

## FLEXIBLE CODE FOR BALANCE AND SUMMARY TABLES IN STATA<sup>1</sup>

Most empirical research papers (and thesis) contain tables with summary statistics. Similarly, research using randomized controlled trials will need to show balance tables to convince readers of the successful randomization. There are standardized tools to produce these tables, however, they often lack the flexibility.<sup>2</sup> However, the exact formatting requirements vary with data structure and RCT design.

This code will enable you to flexibly create LATEX table using Stata. For example, you can relatively easily want to show balance for different subsets of observations in a different panel. In the provided example this reveals that female respondents are not balanced on work experience.

You can also easily add columns with additional variables, for example the number of observations. This is useful if you have varying numbers of observations for each row (e.g. due to missing values as in the example below).

|  | Treatment mean | Control mean | $\Delta$ | p-value(T=C) | # obs |
|--|----------------|--------------|----------|--------------|-------|
| <b>Panel A: Pooled</b>                   |                |              |          |              |       |
| Female                                   | 0.48           | 0.52         | -0.03    | 0.226        | 1322  |
| Age                                      | 27.60          | 27.18        | 0.43     | 0.250        | 1319  |
| Work experience                          | 2.57           | 2.42         | 0.15     | 0.559        | 1331  |
| Vocational education                     | 0.12           | 0.12         | 0.01     | 0.769        | 1331  |
| University education                     | 0.55           | 0.55         | 0.00     | 0.949        | 1331  |
| # observations with treatment assignment | 680            | 651          |          |              |       |
| <b>Panel B: Women</b>                    |                |              |          |              |       |
| Age                                      | 25.39          | 24.95        | 0.43     | 0.335        | 657   |
| Work experience                          | 1.99           | 1.38         | 0.61     | 0.004        | 661   |
| Vocational education                     | 0.22           | 0.19         | 0.02     | 0.475        | 661   |
| University education                     | 0.55           | 0.61         | -0.06    | 0.099        | 661   |
| # observations with treatment assignment | 327            | 334          |          |              |       |

The template do-file produces the balance table above, but it can easily be adjusted to fit your specific needs. You can generate your own balance or summary table in seven simple steps. Before starting, you should have a clear idea about the format of your table (e.g. number of columns and panels). Then implement the following steps (these only capture changes to the do-file that need to be made to adjust the table to your needs).

<sup>1</sup> This Coder's Corner contribution is adapted from the publicly available code by de Quidt et al. (2018).

<sup>2</sup> For example, the `iebalstab` command in the [ietoolkit](#). `ietoolkit` is great at producing standard balance tables, however, it lacks the flexibility of this code.

1. Make sure that all variables are labelled with the text you want to appear in the table. Here is a quick reminder of the code:

```
label var female "Female"
```

2. Create variables indicating empty lines and panel headings. It creates empty variables with an empty label.  
For empty lines use the following code:

```
gen empty=.  
label var empty " "
```

For panel headings lines use the following code. It creates empty variables with labels (note that you can use LATEX code in labels).

```
gen pana = .  
label var pana "\textbf{Panel A: Pooled} \\"
```

3. Define a local foreach panel that contains all variables in the desired order.

```
loc covars_pana "female age exp vocational university"  
loc covars_panb " age exp vocational university"
```

4. Choose number of columns in table and set local accordingly.

```
local number_columns = 5
```

5. Define columns headings in a local (note the need for LATEX code in the headings).

```
loc colnames ""Treatment mean" "Control mean" "$\Delta$" "p-value(T=C)"  
"\# obs" "
```

6. Add desired statistics to the regression results. For the complete code refer to the do-file, which contains code for a loop over all variables in a given panel and shows how to add content only to specific columns (here the number of observations with treatment assignment).

The code crucially relies on the “estadd” command which allows you to add any string or number to a set of regression results. The do-file contains a variety of examples of how to add summary statistics, strings, and p-values. The following code adds the treatment group mean of the variable “female” to the estimation results col1 (column 1) displaying 2 decimal digits (%9.2f). The result can be accessed using the name “female\_`panel’” (the “panel” local is set at the beginning of each panel).

```
qui sum female if treatment ==1  
estadd loc female_`panel' = string(r(mean), "%9.2f"): col1
```

Finally, you just need to set the desired output path and you are good to go.

### **References:**

De Quidt, Jonathan, Johannes Haushofer, and Christopher Roth. 2018 “Measuring and Bounding Experimenter Demand.” *American Economic Review*, 108 (11): 3266-3302. DOI: 10.1257/aer.20171330