mathematical symbols in LaTeX are available here. An incredibly comprehensive list of 2,590 symbols with the necessary packages and commands is available here. The second link also discuss amsmath extensively.

C Templates

Overleaf has a great database of LaTeX templates.