Introduction to LaTeX-Stata Integration

J-PAL

Last updated: January 2024

1 Introduction

This exercise demonstrates how to create a final, reproducible document by integrating LaTeX with Stata to ensure automatic updates to your final report. The exercise walks through the do-file `latex_example.do`, which uses public data from a randomized evaluation of a remedial education program in urban India between 2001 and 2004. For more information on the program and data, see the dataverse entry. To begin, open the do-file `latex_example.do`, located in the `dofiles` folder.

2 Setting up your Stata environment

We start by clearing your current Stata settings and setting your working directory using a `global`[1]. Your `global path` will be your main working directory and should be the main folder which contains all of your project sub-folders. Additional globals for the data, figures and output folders are then set by calling on the main path.

```
clear all
set more off

global path "[SET DIRECTORY HERE]"

* Set subdirectories
global data "$path/data"
global figures "$path/figures"
global output "$path/output"
```

3 Data Cleaning and Set-up

After importing the data, we do some simple data cleaning to prepare for our demonstration exercises. First we drop observations that are missing the variables `pre_tot` (test scores before the intervention) or `post_tot` (test scores after

[1] If you are unfamiliar with using globals and locals in Stata, please review the Stata manual for macros here.

implementation of the intervention) – one of our exercises will require data for both variables for all observations. We then create dummy variables for whether a student is in grade 3 and grade 4. Finally, we clean the variable labels for your main variables, as these will be exported as labels for the tables later.

```
use "$data/baroda_0102_1obs.dta", clear
drop if pre_tot == . | post_tot == .

* Dummy var for grades:
gen gr3 = (std == 3)
gen gr4 = (std == 4)

* Clean Variable Labels:
label variable bal "Treatment"
label variable female "Female"
label variable bigschool "Big School"
label variable std "Grade"
label variable gr3 "Grade 3"
label variable gr4 "Grade 4"
label variable pre_tot "Pre-test Score"
label variable post_tot "Post-test Score"
```

4 Exporting individual tables in pre-formatted LaTeX code

If you have results that do not require extensive or special formatting, using the `esttab` command is relatively easy to get your results into LaTeX code. We start the exercise by running a regression and storing its estimates using `est sto`. After storing the results, we export them into a `.tex` file using `esttab` by specifying the `.tex` extension. You can then integrate the code in the `.tex` file into your full LaTeX document. To further customize your table, you can use optional commands from the `esttab` package in Stata. For example code, see Luke Stein's Github Documentation.

```
* Run regression and store estimates
reg post_tot bal
est sto reg1

* Export with pre-formatted Latex code
esttab reg1 using "$output/reg1.tex",  ///
label title (Regression Results) replace
```

Your resulting table will look like the one below:

Table 1: Regression Results

	(1) Post-test Score
Treatment	2.705***
	(5.07)
Constant	40.88***
	(110.11)
Observations	8426

t statistics in parentheses

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

5 Integrating LaTeX within Stata

When your results require more extensive formatting, you may choose to create your full LaTeX report within Stata. This can help automate your workflow so that you do not need to tweak individual tables if the underlying data or analysis changes.

5.1 Storing results in locals

Storing your results in locals before exporting them can be an easy way to organize your workflow without creating extra variables. All of your analysis can be conducted in the same place and exported later in your code. (To use locals, you must run all of the code that references your locals at the same time.)

5.1.1 Table 2. Descriptive Statistics

In the following code we store the average values of the covariates for the analysis in locals separately for the treatment and control groups using a `foreach loop`.

```
* Store average share in locals
local stats pre_tot post_tot female gr3 gr4 bigschool

foreach var in ‘stats’ {
// treatment
sum ‘var’ if bal == 1
local ‘var’_count_t = ‘r(N)’
local ‘var’_mean_t = round(‘r(mean)’, .01)
// control
sum ‘var’ if bal == 0
local ‘var’_count_c = ‘r(N)’
local ‘var’_mean_c = round(‘r(mean)’, .01)
}
```

5.1.2 Table 3. Regression Results (with and without covariates)

We then run two regressions (one without covariates and one with covariates) and store the estimates of the regressions using locals and a `foreach loop`:

```
* Regression without covariates
reg post_tot bal

* Store regression results in locals:
local b_reg1 = round(_b[bal],0.001)
local N_reg1 = e(N)
local R_reg1 = round(e(r2), 0.001)
local se_reg1 = round(_se[bal], 0.001)

* Calculate P-value manually and store in local:
local p = 2 * ttail(e(df_r), abs(_b[bal]/_se[bal]))
local p_reg1 = string(round(‘p’, 0.001), "%3.2f" )

* Regression with covariates
local controls = "female bigschool std"
reg post_tot bal ‘controls’

* Store regression results in locals:
// Treatment
local b_reg2 = round(_b[bal],0.001)
local N_reg2 = e(N)
local R_reg2 = round(e(r2), 0.001)
local se_reg2 = round(_se[bal], 0.001)
local p = 2 * ttail(e(df_r), abs(_b[bal]/_se[bal]))
local p_reg2 = string(round(‘p’, 0.001), "%3.2f" )
di "‘p_reg2’"

// Covariates
foreach var in ‘controls’ {
local b_reg2_‘var’ = round(_b[‘var’],0.001)
local se_reg2_‘var’ = round(_se[‘var’],0.001)
local p = 2 * ttail(e(df_r), abs(_b[‘var’]/_se[‘var’]))
local p_reg2_‘var’ = string(round(‘p’, 0.001), "%3.2f" )
}
```

5.1.3 Figure 1. Gains in test scores, by pre-test score group

Finally, you may want to include reproducible figures in your LaTeXdocument as well. The following code separates students into 10 bins by their pre-test scores, and then calculates average post-test scores. The code then creates a

scatter plot showing improvement in test scores and scales the markers for each bin by the number of students in that bin. We then export the figure into the `figures` folder using the `.png` format.

```
* Calculate difference in scores and bin by 10 groups
gen diff_scores = post_tot - pre_tot
gen bins  = 10 * floor(pre_tot/10)
gen _freq = 1

preserve
collapse (mean) pre_tot diff_scores (sum) _freq, by(bins bal)


* Scatterplot
twoway (scatter diff_scores pre_tot  if bal == 0 [fw = _freq],         ///
     mcolor("228 92 36")  msymbol(circle_hollow) msize(vsmall))        ///
     (scatter diff_scores pre_tot if bal == 1  [fw = _freq],           ///
     mcolor("44 172 156") msymbol(circle_hollow) msize(vsmall)),       ///
     legend(order(1 "Non-Balsakhi" 2 "Balsakhi") size(small))          ///
     xlabel(, grid gstyle(linestyle(color(white))) labsize(small))     ///
     ysc(r(-10 20)) ylabel(-10 "-10"  0 "0" 10 "10" 20 "20")           ///
     ylabel(, grid gstyle(linestyle(color(white))) labsize(small)      ///
     angle(0)) plotregion(color("220 220 220%30"))                     ///
     graphregion(fcolor(white) ifcolor(white) color(white)             ///
     icolor(white)) ytitle("Average Test Score Improvement (in Bin)",  ///
     margin(small) size(small))                                        ///
     xtitle("Pre-test Scores" "(Scaled by Number of Students in Bin)", ///
     margin(small) size(small))
restore


* Export figure:
graph export "$figures/figure_1.png", as(png) replace
```

6 Exporting Results into Reproducible LaTeX Document

After conducting your analysis and creating any relevant figures, you can export their results into your LaTeX document and dynamically update the final report whenever you make any changes to your original analysis.

6.1 Setting up your LaTeX file in Stata

To open or create a new LaTeX document, you can use the `file open` command. To write LaTeX code within Stata, you can use the `file write` command. While `file write` can take multiple lines of LaTeX code, an easy way to keep your code clean and manageable is to start off by programming a command that adds an additional line after the current one. [2]

In the program below called `line`, the `file write` command will write within the `Report` file anything that you specify as the 1 (i.e., the first thing that comes after the command `line`). All Stata code below (apart from locals being exported) will be outside of the quotation marks, while LaTeX code will be within the quotation marks. The `_n` specified at the end of the command signifies that the line is ending and that the LaTeX code should continue on the next line.

```
* Program for adding a new line in your tex file:
cap program drop line
program define line
file write Report "`1'" _n
end
```

We first create a `.tex` file by using the `file open` command, writing in the document name ("Report" in this example) and specifying its location in your folders. Then we set our `document class` and load the LaTeX packages we need. After loading the packages, we add the title, author and the date, and begin the document.

```
* Open Tex doc:
file open Report using "$output/Report.tex", write replace

* Set your document class
line "\documentclass{article}"

* Load Latex packages
line "\usepackage{graphicx, caption, placeins, color, hyperref}"
line "\usepackage{booktabs, array, morefloats, tabularx, titling}"
line "\usepackage{multirow, subfig, longtable, threeparttable, wrapfig}"


* Create document title, author & date
line "\title{Stata-LaTeX Integration}"
line "\author{J-PAL}"
line "\date{Last Updated: December 2023}"
```

[2] If you are unfamiliar with programming commands in Stata, please review syntax in the Stata manual for programming here.

```
* Begin document
line "\begin{document}"
line "\maketitle"
```

6.2 Exporting your results into LaTeX tables

We next format the locals we saved for `Table 2: Descriptive Statistics` into a table, adding the locals directly within the quotation marks that indicate output for LaTeX. The table has some of the following arguments:

- `\begin{tabular}{p(4.5cm}ccc}` begins the tabular format and sets the table to have 3 columns: with the column on the left set as 4.5cm wide.
- `\toprule \toprule` creates two lines at the top of your table, while `\midrule` and `\bottomrule` each add one line in the middle and bottom of the table, respectively.
- ` & Treatment & Control \\` skips the first column by specifying the `&` character and adds the words `Treatment` in the second column and `Control` in the third column.
- the `foreach` loop exports the local label and averages for all of the variables stored in the local `stats`.

```
* Table 1: Descriptive Stats
line "\begin{center}"
line "\begin{threeparttable}"
line "\caption{Descriptive Statistics}"
line "\begin{tabular}{p{4.5cm}ccc}"
line "\toprule \toprule"
line "& Treatment & Control \\"
line "\midrule"

* Place stored locals into table:
foreach var in `stats' {
    line " `: var lab `var'' & ``var'_mean_t' & ``var'_mean_c' \\ "
    }

* Add the number of observations
line "\midrule"
line "Observations & `post_tot_count_t'  & `post_tot_count_c'  \\ "
line "\bottomrule"

* End table & add note
line "\end{tabular}"
line "\multicolumn{5}{l}{\footnotesize \emph{Notes:} Note. }"
line "\end{threeparttable}"
line "\end{center}"
```

Your resulting table should look like the one below:

Table 2: Descriptive Statistics

	Treatment	Control
Pre-test Score	32.32	32.88
Post-test Score	43.58	40.88
Female	.51	.52
Grade 3	.52	.49
Grade 4	.48	.51
Big School	.64	.67
Observations	4084	4342

Notes: Test scores can range from 0-100.

Next, we write code for exporting the regression results for Table 3 we stored earlier. The process will be very similar, with slight adjustments that allow for specialized formatting. Here, our table is five columns as specified by the five `c` values within the `tabular` argument. Remember that adding the `&` symbol indicates that the next column is beginning, so if we want to skip a column we would add two `&` symbols next to each other.

```
line "\begin{center}"
line "\begin{threeparttable}"
line "\caption{Regression Results}"

line "\begin{tabular}{p{3cm}ccccc}"
line "\toprule \toprule"

line "& & Post-test Score & \\"
line "& Coefficient & P-value & Coefficient & P-value \\"
line "\midrule"
line "Balsakhi Treatment &  `b_reg1'  & `p_reg1' & `b_reg2' & `p_reg2' \\"

line "& (`se_reg1')  &     &  (`se_reg2') &  \\"

* Loop for exporting local controls into table
foreach var in `controls' {
line " `: var lab `var''  &   & &  `b_reg2_`var'' & `p_reg2_`var'' \\ "
line "& & &  (`se_reg2_`var'')  &  \\"
}

line "\midrule"
line "Observations & `N_reg1'  & &  `N_reg2' & \\ "
line "R^2 & `R_reg2'  & &  `R_reg2' & \\ "
```

```
line "\bottomrule"

* End Table 3
line "\end{tabular}"
line "\multicolumn{5}{l}{\footnotesize \emph{Notes:} Standard errors in parentheses. }"
line "\end{threeparttable}"
line "\end{center}"
```

Your resulting table should will look like the one below:

Table 3: Regression Results

	Post-test Score			
	Coefficient	P-value	Coefficient	P-value
Balsakhi Treatment	2.705	0.00	3.135	0.00
	(.533)		(.51)	
Female			2.292	0.00
			(.511)	
Big School			-2.042	0.00
			(.537)	
Grade			14.103	0.00
			(.51)	
Observations	8426		8426	
R^2	.09		.09	

Notes: Standard errors in parentheses.

6.3 Exporting your figures from Stata into LaTeX

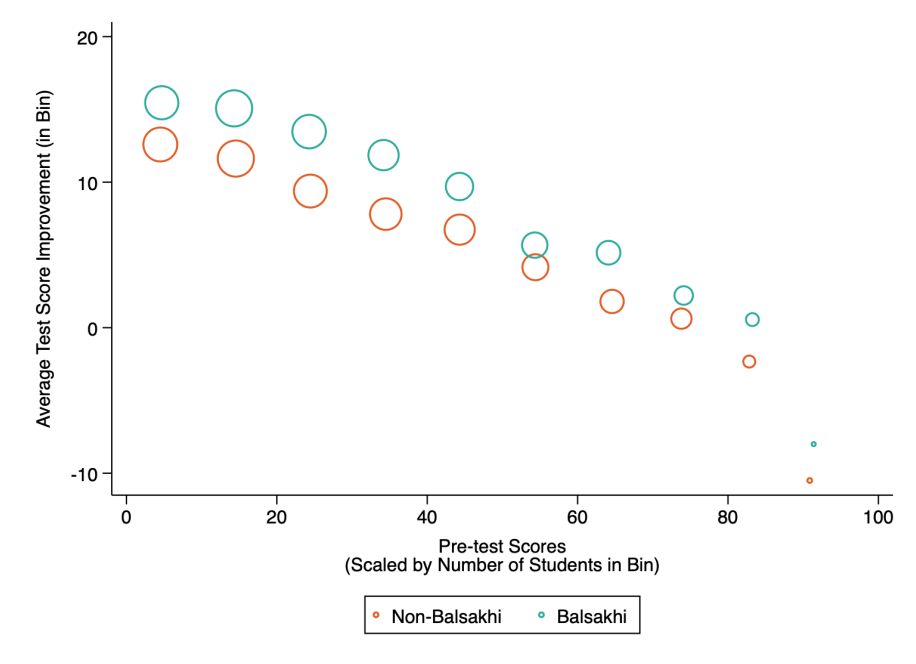
Figures can be easily exported with LaTeX. We finish the exercise by using the `\includegraphics` argument to do so, specifying the width of the figure and the folder directory where it can be found. If you are using an online LaTeX compiler (e.g., Overleaf), you can upload your figures into a virtual folder embedded in the compiler with the same name as your `figures` folder and the code will pull the figure directly from the virtual folder. The `\minipage` argument allows us to add a note under your figure with the argument `\footnotesize` that is the width of the figure and moves correspondingly.

```
\begin{figure}[h]
\centering
\caption[online] {Improvement in test scores following program intervention,
by pre-test score bin}
\begin{minipage}{1\textwidth}
\includegraphics[width=1\linewidth]{./figures/figure_1.png}
{\footnotesize \emph{Notes:} Students are divided into 10 bins based on their
```

```
pre-test scores. Markers are scaled by the number of students in each bin.\par}
\end{minipage}
\end{figure}
```

The code exports the following figure:

Figure 1: Improvement in test scores following program intervention, by pre-test score bin



Notes: Students are divided into 10 bins based on their pre-test scores. Markers are scaled by the number of students in each bin.

6.4 Close out your document

To end the document, we need both a Stata and LaTeX closing command:

```
* End your document
line "\end{document}"
file close Report  // close tex file
```