

Publications quality tables in Stata: a tutorial for the `tabout` program

Ian Watson
mail@ianwatson.com.au

1 Introduction

`tabout` is a Stata program for producing publication quality tables.¹ It is more than just a means of exporting Stata results into spreadsheets, word processors, web browsers or compilers like \LaTeX . `tabout` is actually a complete table building program. This tutorial (which has the appearance of a *Stata Journal* article, and may turn into one in the future)² is intended to present a complete overview of `tabout`, with numerous examples of syntax and the kind of tables produced. You might like to flick ahead and skim these examples before reading the more detailed exposition which follows.

`tabout` allows a novice Stata user to produce multiple panels of cross-tabulations, and to lay out the data in a number of different ways. The output can be oneway or twoway tables of frequencies and/or percentages, as well as summary statistics (means medians etc). Standard errors and/or confidence intervals, based on Stata's `svy` commands, can also be included. Furthermore, a number of statistics (`chi2`, Gamma, Cramer's V, Kendall's tau) can be placed at the bottom of each panel. Finally, formatting of cell contents is simple, and allows users to choose the number of decimal places, and to insert percentage symbols and currency symbols. Before looking more closely at some of the features available in `tabout`, it is worth outlining briefly the design principles behind the program.

At a minimum, *publication quality* tables should be both *informative* and *aesthetically pleasing*. In his discussion of what makes for graphical excellence, Edward Tufte (Tufte 2001) listed several important aspects of data presentation including the following:

1. present many numbers in a small space;
2. encourage the eye to compare different pieces of data.

While Tufte had graphs in mind, the same advice helps define what is meant by 'informative' when it comes to tables. In the case of `tabout`, multiple panels play this role. As will become evident later, repeating vertical panels allow for the succinct presentation of a considerable amount of data. Moreover, comparisons between populations and sub-populations within the one table are also easily achieved using `tabout`.

1. Current version 2.0.3. 17 April 2007

2. This tutorial makes use of the Stata Journal \LaTeX style. These files are freely available for download (see `help sjlatex` inside Stata). Thanks to StataCorp for making them available.

Tufte’s book also canvassed aesthetics, though some critics might argue that his minimalist approach to many of the classic statistical graphs has gone too far. Nevertheless, his core idea of maximising the data component, and minimising the decorative junk, makes for a lot of sense when it comes to table design. It coincides with the sentiments of Simon Fear, the author of the \LaTeX package, `booktabs` (Fear 2003). With respect to the use of lines (called rules in \LaTeX), Fear advocated that one should ‘never, ever use vertical rules’, and, more controversially, one should ‘never use double rules’. These principles—or at least the first one—are commonly followed in the tables presented in academic journals, and routinely violated in the business-type tables produced by spreadsheets.

Further refinements suggested by Fear (and implemented in his `booktabs` package) include: increasing the thickness of rules at the top and bottom of tables compared with the lines used for the mid-rules; and using a small but discernible amount of additional spacing above and below rules. Anyone who has tried to implement these principles inside a word processor knows how tedious this task is, making \LaTeX the obvious choice for achieving aesthetic goals such as these. In the case of `tabout`, the aesthetics largely come through exporting the output as a \LaTeX document and making use of a number of `tabout` options. These include variable rule thicknesses and spacings, rules which span a set number of columns, and the rotation of value labels in the table headers to achieve an economical layout which avoids the ugliness of hyphenation. An additional advantage in using \LaTeX with `tabout` is the ease with which it allows for the batch production of tables, particularly large numbers of routine tables (as often occurs in an appendix).

Despite my bias towards \LaTeX , `tabout` also provides many advantages to those Stata users who import their descriptive tables into spreadsheets or word processors, or require html output. As will be evident below, these users can also gain great efficiencies in using `tabout`, since very little further processing of the cell entries is required once the appropriate options have been turned on in `tabout`.

Principles of user-friendliness underlie both the design and the syntax of `tabout`. While `tabout` aims to offer considerable customisation and flexibility to the end user, it tries to do this without becoming overly complex. It stays close to Stata principles, and also implements a number of consistent requirements in its syntax. There is a preference for single terms as options, either as a switch (simply turning on `svy`, for example to achieve survey results) or as a switch with a single value (such as the `layout` option). `tabout` avoids options within options, with the consequent need for numerous balanced parentheses. Instead, using two or three separate switches (where needed), each with a single value, is the preferred approach. The `n` option, for example, which provides sample counts in a table has five siblings: an `npos` (the position), an `nlab` (the label), an `nwt` (the weight), an `nnoc` to suppress the display of commas and an `noffset` to control placement of `n` counts. `tabout` also allows for the ‘incomplete’ entry of values, and makes up the additional values by repeating the last value entered (for example, with formats or labels). Finally, `tabout` also tries to capture most syntax errors at the outset and to provide a simple explanation. To facilitate this, a table outlining the allowable features of `tabout` is presented to users when they make syntax errors. (This table is presented later in this tutorial.)

2 Overview

What kinds of tables does `tabout` produce? Using a simple terminology, you can produce *basic* tables and *summary* tables. The first are twoway and oneway tables of frequencies and percentages. Essentially, all of the output from Stata's `tabulate` is available in a basic table. You can also produce basic tables with standard errors and confidence intervals, reflecting most of the output from Stata's `svy:tab` commands. As for *summary* tables, these are twoway or oneway tables of summary statistics derived from Stata's `summarize` command but laid out in a much more aesthetically pleasing fashion. In many respects, these tables mimic most of the output from Stata's `tabstat` and `table` commands. Finally, you can also have summary tables with standard errors and confidence intervals, though this is restricted to mean values. These tables make use of Stata's `svy:mean` command.³ With a large dataset, the survey option in `tabout` can be quite slow. This is partly because the survey commands run slower in Stata 9 than they did in previous releases, and partly because `tabout` needs to run the survey commands twice to retrieve column and row totals. The latest version of `tabout` now includes the 'dot counter' view, which indicates to the user that something is actually happening (however, slowly).

With a basic table, the cells in a table can be any one or all of the following: frequencies, cell percentages, column percentages, row percentages, cumulative percentages. With summary tables the list is quite extensive: N mean var sd skewness kurtosis sum uwsum min max count median iqr r9010 r9050 r7525 r1050 p1 p5 p10 p25 p50 p75 p90 p95 p99.⁴

There is considerable flexibility in the layout of the tables. All tables can be produced using multiple 'vertical' panels if desired. A command like `tabout occupation industry south` will produce two vertical panels (variables `occupation` and `industry`) cross-tabulated against a 'horizontal' variable, `south`. The cell contents can be laid out in columns or rows (for example, frequencies and column percentages alternating, as in: No. % No. % No. %). They can also be laid out in column block or row block mode (for example, frequencies and column percentages in contiguous blocks, as in: No. No. No. % % %).

As well as the contents of the cells, additional information can be placed in the table. The most notable of these inclusions are sample counts (or population estimates),⁵ which can be placed in the far right column, along the bottom of the table or alongside the value labels in the first column. A range of statistics can also be included at the bottom of the table. These consist of: Pearson `chi2`, `gamma`, Cramer's `V`, Kendall's `tau` and the likelihood-ratio `chi2`. Finally, you can also include additional information at the bottom

3. If an `if` or `in` condition is specified with any of the `svy` options, `tabout` makes use of the `subpopulation` option, as recommended in the Stata manual.

4. The `uwsum` is not mainstream Stata. It stands for 'unweighted sum' and is a useful statistic in tables where you to present weighted data, but would like one (or more) columns to contain an unweighted sum of a variable. For example, 'uwsum subpop' can be used to create an 'n' count in the middle of a table of means.

5. The main difference between these two is that the latter are weighted counts, achieved through `tabout`'s `nwt` option.

of the table, such as the source of the data, the population or various notes. Headings and sub-headings for tables can also be placed at the top of the table.

The table below summarises these categories of tables, the kinds of contents allowed and the available layouts. (As mentioned earlier, if you make an error when typing the syntax of `tabout`, the following table is displayed on your screen, alongside a hint as to the nature of your error.)

Type of table	Allowable cell contents	Available layout
Basic	freq cell row col cum any number of above, in any order <i>for example: cells(freq col)</i>	col row cb rb
Basic with SE or CI (turn on <i>svy</i> option)	freq cell row col se ci lb ub only one of: freq cell row col <i>(must come first in the cell)</i> and any number of: se ci lb ub <i>for example: cells(col se lb ub)</i>	col row cb rb
Summary -as a oneway table (turn on <i>sum</i> option; also may need to turn on <i>oneway</i> option)	any number of: N mean var sd skewness kurtosis sum uwsun min max count median iqr r9010 r9050 r7525 r1050 p1 p5 p10 p25 p50 p75 p90 p95 p99 with each followed by variable name <i>for example: cells(min wage mean age)</i>	no options (fixed)
Summary -as a twoway table (turn on <i>sum</i> option)	only one of: N mean var sd skewness kurtosis sum uwsun min max count median iqr r9010 r9050 r7525 r1050 p1 p5 p10 p25 p50 p75 p90 p95 p99 followed by one variable name <i>for example: cells(sum income)</i>	no options (fixed)
Summary with SE or CI (turn on <i>sum</i> option and <i>svy</i> option)	mean followed by one variable name and any number of: se ci lb ub <i>for example: cells(mean weight se ci)</i>	col row cb rb

3 Syntax

`tabout varlist [weight] [if] [in] using [, replace append cells(contents)
format(string) clab(string) layout(layouts) oneway sum
stats(statstypes)`

VARIOUS 'N' OPTIONS:
npos(*positions*) nlab(*string*) nwt(*string*) noc noffset(*string*)

VARIOUS SVY OPTIONS:
svy sebnone cibnone cisep(*string*) ci2col percent level(#)

USER CUSTOMISATION OF LABELS:
total(*string*) ptotal(*totaltype*) h1(*string*) h2(*string*) h3(*string*)

STYLE OPTIONS, MOSTLY FOR LATEX OUTPUT:
style(*styles*) lines(*linetypes*) font(*fontstyles*) bt rotate(#)
cl1(#-#) cl2(#-#) cltr1(*string*) cltr2(*string*)

OPTIONS RELATED TO EXTERNAL FILES:
body topf(*string*) botf(*string*) topstr(*string*) botstr(*string*)
psymbol(*string*) delim(*string*)

MISCELLANEOUS OPTIONS: money(*string*) chkwtnone debug noborder
show(*showtypes*) wide(#)]

where *varlist* is a list of vertical (row) variables, followed by the horizontal (column) variable last. if the *oneway* option is specified, then all the variables are regarded as vertical.

where *contents* consist of: *freq cell row col cum* for basic tables and *N mean var sd skewness kurtosis sum uwsum min max count median iqr r9010 r9050 r7525 r1050 p1 p5 p10 p25 p50 p75 p90 p95 p99* for summary tables. The default is *freq*. When the *svy* option is used, you can also specify *se ci lb ub*.

where *layouts* consist of: *col row cblock rblock*. The default is *col*.

where *positions* consist of: *col row both lab tufte*. The default is *col*.

where *statstypes* consist of: *chi2 gamma V taub lrchi2*, though only *chi2* is available for *svy* tables.

where *totaltypes* consist of: *none single all*. The default is *all*.

where *styles* consist of: *tab tex htm csv*. The default is *tab*.

where *lines* consist of: *single double none*. The default is *single*.

where *fontstyle* consist of: *bold italic*. The default is plain formatting.

where *showtypes* consist of: **none all output**. The default is **output**.

fweights aweights iweights and **pweights** are allowed with **tabout**, depending on the underlying command; see [U] 14.1.6 **weight** and individual entries for **tabulate** and **summarize**. For tables of summary statistics, **iweights** are not allowed, because **tabout** uses the **detail** option in Stata's **summarize** command (which does not allow **iweights**). Note that the **svy** option requires that the data be already **svyset** and an error message reminds you of this if you forget. The weight set by **svyset** will override any other **weight** command you enter **tabout** if you have specified the **svy** option.

Note that **tabout** will work under Stata 9.2 onward.

4 Options

using is required, and indicates the filename for the output. Some applications (particularly MS Excel) 'lock' files when they're open. **tabout** cannot write to these files and consequently issues an error message, suggesting that you check to see if the file is already open in another application.

replace and **append** are file options, and determine whether the current output will overwrite an existing file, or be appended to the end of that file. If you omit **append** or **replace**, **tabout** issues a warning if the file already exists.

cells determines the contents of table cells. As the table on the previous page showed, you can enter any one or more of **freq cell row col cum** in a basic table. They can be in any order. When you choose the **svy** option, you can only have one of these choices, and it must come first. The additional choices which are then available are: **se ci lb ub**. For summary tables, you can have any of the **contents** listed earlier. If you are creating a twoway table, only one summary statistic may go in a cell (eg. **median wage**); if it's a oneway table, any number of statistics (followed by a variable name) may go in the cell (eg. **median wage mean age iqr weight**). When you choose the **svy** option with summary tables, only **mean** is allowed (eg. **mean wage se ci**.)

format indicates the number of decimal points. Unlike mainstream Stata, this option only requires a number. Do not enter '%' or 'f' symbols. You can however, enter **c** for comma, **p** for percentage, and **m** for money (currency) and you can use the **money** option (see below) to specify the currency. For example, you might enter **f(0c 1p 1p 2)** to produce: 1,291 9.2% 10.3% 23.93. The entries should be in the same order as the **cells** order, that is, if **freq** comes first, then **0c** should come first if you want 0 decimal points (with commas) as the format for frequencies. You do not have to type in the same number of format entries as there are cell entries. If you include more, **tabout** ignores them; if you include less, the last **format** entry is repeated for the remaining cell entries.

clab determines the column headings for the third row of the table, that is, the headings just above the data. By default, **tabout** places the 'horizontal' variable's name in the

first row, its value labels in the second row, and an abbreviation for the cell contents (eg. No. Row % etc) in the third row. You can over-ride all of these defaults using the `h1 h2` and `h3` options (see below). Most of the time, however, it will only be the third row which you need to change, so the `clab` option makes this easy for you. Just enter the column titles as you want them to display, without quote marks or other symbols. *However*, you must include underscores between words if there are spaces in the column title, for example `clab(No. Row_% Col_%)`. You do not have to type in the same number of `clab` entries as there are cell entries. If you include more, `tabout` ignores them; if you include less, the last `clab` entry is repeated for the remaining cell entries. For example if your cell entry was `freq col row cum` you could just enter `clab(No. %)` and all but the first column of data would have % symbols at the top.

`layout` determines how the columns will be laid out. They can be in alternating columns (No. % No. % No. %) and alternating rows (No. on the first row, % on the next two, then back to No. and so on). They can be in column blocks, or in row blocks, where the data is kept contiguous, for example: No. No. No. % % %. The exception to this is summary tables where the layout is fixed and you have no choice. (However, an exception to this is the `svy` option, which can be laid out using all of these options. See the earlier table for clarification.)

`oneway` tells `tabout` that the list of variables are all ‘vertical’. Normally, `tabout` assumes that the last variable in the list is the ‘horizontal’ variable, to be used in a twoway cross-tabulation. To override this default behaviour, specify `oneway`.

`sum` tells `tabout` that the table is to be a summary table. Normally, `tabout` assumes that the table will be a basic table and checks to see if the `cells` contents have the correct entries (`freq row col` etc). By telling `tabout` that the table is a summary table, this checking process includes checks for the various summary statistics and the variables in the data set. The `sum` option is essential if you wish to produce a summary table.

`stats` allows you to include additional information based on the various statistics available in `tabulate`. Note that, unlike `tabulate`, `tabout` requires that you enter the full term (and not an abbreviation) and will only allow *one* statistic in a table. You must enter `chi2`, not just `chi`.

`npos` determines where the ‘n’ information will be placed. The various ‘n’ options (`npos nlab nwt nnoc`) provide sample counts for the table. You need only enter one of these options for the ‘n’ to be included. For the options you have not entered, `tabout` places make use of the default values. **A cautionary note:** if you select `npos(row)` and are using multiple panels, the n counts you see at the bottom of the table reflect only those observations included in the bottom panel. They may not be accurate n counts for panels higher in the table, depending on whether there are missing observations in the bottom panel’s vertical variable.

`lab` determines the label for the ‘n’ counts. The default for `col` and `row` positions is a simple uppercase N; for the `lab` position it is `(n=#)` where # stands for number;

and for the `tufte` position it is (%). You can change all of these except the `tufte` position (which is fixed), and if you wish to alter the `lab` position, use the `#` symbol to indicate where the number should go. For example, `npos(lab) nlab(Sample count=#)`. The `npos(tufte)` option provides a convenient way of displaying a percentage breakdown, rather than a count, for the main ‘vertical’ variables. The name comes from the approach adopted by Edward Tufte in his construction of a ‘supertable’, which he designed for the *New York Times* in 1980 (Tufte 2001, p. 179).

`nwt` indicates that the ‘n’ count be weighted by this variable. This can be useful for producing population estimates in a table, rather than just sample counts. Note that `tabout` always uses Stata’s `iweight` option for this weighting.

`nnoc` stands for n-no-comma and turns off the comma in the ‘n’ count. Because `tabout` does not provide a `format` option for ‘n’ counts (decimal points don’t really make sense here), the default behaviour is to include commas. The `nnoc` option over-rides this default behaviour.

`noffset` stands for n offset and determines where the n counts should be placed. The default is 1, which means the n counts will be in the first data column and/or the first data row in a table. Setting `noff(2)` for example, allows you to shift the n counts further along (or down) in the table, into either the second data column or the second data row. If you are using block layouts (`layout(cb)` or `layout(rb)`), the `noffset` option applies to blocks rather than individual columns or rows. The example below makes this clearer.

`svy` tells `tabout` that the `cell` contents include survey output, and so the checking procedure (mentioned earlier) looks for things like `se`, `ci` and so forth. You must turn on `svy` if you wish to include survey output in your table.

`sebnone` stands for se-brackets-none and tells `tabout` to suppress the parentheses which normally surround the standard errors.

`cibnone` stands for ci-brackets-none and tells `tabout` to suppress the square brackets which normally surround the confidence intervals.

`cisep` stands for ci-separator and tells `tabout` to replace the default (which is a comma) by whatever the user enters (for example, a dash).

`ci2col` stands for ci-in-two-columns and tells `tabout` to place the `lb` and `ub` estimates in two columns (as it normally does), and to place a ‘[’ and a ‘,’ in the first column, and a ‘]’ in the second column. This can be useful for layout in a word processor, because the first column can be right aligned (to the comma) and the second column can be left aligned, and it appears that you have a single column for your `ci`, which is neatly aligned according to the commas. Note that if you select `ci` in the `cells`, `tabout` normally places both the lower bound and the upper bound in a single cell and includes brackets and separator. The `ci2col` does not apply in this case. For it to work, you need to specify the upper and lower bound options, for example: `cell(freq lb ub) ci2col`.

`percent` tells `tabout` that the `svy` output should be shown as percentages, not propor-

tions. This follows the default behaviour of `svy:tab`.

`level` specifies the level for the `svy` estimates. The default is 95%.

`total` tells `tabout` what labels to use for totals. The ‘vertical’ total comes first, the ‘horizontal’ second. The default labels for these variables are ‘Total’. If there are spaces in either of the labels which you wish to enter, use underscores. For example, `total(All_persons Total)`.

`ptotal` tells `tabout` how to treat the totals for each panel, when you have multiple panels in a table. The default behaviour is to show all totals, but this can sometimes be repetitive, so you can specify `ptotal(single)` to have a single total row shown at the bottom of the table. You can also turn off all totals with `ptotal(none)`.

`h1` through to `h3` over-ride the default headings for a table. If you choose to use these, there are a couple of requirements. If you have selected either `tex` or `htm` as your output style, you are responsible for all the various code needed. `tabout` does not make *any* adjustments to what you enter, it just outputs it as it finds it. If you have chosen `tab` or `csv` as your output style, you must enter a delimiter to indicate where the columns are in your heading. Unlike the usual `tabout` practice, you do not need to worry about spaces in your titles (no need for underscores!) because this column delimiter takes care of things. However, the number of delimiters must match the number of columns in the table or the headings may be out of alignment. You might enter: `h2(| Very good | Good | Bad | Very bad | Total | N)` and the first column heading would be empty, and the remaining columns would have the appropriate labels. Note that the `npos(col)` option usually places the `nlab` on the `h2` line so you may need to include this yourself in your `h2` label, as in the example just given. To suppress the display of any of these headings, enter ‘nil’ into the appropriate option (for example, `h3(nil)`).

`style` The default is `style(tab)`, which is useful for importing into spreadsheets or word processors. Note that the first row always has the correct number of tabs, even when a single title is involved. This helps other applications parse the table correctly. Note also that the repetition of labels in headings can be easily dealt with by using a ‘merge cells’ command in your spreadsheet or word processor. The `style(csv)` option is useful for importing into spreadsheets (like MS Excel) because it opens immediately as a spreadsheet. Note, however, that some spreadsheets ignore trailing 0s, so this may muck up your neat formatting. To avoid this, export the table from `tabout` as `style(tab)` and use the wizard in your spreadsheet to indicate that all columns are ‘text’ rather than ‘general’.

`lines` indicates how much space (for `style(tab)` and `style(csv)`) or how many lines (for `style(tex)`) should separate tables between panels. The default is `single`.

`font` only applies to `style(tex)` and `style(htm)` and provides bold and italic fonts for the ‘vertical’ variable names and the ‘horizontal’ variable names and value labels. The totals are also given this font. You can also use the `h1` to `h3` options to manually set up fonts for your titles.

bt only applies to users of L^AT_EX, and requires that you have the **booktabs** package installed. This allows the use of the **toprule**, **midrule** and **bottomrule** commands, rather than the usual **hline** command. It produces more pleasing output.

rotate only applies to users of L^AT_EX, and can be used to rotate the ‘horizontal’ variable’s labels through whatever angle is entered in this option. For example, **rotate(60)** produces quite a pleasing effect. You will also need to include the following L^AT_EX code (courtesy of Goossens, Rahtz, and Mittelbach (1997, pp. 48–49)) in your document’s preamble:

```
\newcommand{\rot}[2]{\rule{1em}{0pt}%  
\makebox[0cm][c]{\rotatebox{#1}{\ #2}}}
```

cl1 and **cl2** only apply to users of L^AT_EX, and also requires that you use the **booktabs** package in your L^AT_EX document. These options can be used to place column lines (hence **cl**) between the first and second heading rows, and between the second and third heading rows (hence two sets). You enter the column numbers which you wish to span, separated with a dash. For example, to place a line under the ‘horizontal’ variable’s name, you might enter: **cl1(2–6)** in a table with six columns. If you are entering lines spanning blocks of columns (2-4 5-7), you might need to fine tune the gap between them using **cltr1** and **cltr2**. By default, whenever you specify either of the **cl** options, **tabout** places a small gap (0.75em) between adjacent lines.

cltr1 and **cltr2** stand for column-line-trim, and allow you to specify an amount of trim to be applied to the left side of the **cl1** or **cl2** lines which you have entered. You can specify the amount in whatever acceptable tex measurement you like. For example: **cl2(2-3 4-5 6-7) cltr2(1.5em)**. As just noted, the default amount is 0.75em.

body is used to insert some basic html or L^AT_EX code above and below the table. This allows you to view the table without further coding.

topf and **botf** allow you to insert code stored in files which **tabout** can insert above and below the tables. These are particularly useful for html and L^AT_EX users, and allow you to control the layout of the tables more precisely. All users will find them useful as a way of inserting additional information above and below the table, such as notes, populations, data sources (for the bottom of the table) and titles (for the top of the table).

topstr and **botstr** contain text which you can pass to the **topf** and **botf** files. This text will be inserted into the files where ever the placeholder (default **#**) has been placed. Note that each placeholder must be on a separate line in these files. The strings designated in the **topstr** and **botstr** must be separated with the pipe delimiter (or other user-chosen delimiter) if there is more than one block of text being passed.

psymbol stands for placeholder-symbol and can be any symbol the user chooses. The default is **#** and it provides a ‘placeholder’ in the stored files (the **topf** and **botf**) which **tabout** places above and below the tables.

delimit can be any symbol the user chooses. The default is the pipe delimiter as shown

in the earlier example. It is used to specify columns within the `h1` to `h3` options, and for separating the contents of the `topstr` and `botstr` options. Note, unlike earlier versions of `tabout`, the `delimit` symbol is no longer used for labels. Instead, underscores are used to close-up spaces and parsing is done on the remaining spaces.

`money` indicates the currency to be used if you have chosen the money format. For example, `format(2m) money(£)`. You can enter any symbol that your keyboard allows. For LaTeX users, you can enter any text which LaTeX accepts, though you may need to include quotes.

`chkwtnone` prevents `tabout` from checking the legality of your weights. Stata commands will not allow you to use non-integer frequency weights and `tabout` normally checks for this. You can over-ride this behaviour with the `tt chkwtnone` option. Note that this option does not stop Stata itself from refusing to use non-integer frequency weights.

`debug` shows you most of the underlying Stata commands (though not for summary tables) from which the tables are built. This can be useful for confirming your results.

`noborder` only applies to html output, and determines whether the table and cells should be surrounded by borders. This only applies when the `body` option is turned on.

`show` determines what will be seen on the screen. The `show(all)` option displays the final table output as well as the Mata string matrices which are used to build this final output. The contents of these matrices may not exactly match the final output, in terms of formatting and labelling. The `show(none)` option suppresses all output except for the name of the file to which the table has been exported. The default option is to show the output which has been sent to a file. It may look messy on the screen, but open it in the appropriate application to check it first before panicking.

`wide` is used in conjunction with `show(all)` and specifies the width of the columns in the Mata matrices. The default is 10 spaces. Note that even if you reduce this to a very small number, `tabout` will always increase the width of the columns to accommodate the widest cell entry in the data.

5 Some examples

In the following examples I present some tables based on several of Stata's shipped datasets, though they mainly draw on the `nlsw88.dta` dataset (which is a handy dataset for illustration because it has so many categorical variables. Apologies to those with a more medical inclination to their stats.). All these shipped datasets can be loaded with the `sysuse` command. In the case of the `nlsw88.dta`, a weight variable was also constructed (using a random number multiplied tenfold) so as to demonstrate the `svy` option.

Below each table is shown the syntax which produced it. If you are not a \LaTeX user, the first syntax is the minimum syntax required to get essentially the same table you see on the page, though you will probably need to add some formatting in your spreadsheet or word processor. If you use \LaTeX , the second block of syntax shows the extra commands which were used to produce the tables for this document. The surrounding \LaTeX table commands are not shown, but are found in the top file and bottom file text shown at the end of this tutorial. Finally, for \LaTeX users, the file suffix (`.txt`) in these examples will need to be changed to (`.tex`).

While it might appear from the following examples that the top file and bottom file options are only useful for \LaTeX and html users, this is not the case. All of these examples show the source of the data as a note at the bottom of the table, and this device may be useful to all users. Indeed, encapsulating titles, notes, sources, populations, weighting information, and so forth within the code which produces a table is a very good practice, and is particularly useful for the batch production of tables, where copying such information bit-by-bit is error prone. Anyway, if you are a tab-delimited or csv user, have a look at the code for the contents of these files at the end of this tutorial and you will see how you can also make use of top files and bottom files for including extra information with your tables. Just ignore the \LaTeX verbiage and focus on how the `#` symbol is used.⁶

5.1 Basic tables

While `tabout` is based closely on `tabulate`, it goes a bit beyond it. Not only can you specify the `cells` contents in any order you please—and they will display in that order—but you can also use cumulative percentages inside twoway tables. The following table illustrates this possibility.

6. Note that while the examples in this tutorial just show the use of one argument being passed to a file, you can use multiple arguments. Just add as many `#` symbols as you need. However, make sure each `#` symbol is on a new line in your top and bottom files. Inside your `tabout` syntax, just use the pipe delimiter (or your defined symbol) to separate all the arguments.

To died or exp. end	Patient died								
	No			Yes			Total		
	No.	Col %	Cum %	No.	Col %	Cum %	No.	Col %	Cum %
10 or less months	4	23.5	23.5	15	48.4	48.4	19	39.6	39.6
11 to 20 months	6	35.3	58.8	8	25.8	74.2	14	29.2	68.8
21 to 30 months	2	11.8	70.6	7	22.6	96.8	9	18.8	87.5
31 or more months	5	29.4	100.0	1	3.2	100.0	6	12.5	100.0
Total	17	100.0		31	100.0		48	100.0	

Source: cancer.dta

```

sysuse cancer, clear
la var died "Patient died"
la def ny 0 "No" 1 "Yes", modify
la val died ny
recode studytime (min/10 = 1 "10 or less months") ///
(11/20 = 2 "11 to 20 months") ///
(21/30 = 3 "21 to 30 months") ///
(31/max = 4 "31 or more months") ///
, gen(stime)
la var stime "To died or exp. end"

tabout stime died using table1.txt, ///
c(freq col cum) f(0 1) clab(No. Col_% Cum_%) ///

style(tex) bt c11(2-10) c12(2-4 5-7 8-10) font(bold) ///
topf(top.tex) botf(bot.tex) topstr(14cm) botstr(cancer.dta)

```

After some recoding to improve presentation, this syntax illustrates a number of features of `tabout`. The `format` option only needs two entries, and the third item in the cell contents (the cumulative percentage) is automatically assigned the second format. The underscores are used in the `clab` option to indicate spaces. In the `tex` output, the `top` file and `bottom` file options are used to pass some useful L^AT_EX code to the table.

	College graduate		
	Not college graduate	College graduate	Total
Lives in the south			
Does not live in south	4,347	1,524	5,871
Lives in the south	3,221	914	4,135
Total	7,568	2,438	10,006
Does not live in south	74.0%	26.0%	100.0%
Lives in the south	77.9%	22.1%	100.0%
Total	75.6%	24.4%	100.0%
Does not live in south	57.4%	62.5%	58.7%
Lives in the south	42.6%	37.5%	41.3%
Total	100.0%	100.0%	100.0%
Lives in SMSA			
Non-SMSA	2,346	478	2,824
SMSA	5,222	1,960	7,182
Total	7,568	2,438	10,006
Non-SMSA	83.1%	16.9%	100.0%
SMSA	72.7%	27.3%	100.0%
Total	75.6%	24.4%	100.0%
Non-SMSA	31.0%	19.6%	28.2%
SMSA	69.0%	80.4%	71.8%
Total	100.0%	100.0%	100.0%

Source: nlsw88.dta

Because most documents are in portrait mode, rather than landscape, fitting multiple columns into tables is always a challenge. One answer provided by `tabout` is the row block layout (`layout(rb)`) which makes for efficient use of page space. The underscores are used in `clab` to indicate blanks, and thereby remove redundant titles. This is partly because the `f(1p)` option has added percent symbols to the data and the 100% indicate which are row percentages and which are column percentages.

```

tabout south race smsa coll [iw=wt] using table2.txt, ///
c(freq row col) f(0c 1p 1p) clab(_ _ _) ///
layout(rb) h3(nil)

style(tex) bt font(bold) cl1(2-4) ///
topf(top.tex) botf(bot.tex) topstr(11cm) botstr(nlsw88.dta)

```

	<i>Not college graduate</i>	<i>College graduate</i>	<i>Total</i>	<i>Not college graduate</i>	<i>College graduate</i>	<i>Total</i>	<i>Not college graduate</i>	<i>College graduate</i>	<i>Total</i>
Race									
White	5,320	1,858	7,178	74.1%	25.9%	100.0%	70.3%	76.2%	71.7%
Black	2,169	533	2,702	80.3%	19.7%	100.0%	28.7%	21.9%	27.0%
Other	79	47	126	62.7%	37.3%	100.0%	1.0%	1.9%	1.3%
Total	7,568	2,438	10,006	75.6%	24.4%	100.0%	100.0%	100.0%	100.0%
Lives in the south									
Does not live in south	4,347	1,524	5,871	74.0%	26.0%	100.0%	57.4%	62.5%	58.7%
Lives in the south	3,221	914	4,135	77.9%	22.1%	100.0%	42.6%	37.5%	41.3%
Total	7,568	2,438	10,006	75.6%	24.4%	100.0%	100.0%	100.0%	100.0%
Lives in SMSA									
Non-SMSA	2,346	478	2,824	83.1%	16.9%	100.0%	31.0%	19.6%	28.2%
SMSA	5,222	1,960	7,182	72.7%	27.3%	100.0%	69.0%	80.4%	71.8%
Total	7,568	2,438	10,006	75.6%	24.4%	100.0%	100.0%	100.0%	100.0%
N	1,714	532	2,246						

Source: nlsw88.dta

This table shows how the column block layout `layout(cb)` can be used effectively. It does rely, however, on a `LaTeX` option (label rotation) to fit everything into the limited horizontal space. (Users of word processors and spreadsheets can emulate this manually, using their cell ‘text direction’ menu item.) This table also shows the use of the ‘n’ option, with the sample counts placed at the bottom of the table, using `npos(row)`.

```

tabout race south smsa coll [iw=wt] using table3.txt, ///
c(freq row col) f(0c 1p 1p) layout(cb) h1(nil) h3(nil) npos(row)

style(tex) bt font(bold) rotate(60) ///
topf(top.tex) botf(bot.tex) topstr(15cm) botstr(nlsw88.dta)

```

	<i>Not college graduate</i>	<i>College graduate</i>	<i>Total</i>	<i>Not college graduate</i>	<i>College graduate</i>	<i>Total</i>	<i>Not college graduate</i>	<i>College graduate</i>	<i>Total</i>
Race									
White	5,320	1,858	7,178	74.1%	25.9%	100.0%	70.3%	76.2%	71.7%
Black	2,169	533	2,702	80.3%	19.7%	100.0%	28.7%	21.9%	27.0%
Other	79	47	126	62.7%	37.3%	100.0%	1.0%	1.9%	1.3%
Total	7,568	2,438	10,006	75.6%	24.4%	100.0%	100.0%	100.0%	100.0%
Lives in the south									
Does not live in south	4,347	1,524	5,871	74.0%	26.0%	100.0%	57.4%	62.5%	58.7%
Lives in the south	3,221	914	4,135	77.9%	22.1%	100.0%	42.6%	37.5%	41.3%
Total	7,568	2,438	10,006	75.6%	24.4%	100.0%	100.0%	100.0%	100.0%
Lives in SMSA									
Non-SMSA	2,346	478	2,824	83.1%	16.9%	100.0%	31.0%	19.6%	28.2%
SMSA	5,222	1,960	7,182	72.7%	27.3%	100.0%	69.0%	80.4%	71.8%
Total	7,568	2,438	10,006	75.6%	24.4%	100.0%	100.0%	100.0%	100.0%
N							1,714	532	2,246

Source: nlsw88.dta

This table reproduces the last one, but shows the effect of the `noffset` option. A common layout is frequencies first, then either column or row percentages, so it often makes more sense to ‘line up’ the n counts below the column percentages. The `noffset` option allows you to ‘shift’ the n counts along to line up under a column (or column block) of your choosing. In this example, the `noff(3)` shifts the n counts into the third block. If you weren’t using the `layout(cb)` option and allowed the data to be in alternating columns (eg. freq row col freq row col etc), then the effect of `noff(3)` would be to place the n counts in the third alternating column: blank blank n blank blank n etc. Keep in mind that `noffset` refers to the data columns and rows, ignoring labels and headings (since to be precise, the first column always has labels in it). If you are using the `npos(row)` or `npos(rb)` options, the same principles apply (just read ‘row’ instead of ‘column’ in the above explanation).

```

tabout race south smsa coll [iw=wt] using table3b.txt, ///
c(freq row col) f(0c 1p 1p) layout(cb) h1(nil) h3(nil) npos(row) ///
noff(3)

style(tex) bt font(bold) rotate(60) ///
topf(top.tex) botf(bot.tex) topstr(15cm) botstr(nlsw88.dta)

```

	Race				N
	White Col %	Black Col %	Other Col %	Total Col %	
College graduate					
Not college graduate	74.3	82.3	65.4	76.3	1,714
College graduate	25.7	17.7	34.6	23.7	532
Total	100.0	100.0	100.0	100.0	2,246
Gamma =	-0.1990	ASE =	0.057		
Lives in SMSA					
Non-SMSA	31.3	24.9	26.9	29.6	665
SMSA	68.7	75.1	73.1	70.4	1,581
Total	100.0	100.0	100.0	100.0	2,246
Gamma =	0.1539	ASE =	0.052		
Lives in the south					
Does not live in south	65.4	36.0	88.5	58.1	1,304
Lives in the south	34.6	64.0	11.5	41.9	942
Total	100.0	100.0	100.0	100.0	2,246
Gamma =	0.4834	ASE =	0.037		
N	1,637	583	26	2,246	

Source: nlsw88.dta

As with `tabulate`, `tabout` allows you to include various statistics at the bottom of your tables. Unlike `tabulate`, however, only one statistic can be included with each table. Note the use of the `npos(both)` option here to provide both row count and column counts.

```

tabout coll smsa south race using table4.txt, ///
c(col) f(1) clab(Col_%) stats(gamma) npos(row)

style(tex) bt font(bold) c11(2-5) ///
topf(top.tex) botf(bot.tex) topstr(11cm) botstr(nlsw88.dta)

```

	Geographical location					
	Does not live in south		Lives in the south		Total	
	Col %	Cell %	Col %	Cell %	Col %	Cell %
Occupation						
Professional/technical	15.3	8.9	12.6	5.3	14.2	14.2
Managers/admin	12.8	7.5	10.4	4.3	11.8	11.8
Sales	35.0	20.3	28.9	12.1	32.5	32.5
Clerical/unskilled	5.0	2.9	3.9	1.7	4.6	4.6
Craftsmen	2.2	1.3	2.6	1.1	2.4	2.4
Operatives	9.1	5.3	13.7	5.7	11.0	11.0
Transport	0.8	0.5	1.8	0.8	1.3	1.3
Laborers	11.5	6.7	14.6	6.1	12.8	12.8
Farmers	0.0	0.0	0.1	0.0	0.0	0.0
Farm laborers	0.2	0.1	0.7	0.3	0.4	0.4
Service	0.2	0.1	1.4	0.6	0.7	0.7
Household workers	0.0	0.0	0.2	0.1	0.1	0.1
Other	7.8	4.6	9.1	3.8	8.4	8.4
Total	100.0	58.1	100.0	41.9	100.0	100.0
Industry						
Ag/Forestry/Fisheries	0.7	0.4	0.9	0.4	0.8	0.8
Mining	0.2	0.1	0.1	0.0	0.2	0.2
Construction	0.8	0.5	1.9	0.8	1.3	1.3
Manufacturing	15.5	9.0	17.8	7.4	16.4	16.4
Transport/Comm/Utility	5.0	2.9	2.7	1.1	4.0	4.0
Wholesale/Retail Trade	14.8	8.6	15.1	6.3	14.9	14.9
Finance/Ins/Real Estate	9.6	5.6	7.2	3.0	8.6	8.6
Business/Repair Svc	4.4	2.6	3.1	1.3	3.9	3.9
Personal Services	3.7	2.2	5.2	2.2	4.3	4.3
Entertainment/Rec Svc	0.5	0.3	1.1	0.4	0.8	0.8
Professional Services	37.4	21.8	36.2	15.1	36.9	36.9
Public Administration	7.2	4.2	8.8	3.7	7.9	7.9
Total	100.0	58.2	100.0	41.8	100.0	100.0
Sample size	1,298		934		2,232	

Source: nlsw88.dta

This table shows how you can change the presentation of labels by temporarily changing the variable label. Note also the use of the `nlab` option to change the default label. Also, notice that underscores are used to indicate spaces in the `clab` option.

```

la var south "Geographical location"

tabout occupation industry south using table5.txt, ///
c(col cell) f(1) clab(Col_% Cell_%) npos(row) nlab(Sample size) ///

style(tex) bt font(bold) c11(2-7) c12(2-3 4-5 6-7) ///
topf(top.tex) botf(bot.tex) topstr(14cm) botstr(nlsw88.dta)

```

	Geographical location					
	Outside south	In south	Total	Outside south	In south	Total
	Column percentages			Cell percentages		
Occupation						
Professional/technical	15.3	12.6	14.2	8.9	5.3	14.2
Managers/admin	12.8	10.4	11.8	7.5	4.3	11.8
Sales	35.0	28.9	32.5	20.3	12.1	32.5
Clerical/unskilled	5.0	3.9	4.6	2.9	1.7	4.6
Craftsmen	2.2	2.6	2.4	1.3	1.1	2.4
Operatives	9.1	13.7	11.0	5.3	5.7	11.0
Transport	0.8	1.8	1.3	0.5	0.8	1.3
Laborers	11.5	14.6	12.8	6.7	6.1	12.8
Farmers	0.0	0.1	0.0	0.0	0.0	0.0
Farm laborers	0.2	0.7	0.4	0.1	0.3	0.4
Service	0.2	1.4	0.7	0.1	0.6	0.7
Household workers	0.0	0.2	0.1	0.0	0.1	0.1
Other	7.8	9.1	8.4	4.6	3.8	8.4
Total	100.0	100.0	100.0	58.1	41.9	100.0
Industry						
Ag/Forestry/Fisheries	0.7	0.9	0.8	0.4	0.4	0.8
Mining	0.2	0.1	0.2	0.1	0.0	0.2
Construction	0.8	1.9	1.3	0.5	0.8	1.3
Manufacturing	15.5	17.8	16.4	9.0	7.4	16.4
Transport/Comm/Utility	5.0	2.7	4.0	2.9	1.1	4.0
Wholesale/Retail Trade	14.8	15.1	14.9	8.6	6.3	14.9
Finance/Ins/Real Estate	9.6	7.2	8.6	5.6	3.0	8.6
Business/Repair Svc	4.4	3.1	3.9	2.6	1.3	3.9
Personal Services	3.7	5.2	4.3	2.2	2.2	4.3
Entertainment/Rec Svc	0.5	1.1	0.8	0.3	0.4	0.8
Professional Services	37.4	36.2	36.9	21.8	15.1	36.9
Public Administration	7.2	8.8	7.9	4.2	3.7	7.9
Total	100.0	100.0	100.0	58.2	41.8	100.0
Sample size	1,298	934	2,232			

Source: nlsw88.dta

While the previous table looks neat, cell percentages are more easily grasped as a block, so this table duplicates the last, but changes the layout to column block (layout(cb)).

```

la def south 0 "Outside south" 1 "In south", modify
la val south south

tabout occupation industry south using table6.txt, ///
c(col cell) f(1) npos(row) nlab(Sample size) ///
layout(cb) ///

style(tex) bt font(bold) c11(2-7) c12(2-4 5-7) ///
h3(& \multicolumn{3}{c}{Column percentages} & ///
\multicolumn{3}{c}{Cell percentages} \\\) ///
topf(top.tex) botf(bot.tex) topstr(13cm) botstr(nlsw88.dta)

```

5.2 Basic tables with survey data

	Lives in the south				
	Does not live in south		Lives in the south		Total
	Row %	95% CI	Row %	95% CI	Row %
College graduate					
Not college graduate (n=1,714)	57.4	[54.6,60.2]	42.6	[39.8,45.4]	100.0
College graduate (n=532)	62.5	[57.6,67.2]	37.5	[32.8,42.4]	100.0
Total (n=2,246)	58.7	[56.2,61.1]	41.3	[38.9,43.8]	100.0
Pearson: Uncorrected chi2(1) =	4.3897				
Design-based F(1.00, 2245.00) =	3.1109	Pr =	0.078		
Race					
White (n=1,637)	66.5	[63.7,69.1]	33.5	[30.9,36.3]	100.0
Black (n=583)	36.5	[32.1,41.2]	63.5	[58.8,67.9]	100.0
Other (n=26)	90.5	[70.7,97.4]	9.5	[2.6,29.3]	100.0
Total (n=2,246)	58.7	[56.2,61.1]	41.3	[38.9,43.8]	100.0
Pearson: Uncorrected chi2(2) =	174.6776				
Design-based F(1.98, 4454.21) =	66.8151	Pr =	0.000		
Lives in SMSA					
Non-SMSA (n=665)	46.0	[41.5,50.6]	54.0	[49.4,58.5]	100.0
SMSA (n=1,581)	63.6	[60.8,66.4]	36.4	[33.6,39.2]	100.0
Total (n=2,246)	58.7	[56.2,61.1]	41.3	[38.9,43.8]	100.0
Pearson: Uncorrected chi2(1) =	58.1985				
Design-based F(1.00, 2245.00) =	41.5602	Pr =	0.000		

Source: nlsw88.dta

When it comes to survey data, confidence intervals are easily handled by `tabout`. The `cells(row ci)` option indicates that CIs are required, and the default settings include square brackets and a comma separator (though the former can be removed and the latter modified using `nocib` and `cisep()`). The `percent` option also turns proportions into percentages and the survey chi2 results are also included. Note the use of the `npos(lab)` option to present n counts within the value labels of the vertical variables.

```

tabout coll race smsa south using table7.txt, ///
c(row ci) f(1 1) clab(Row_% 95%_CI) svy stats(chi2) ///
npos(lab) per

style(tex) bt font(bold) cl1(2-6) ///
topf(top.tex) botf(bot.tex) topstr(14cm) botstr(nlsw88.dta)

```

<i>Average wages according to location</i>									
	<i>Does not live in south</i>			<i>Lives in the south</i>			<i>Total</i>		
	Mean	LB	UB	Mean	LB	UB	Mean	LB	UB
<i>Lives in SMSA</i>									
Non-SMSA	\$6.50	\$5.97	\$7.02	\$5.68	\$5.12	\$6.25	\$6.06	\$5.67	\$6.45
SMSA	\$8.93	\$8.49	\$9.36	\$7.62	\$7.14	\$8.10	\$8.45	\$8.12	\$8.78
<i>Total</i>	\$8.39	\$8.02	\$8.75	\$6.91	\$6.53	\$7.28	\$7.78	\$7.51	\$8.04
<i>Race</i>									
White	\$8.31	\$7.91	\$8.71	\$7.42	\$6.92	\$7.92	\$8.01	\$7.70	\$8.33
Black	\$8.67	\$7.70	\$9.63	\$6.18	\$5.62	\$6.75	\$7.09	\$6.58	\$7.61
Other	\$9.13	\$6.47	\$11.79	\$7.25	\$1.65	\$12.85	\$8.95	\$6.48	\$11.42
<i>Total</i>	\$8.39	\$8.02	\$8.75	\$6.91	\$6.53	\$7.28	\$7.78	\$7.51	\$8.04

Source: nlsw88.dta

As well as a combined CI, `tabout` also allows for separate LB and UB estimates, as shown here. This example also illustrates the money format (`f(2m)`). Currencies other than the \$ can be specified using the `money()` option. In comparison to the earlier approach of relabelling the variable, in this example the `h1` option is used to change the default label. In the case of the `tex` output, the user takes responsibility for all of the L^AT_EX code needed for this heading.

```

tabout smsa race south using table8.txt, ///
c(mean wage lb ub) f(2m) svy sum ///

style(tex) bt font(italic) c11(2-10) c12(2-4 5-7 8-10) ///
h1(& \multicolumn{9}{c}{\emph{Average wages according to location}} \\\) ///
topf(top.tex) botf(bot.tex) topstr(12cm) botstr(nlsw88.dta) ///

```

5.3 Summary tables

Summary tables in `tabout` can be as simple as the following table, where two variables (`inc` and `candidat`) are cross-tabulated and the cell contents are based on the mean of another variable (`pfrac`). This is essentially the same as Stata's `table` command. Note that the `sum` option is required to indicate that this is a summary table.

Family Income	Candidate voted for, 1992			
	Clinton %	Bush %	Perot %	Total %
<\$15k	8.3	3.2	2.5	4.7
\$15-30k	10.8	8.4	4.8	8.0
\$30-50k	12.3	11.4	6.3	10.0
\$50-75k	8.0	8.4	3.6	6.7
\$75k+	4.7	6.2	2.1	4.3
Total	8.8	7.5	3.9	6.7

Source: voter.dta

```
tabout inc candidat using table9.txt, ///
c(mean pfrac) f(1) clab(%) sum ///

style(tex) bt font(bold) cl1(2-5) ///
topf(top.tex) botf(bot.tex) topstr(12cm) botstr(voter.dta)
```

Repair Record 1978	Inter-quartile range of weight			N
	Domestic	Foreign	Total	
One repair	740		740	2
Two repairs	685		685	8
Three repairs	630	300	790	30
Four repairs	370	390	1,710	18
Five repairs	320	620	630	11
Total	730	390	1,430	69
N	48	21	69	

Source: auto.dta

This table shows another example. `tabout` works out that this is a twoway table and uses the last variable in the list (`foreign`) as the 'horizontal' variable.

```
tabout rep78 foreign using table10.txt, ///
c(iqr weight) f(0c) sum h3(nil) npos(both)

style(tex) bt font(bold) cl1(2-4) cltr1(.5em) ///
h1(& \multicolumn{3}{c}{\textbf{Inter-quartile range of weight}} \\\) ///
topf(top.tex) botf(bot.tex) topstr(10cm) botstr(auto.dta)
```

	Mean			Median	
	MPG	Weight (lbs)	Length (in)	Price	Headroom (in)
Car type					
Domestic (70%)	19.8	3,317.1	196.1	\$4,782.50	3.5
Foreign (29%)	24.8	2,315.9	168.5	\$5,759.00	2.5
Total (100%)	21.3	3,019.5	187.9	\$5,006.50	3.0
Repair Record 1978					
One repair (2%)	21.0	3,100.0	189.0	\$4,564.50	1.8
Two repairs (11%)	19.1	3,353.8	199.4	\$4,638.00	3.8
Three repairs (43%)	19.4	3,299.0	194.0	\$4,741.00	3.5
Four repairs (26%)	21.7	2,870.0	184.8	\$5,751.50	3.0
Five repairs (15%)	27.4	2,322.7	170.2	\$5,397.00	2.5
Total (100%)	21.3	3,032.0	188.3	\$5,079.00	3.0

Source: auto.dta

This table illustrates a oneway summary table, but it is not necessary to specify `oneway` because `tabout` works this out from the `cells` contents. It is essential, however, to include the `sum` option to indicate that this is a summary table. The `clab` option also shows the use of underscores to indicate spaces. Finally, the `npos(tufte)` option is shown.

```

tabout foreign rep78 using table11.txt, ///
c(mean mpg mean weight mean length median price median headroom) ///
f(1c 1c 1c 2cm 1c) ///
clab(MPG Weight_(lbs) Length_(in) Price Headroom_(in)) ///
sum npos(tufte)

style(tex) bt cl2(2-4 5-6) cltr2(.75em 1.5em) ///
topf(top.tex) botf(bot.tex) topstr(10cm) botstr(auto.dta)

```

5.4 Summary tables with survey data

	College graduate					
	Not college graduate		College graduate		Total	
	Mean wage	SE	Mean wage	SE	Mean wage	SE
Occupation						
Professional/technical (n=317)	9.64	(0.50)	12.59	(0.68)	11.08	(0.43)
Managers/admin (n=264)	9.22	(0.50)	13.47	(0.94)	10.38	(0.46)
Sales (n=726)	7.01	(0.24)	7.76	(0.46)	7.10	(0.22)
Clerical/unskilled (n=102)	8.44	(1.09)	8.72	(2.05)	8.49	(0.96)
Craftsmen (n=53)	6.24	(0.48)	10.77	(0.60)	6.53	(0.48)
Operatives (n=246)	5.57	(0.19)	4.60	(1.64)	5.55	(0.19)
Transport (n=28)	2.90	(0.24)			2.90	(0.24)
Laborers (n=286)	4.60	(0.18)	6.76	(0.93)	4.73	(0.18)
Farmers (n=1)			8.05	(0.00)	8.05	(0.00)
Farm laborers (n=9)	3.06	(0.50)	2.51	(0.00)	2.96	(0.42)
Service (n=16)	6.48	(0.72)	4.03	(0.00)	6.40	(0.71)
Household workers (n=2)	6.34	(0.15)			6.34	(0.15)
Other (n=187)	4.48	(0.40)	9.65	(0.38)	9.22	(0.37)
Total (n=2,237)	6.88	(0.14)	10.56	(0.31)	7.79	(0.14)
Lives in the south						
Does not live in south (n=1,304)	7.51	(0.20)	10.88	(0.41)	8.39	(0.19)
Lives in the south (n=942)	6.02	(0.19)	10.03	(0.46)	6.91	(0.19)
Total (n=2,246)	6.88	(0.14)	10.56	(0.31)	7.78	(0.14)
Race						
White (n=1,637)	7.22	(0.18)	10.28	(0.32)	8.01	(0.16)
Black (n=583)	6.02	(0.22)	11.45	(0.84)	7.09	(0.26)
Other (n=26)	7.40	(0.87)	11.57	(2.81)	8.95	(1.26)
Total (n=2,246)	6.88	(0.14)	10.56	(0.31)	7.78	(0.14)

Source: nlsw88.dta

As mentioned earlier, you can include standard errors and confidence intervals in your summary tables. You are, however, restricted to a single measure: the mean. (`tabout` uses Stata's `svy:mean` command). This table illustrates one approach to presenting standard errors. Note that you must include both the `sum` option and the `svy` option for tables like these.

```

tabout occ south race coll using table12.txt, ///
c(mean wage se) f(2 2) clab(Mean_wage SE) ///
sum svy npos(lab)

style(tex) bt cll(2-7) font(bold) ///
topf(top.tex) botf(bot.tex) topstr(14cm)

```

	College graduate		
	Not college graduate	College graduate	Total
Average wage			
Lives in the south			
No (n=1,304)	7.51	10.88	8.39
(SE)	(0.20)	(0.41)	(0.19)
(90% CI)	[7.19,7.84]	[10.21,11.56]	[8.08,8.70]
Yes (n=942)	6.02	10.03	6.91
(SE)	(0.19)	(0.46)	(0.19)
(90% CI)	[5.70,6.34]	[9.27,10.79]	[6.59,7.22]
Total (n=2,246)	6.88	10.56	7.78
(SE)	(0.14)	(0.31)	(0.14)
(90% CI)	[6.65,7.11]	[10.05,11.07]	[7.55,8.00]
Race			
White (n=1,637)	7.22	10.28	8.01
(SE)	(0.18)	(0.32)	(0.16)
(90% CI)	[6.93,7.51]	[9.76,10.80]	[7.75,8.28]
Black (n=583)	6.02	11.45	7.09
(SE)	(0.22)	(0.84)	(0.26)
(90% CI)	[5.66,6.38]	[10.07,12.84]	[6.66,7.53]
Other (n=26)	7.40	11.57	8.95
(SE)	(0.87)	(2.81)	(1.26)
(90% CI)	[5.97,8.82]	[6.94,16.19]	[6.88,11.02]
Total (n=2,246)	6.88	10.56	7.78
(SE)	(0.14)	(0.31)	(0.14)
(90% CI)	[6.65,7.11]	[10.05,11.07]	[7.55,8.00]

Source: nlsw88.dta

This table shows similar data, but with the layout designated as row. Note the use of the first underscore in `c1ab(_ (SE) (90%_CI))` to indicate an empty label. This suits row format where the value labels for the ‘vertical’ variables occupy the main part of the first row. Notice also the use of the `h3` option to place useful information above the data in the table. Normally, this line would be occupied by the SE and CI information, but the row layout leaves this line blank. You can either issue a `h3(nil)` to close up this blank line, or you can insert something useful. In the syntax below, the `h3` line is shown in two versions: the first one for tab-delimited output and the second for tex output.

```

tabout south race coll using table13.txt, ///
c(mean wage se ci) f(2 2) sum svy npos(lab) layout(row) ///
level(90) c1ab(_ (SE) (90%_CI)) ///
h3( | Average wage | Average wage | Average wage)

style(tex) bt c11(2-4) c12(2-4) font(bold) ///
h3(& \multicolumn{3}{c}{Average wage} \\\) ///
topf(top.tex) botf(bot.tex) topstr(10cm) botstr(nlsw88.dta)

```

	College graduate		
	Not college graduate	College graduate	Total
Average wage			
Lives in the south			
No (Sample size = 1,304)	7.51	10.88	8.39
Lower bound	7.19	10.21	8.08
Upper bound	7.84	11.56	8.70
Yes (Sample size = 942)	6.02	10.03	6.91
Lower bound	5.70	9.27	6.59
Upper bound	6.34	10.79	7.22
Total (Sample size = 2,246)	6.88	10.56	7.78
Lower bound	6.65	10.05	7.55
Upper bound	7.11	11.07	8.00
Race			
White (Sample size = 1,637)	7.22	10.28	8.01
Lower bound	6.93	9.76	7.75
Upper bound	7.51	10.80	8.28
Black (Sample size = 583)	6.02	11.45	7.09
Lower bound	5.66	10.07	6.66
Upper bound	6.38	12.84	7.53
Other (Sample size = 26)	7.40	11.57	8.95
Lower bound	5.97	6.94	6.88
Upper bound	8.82	16.19	11.02
Total (Sample size = 2,246)	6.88	10.56	7.78
Lower bound	6.65	10.05	7.55
Upper bound	7.11	11.07	8.00

Source: nlsw88.dta

You can also imitate the `vertical` option in `svy:tab` by specifying `lb` and `ub` in your `cells` content. Also, notice the use of the `nlab` option and how the position of the number is indicated with a `#` symbol. Again, two versions for `h3` are shown. The tab-delimited version here differs from that in the last table. Deciding which to use depends on how you use your word processor or spreadsheet when it comes to merging cells which span columns. Whichever version you prefer, it is essential that you place the pipe symbols (or whatever you have defined as your delimiter) to indicate columns.

```

tabout south race coll using table14.txt, ///
c(mean wage lb ub) f(2 2) sum svy ///
npos(lab) nlab((Sample size = #)) ///
layout(row) level(90) clab(_ Lower_bound Upper_bound) ///
h3( | | Average wage | )

style(tex) bt c11(2-4) c12(2-4) font(bold) ///
h3(& \multicolumn{3}{c}{Average wage} \\\) ///
topf(top.tex) botf(bot.tex) topstr(11cm) botstr(nlsw88.dta)

```


7 Appendix: an ancillary program figout

7.1 Overview

figout allows you to take a table like this (produced by tabout):

	Lives in the south		Total %
	Does not live in south %	Lives in the south %	
College graduate			
Not college graduate	57.4	42.6	100.0
College graduate	62.5	37.5	100.0
Race			
White	66.5	33.5	100.0
Black	36.5	63.5	100.0
Other	90.5	9.5	100.0
Lives in SMSA			
Non-SMSA	46.0	54.0	100.0
SMSA	63.6	36.4	100.0
Total	58.7	41.3	100.0

Source: nlsw88.dta

and then automatically convert a panel from it into a graph like this:

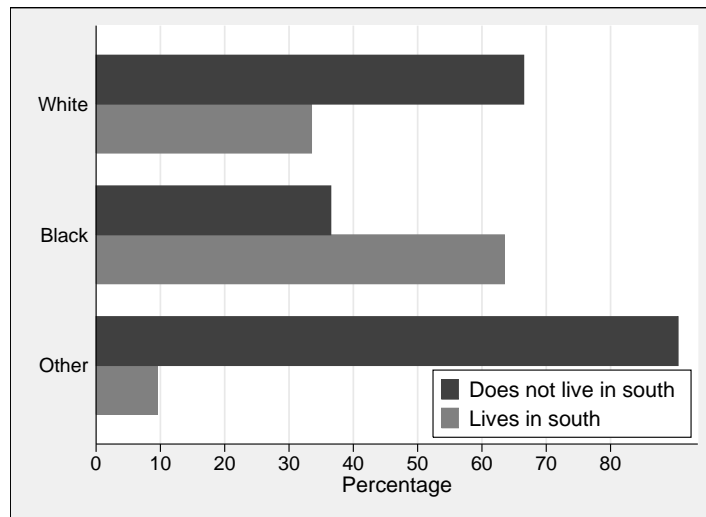


Figure 1: Example of figout


```

\bottomrule
\addlinespace[.75ex]
\scriptsize{\emph{Source: }nlsw88.dta}
\end{tabularx}
\normalsize
\end{center}

```

Finally, here is the mini dataset which is produced by `figout`. This is the screen shot which is presented to the user when `figout` concludes successfully:

```

(5 vars, 3 obs)
file fig_fig.dta saved

+-----+
| race  not_so~h  south  order |
+-----+
1. | White      66.5   33.5    1 |
2. | Black      36.5   63.5    2 |
3. | Other      90.5    9.5    3 |
+-----+

```

Essentially, the process is this: `figout` reads the output file looking for a start word or phrase (in this example, it was 'Race'). It then extracts the numbers it finds until it reaches the stop word or phrase (in this example, is was '`\midrule`'). It only extracts the number of columns for which you indicate `gvars`, which conveniently avoids extracting the totals column. Each line in the panel becomes the basis for the `over` option in the subsequent graph. `figout` then loads those numbers into a mini dataset and saves it under the name you specify. If `figout` fails to find either the start word or the stop word, no mini dataset is produced and you are issued a warning.

Once the mini dataset is created, it is an easy matter to create a `Stata` graph. The data is in a form suitable for graph's `over` option, and `figout` automatically creates an order variable for you, to preserve the same order that was used in your original table.

While `figout` works fine with text files (such as `tab-delimited` files) it is ideally suited to batch processing with `LATEX`. Here is a typical sequence. First, the pseudo-code for your `Stata` do file:

```

use mydata, clear
tabout etc using table1
tabout etc using table2
etc
tabout etc using table99
clear

figout using fig1, infile(table1.tex)
etc
grexportpdf using fig1

(and repeat for as many graphs as you need)

```

Inside your `LATEX` file you would probably do something like this:

```

\documentclass{report}
etc
\begin{document}

```

```

\frontmatter
etc
etc
\mainmatter
Some discussion of your results etc referring to
both table 1 in the appendix for full results and
figure 1 in the following text for key findings.
\input{fig1}
etc
etc
\input{fig2}
etc
etc
\endmatter
\appendix
\input{table1}
\input{table2}
\end{document}

```

Essentially what happens here is that you open your dataset, run all your `tabout` commands to produce a complete set of tables. You then close your dataset (since you are about to create a whole set of mini datasets) and run all your `figout` commands (alternating with your graph commands). `figout` extracts the appropriate blocks of numbers from the tables specified, and saves them in a set of mini datasets which becomes the basis for producing graphs. The key points to remember are: run all your tables first, because this avoids the need to open and close your main dataset. Then close the dataset and run your `figout` command alternating with your `graph` commands. The advantage of a system like this is that you can change your original dataset, perhaps through the addition of new data or a different weight, rerun all your tables in a batch file and have the graphs which depend on those tables updated as well. Everything remains in sync.

Note that `figout` is a Stata Version 8.2 program, so it will work with the current version of `tabout` as well as the earlier version.

7.2 Syntax

```

figout using , infile(string) gvars(string) over(string) start(string)
stop(string) [replace ]

```

7.3 Options

`using` is required, and indicates the filename for the output of the mini Stata dataset.

Note that you do not need to add the `dta` filename extension.

`infile` is required and is the name of the output file produced by `tabout`, for example, `table1.tex`. Note that you do need to add the filename extension because you may be using `figout` with any number of file types (L^AT_EX, csv, or tab-delimited).

gvars are required and are names you wish to assign to your graph variables, and they need to match a contiguous block of cells in your table. They are basically the categories of the ‘horizontal’ variable in your table. (See the example above). Note that this allows you to leave out total columns (since these are rarely used in graphs).

over is required is the name of the graph variable to be used by the **over** option in the **graph** command. It is one the panels in your table, and basically matches one of your ‘vertical’ variables.

start is required and is a unique word or phrase on the line above the block of cells. It can usually refer to the panel title in a **tabout** table, unless the title is repeated in another panel.

stop is required and is a unique word or phrase on the line beneath the block of cells. In the case of L^AT_EX, you can just use `\midrule` since this generally indicates the end of a panel if you are using the **ptotal(single)** option.

replace is optional and follows usual Stata convention and prevents you accidentally over-writing an existing Stata dataset with your new mini dataset. If you are confident that there are no other datasets with the same name, you can use the **replace** option and this makes it more convenient if you need to develop your **figout** code using several attempts.

8 Acknowledgements

Numerous people have provided feedback and advice over the last two years and I am very grateful for their comments. In particular I’d like to thank: JP Azevedo, Megan Blaxland, Simon Coulombe, Enzo Coviello, Nick Cox, Bill Gould, Daniel Hoechle, Ben Jann, Stephen Jenkins, Rafael Martins de Souza, Scott Merryman, Benjamin Schirge, Urvi Shah, Tim Stegmann, Herve Stolowy, Amanda Tzy-Chyi Yu and Chris Wallace.

About the author

Ian Watson is a freelance researcher and Visiting Senior Research Fellow at Macquarie University, Sydney, Australia.

9 References

- Fear, S. 2003. Publication Quality Tables in L^AT_EX. *Booktabs Documentation* .
www.ctan.org/tex-archive/macros/latex/contrib/booktabs/booktabs.pdf
- Goossens, M., S. Rahtz, and F. Mittelbach. 1997. *The L^AT_EX Graphics Companion: Illustrating documents with T_EX and Postscript*. Boston: Addison-Wesley.
- Tufte, E. R. 2001. *The Visual Display of Quantitative Information*. Cheshire, Connecticut: Graphics Press.