Title

stata.com

table — Flexible table of summary statistics

Description

`table` calculates and displays tables of statistics.

Quick start

Table of number of nonmissing observations in each category of `catvar1` by `catvar2`

```
table catvar1 catvar2
```

Add supercolumns defined by `catvar3` and superrows defined by `catvar4`

```
table catvar1 catvar2 catvar3, by(catvar4)
```

Table of mean and SD of `v1` for each combination of `catvar1` and `catvar2`

```
table catvar1 catvar2, contents(mean v1 sd v1)
```

Add total row showing mean and SD of `v1` for each value of `catvar2`

```
table catvar1 catvar2, contents(mean v1 sd v1) row
```

Add total column showing mean and SD of `v1` for each value of `catvar1`

```
table catvar1 catvar2, contents(mean v1 sd v1) by(catvar3) row col
```

25th, 50th, and 75th percentiles of `v1` for each level of `catvar1` displayed using the `%7.1fc` format

```
table catvar1, contents(p25 v1 p50 v1 p75 v1) format(%7.1fc)
```

Means of `v1`, `v2`, and `v3` for each level of `catvar2` centered in columns of width 10

```
table catvar2, contents(mean v1 mean v2 mean v3) cellwidth(10) center
```

Menu

Statistics > Summaries, tables, and tests > Other tables > Flexible table of summary statistics

Syntax

```
table rowvar [colvar [supercolvar]] [if] [in] [weight] [, options]
```

options	Description
Main	
`contents(`*clist*`)`	contents of table cells; select up to five statistics; default is `contents(freq)`
`by(`*superrowvarlist*`)`	superrow variables
Options	
`cellwidth(#)`	cell width
`csepwidth(#)`	column-separation width
`stubwidth(#)`	stub width
`scsepwidth(#)`	supercolumn-separation width
`center`	center-align table cells; default is right-align
`left`	left-align table cells; default is right-align
`cw`	perform casewise deletion
`row`	add row totals
`column`	add column totals
`scolumn`	add supercolumn totals
`concise`	suppress rows with all missing entries
`missing`	show missing statistics with period
`replace`	replace current data with table statistics
`name(`*string*`)`	name new variables with prefix *string*
`format(%`*fmt*`)`	display format for numbers in cells; default is `format(%9.0g)`

`by` is allowed; see [D] **by**.

`fweights`, `iweights`, and `pweights` are allowed; see [U] **11.1.6 weight**. `pweights` may not be used with `sd`, `semean`, `sebinomial`, or `sepoisson`. `iweights` may not be used with `semean`, `sebinomial`, or `sepoisson`.

where the elements of *clist* may be

`freq`	frequency	`n` *varname*	same as `count`
`mean` *varname*	mean of *varname*	`max` *varname*	maximum
`sd` *varname*	standard deviation	`min` *varname*	minimum
`semean` *varname*	standard error of the mean (`sd/sqrt(n)`)	`median` *varname*	median
		`p1` *varname*	1st percentile
`sebinomial` *varname*	standard error of the mean, binomial distribution (`sqrt(p(1-p)/n)`)	`p2` *varname*	2nd percentile
		`...`	3rd–49th percentiles
`sepoisson` *varname*	standard error of the mean, Poisson distribution (`sqrt(mean)`)	`p50` *varname*	50th percentile (`median`)
		`...`	51st–97th percentiles
`sum` *varname*	sum	`p98` *varname*	98th percentile
`rawsum` *varname*	sums ignoring optionally specified weight	`p99` *varname*	99th percentile
`count` *varname*	count of nonmissing observations	`iqr` *varname*	interquartile range

Rows, columns, supercolumns, and superrows are thus defined as

row 1	.
row 2	.

	supercol 1		supercol 2	
	col 1	col 2	col 1	col 2
row 1	.	.	.	.
row 2	.	.	.	.

	col 1	col 2
row 1	.	.
row 2	.	.

	supercol 1		supercol 2	
	col 1	col 2	col 1	col 2
superrow 1:				
row 1	.	.	.	.
row 2	.	.	.	.
superrow 2:				
row 1	.	.	.	.
row 2	.	.	.	.

Options

Main

`contents(`*clist*`)` specifies the contents of the table's cells; if not specified, `contents(freq)` is used by default. `contents(freq)` produces a table of frequencies. `contents(mean mpg)` produces a table of the means of variable `mpg`. `contents(freq mean mpg sd mpg)` produces a table of frequencies together with the mean and standard deviation of variable `mpg`. Up to five statistics may be specified.

`by(`*superrowvarlist*`)` specifies that numeric or string variables be treated as superrows. Up to four variables may be specified in *superrowvarlist*. The `by()` option may be specified with the `by` prefix.

Options

`cellwidth(#)` specifies the width of the cell in units of digit widths; 10 means the space occupied by 10 digits, which is `0123456789`. The default `cellwidth()` is not a fixed number, but a number chosen by `table` to spread the table out while presenting a reasonable number of columns across the page.

`csepwidth(#)` specifies the separation between columns in units of digit widths. The default is not a fixed number, but a number chosen by `table` according to what it thinks looks best.

`stubwidth(#)` specifies the width, in units of digit widths, to be allocated to the left stub of the table. The default is not a fixed number, but a number chosen by `table` according to what it thinks looks best.

`scsepwidth(#)` specifies the separation between supercolumns in units of digit widths. The default is not a fixed number, but a number chosen by `table` to present the results best.

`center` specifies that results be centered in the table's cells. The default is to right-align results. For centering to work well, you typically need to specify a display format as well. `center format(%9.2f)` is popular.

`left` specifies that column labels be left-aligned. The default is to right-align column labels to distinguish them from supercolumn labels, which are left-aligned.

`cw` specifies casewise deletion. If `cw` is not specified, all observations possible are used to calculate each of the specified statistics. `cw` is relevant only when you request a table containing statistics on multiple variables. For instance, `contents(mean mpg mean weight)` would produce a table reporting the means of variables `mpg` and `weight`. Consider an observation in which `mpg` is known but `weight` is missing. By default, that observation will be used in the calculation of the mean of `mpg`. If you specify `cw`, the observation will be excluded in the calculation of the means of both `mpg` and `weight`.

`row` specifies that a row be added to the table reflecting the total across the rows.

`column` specifies that a column be added to the table reflecting the total across columns.

`scolumn` specifies that a supercolumn be added to the table reflecting the total across supercolumns.

`concise` specifies that rows with all missing entries not be displayed.

`missing` specifies that missing statistics be shown in the table as periods (Stata's missing-value indicator). The default is that missing entries be left blank.

`replace` specifies that the data in memory be replaced with data containing 1 observation per cell (row, column, supercolumn, and superrow) and with variables containing the statistics designated in `contents()`.

This option is rarely specified. If you do not specify this option, the data in memory remain unchanged.

If you do specify this option, the first statistic will be named `table1`, the second `table2`, and so on. For instance, if `contents(mean mpg sd mpg)` was specified, the means of `mpg` would be in variable `table1` and the standard deviations in `table2`.

`name(`*string*`)` is relevant only if you specify `replace`. `name()` allows changing the default stub name that `replace` uses to name the new variables associated with the statistics. If you specify `name(stat)`, the first statistic will be placed in variable `stat1`, the second in `stat2`, and so on.

`format(%`*fmt*`)` specifies the display format for presenting numbers in the table's cells. `format(%9.0g)` is the default; `format(%9.2f)` and `format(%9.2fc)` are popular alternatives. The width of the format you specify does not matter, except that `%`*fmt* must be valid. The width of the cells is chosen by `table` to present the results best. The `cellwidth()` option allows you to override `table`'s choice.

Limits

Up to four variables may be specified in the by(), so with the three row, column, and supercolumn variables, seven-way tables may be displayed.

Up to five statistics may be displayed in each cell of the table.

The sum of the number of rows, columns, supercolumns, and superrows is called the number of margins. A table may contain up to 3,000 margins. Thus a one-way table may contain 3,000 rows. A two-way table could contain 2,998 rows and two columns, 2,997 rows and three columns, . . ., 1,500 rows and 1,500 columns, . . ., two rows and 2,998 columns. A three-way table is similarly limited by the sum of the number of rows, columns, and supercolumns. A $r \times c \times d$ table is feasible if $r + c + d \leq 3{,}000$. The limit is set in terms of the sum of the rows, columns, supercolumns, and superrows, and not, as you might expect, in terms of their product.

Remarks and examples

stata.com

Remarks are presented under the following headings:

One-way tables

▷ Example 1

From the automobile dataset, here is a simple one-way table:

```
. use http://www.stata-press.com/data/r15/auto2
(1978 Automobile Data)
. table rep78, contents(mean mpg)
```

Repair Record 1978	mean(mpg)
Poor	21
Fair	19.125
Average	19.4333
Good	21.6667
Excellent	27.3636

We are not limited to including only one statistic:

```
. table rep78, c(n mpg  mean mpg  sd mpg  median mpg)
```

Repair Record 1978	N(mpg)	mean(mpg)	sd(mpg)	med(mpg)
Poor	2	21	4.24264	21
Fair	8	19.125	3.758324	18
Average	30	19.4333	4.141325	19
Good	18	21.6667	4.93487	22.5
Excellent	11	27.3636	8.732385	30

We abbreviated `contents()` as `c()`. The `format()` option will allow us to better format the numbers in the table:

```
. table rep78, c(n mpg  mean mpg  sd mpg  median mpg) format(%9.2f)
```

Repair Record 1978	N(mpg)	mean(mpg)	sd(mpg)	med(mpg)
Poor	2	21.00	4.24	21.00
Fair	8	19.12	3.76	18.00
Average	30	19.43	4.14	19.00
Good	18	21.67	4.93	22.50
Excellent	11	27.36	8.73	30.00

The `center` option will center the results under the headings:

```
. table rep78, c(n mpg  mean mpg  sd mpg  median mpg) format(%9.2f) center
```

Repair Record 1978	N(mpg)	mean(mpg)	sd(mpg)	med(mpg)
Poor	2	21.00	4.24	21.00
Fair	8	19.12	3.76	18.00
Average	30	19.43	4.14	19.00
Good	18	21.67	4.93	22.50
Excellent	11	27.36	8.73	30.00

◁

Two-way tables

▷ Example 2

In example 1, when we typed '`table rep78, ...`', we obtained a one-way table. If we were to type '`table rep78 foreign, ...`', we would obtain a two-way table:

```
. table rep78 foreign, c(mean mpg)
```

Repair Record 1978	Car type Domestic	Foreign
Poor	21	
Fair	19.125	
Average	19	23.3333
Good	18.4444	24.8889
Excellent	32	26.3333

Note the missing cells. Certain combinations of repair record and car type do not exist in our dataset.

As with one-way tables, we can specify a display format for the cells and center the numbers within the cells if we wish.

```
. table rep78 foreign, c(mean mpg) format(%9.2f) center
```

Repair Record 1978	Car type	
	Domestic	Foreign
Poor	21.00	
Fair	19.12	
Average	19.00	23.33
Good	18.44	24.89
Excellent	32.00	26.33

We can obtain row totals by specifying the `row` option and obtain column totals by specifying the `col` option. We specify both below:

```
. table rep78 foreign, c(mean mpg) format(%9.2f) center row col
```

Repair Record 1978	Car type		
	Domestic	Foreign	Total
Poor	21.00		21.00
Fair	19.12		19.12
Average	19.00	23.33	19.43
Good	18.44	24.89	21.67
Excellent	32.00	26.33	27.36
Total	19.54	25.29	21.29

`table` can display multiple statistics within cells, but once we move beyond one-way tables, the table becomes busy:

```
. table foreign rep78, c(mean mpg  n mpg) format(%9.2f) center
```

	Repair Record 1978				
Car type	Poor	Fair	Average	Good	Excellent
Domestic	21.00	19.12	19.00	18.44	32.00
	2	8	27	9	2
Foreign			23.33	24.89	26.33
			3	9	9

This two-way table with two statistics per cell works well here. That was, in part, helped along by our interchanging the rows and columns. We turned the table around by typing `table foreign rep78` rather than `table rep78 foreign`.

Another way to display two-way tables is to specify a row and superrow rather than a row and column. We do that below and display three statistics per cell:

```
. table foreign, by(rep78) c(mean mpg  sd mpg  n mpg) format(%9.2f) center
```

Repair Record 1978 and Car type	mean(mpg)	sd(mpg)	N(mpg)
Poor			
Domestic	21.00	4.24	2
Foreign			
Fair			
Domestic	19.12	3.76	8
Foreign			
Average			
Domestic	19.00	4.09	27
Foreign	23.33	2.52	3
Good			
Domestic	18.44	4.59	9
Foreign	24.89	2.71	9
Excellent			
Domestic	32.00	2.83	2
Foreign	26.33	9.37	9

Three-way tables

▷ Example 3

We have data on the prevalence of byssinosis, a form of pneumoconiosis to which workers exposed to cotton dust are susceptible. The dataset is on 5,419 workers in a large cotton mill. We know whether each worker smokes, his or her race, and the dustiness of the work area. The categorical variables are

smokes	Smoker or nonsmoker in the last five years.
race	White or other.
workplace	1 (most dusty), 2 (less dusty), 3 (least dusty).

Moreover, this dataset includes a frequency-weight variable pop. Here is a three-way table showing the fraction of workers with byssinosis:

```
. use http://www.stata-press.com/data/r15/byssin
(Byssinosis incidence)
. table workplace smokes race [fw=pop], c(mean prob)
```

Dustiness of workplace	Race and Smokes			
	other		white	
	no	yes	no	yes
least	.0107527	.0101523	.0081549	.0162774
less	.02	.0081633	.0136612	.0143149
most	.0820896	.1679105	.0833333	.2295082

This table would look better if we showed the fraction to four digits:

```
. table workplace smokes race [fw=pop], c(mean prob) format(%9.4f)
```

Dustiness of workplace	Race and Smokes			
	other		white	
	no	yes	no	yes
least	0.0108	0.0102	0.0082	0.0163
less	0.0200	0.0082	0.0137	0.0143
most	0.0821	0.1679	0.0833	0.2295

In this table, the rows are the dustiness of the workplace, the columns are whether the worker smokes, and the supercolumns are the worker's race.

Now we request that the table include the supercolumn totals by specifying the `sctotal` option, which we can abbreviate as `sc`:

```
. table workplace smokes race [fw=pop], c(mean prob) format(%9.4f) sc
```

Dustiness of workplace	Race and Smokes					
	other		white		Total	
	no	yes	no	yes	no	yes
least	0.0108	0.0102	0.0082	0.0163	0.0090	0.0145
less	0.0200	0.0082	0.0137	0.0143	0.0159	0.0123
most	0.0821	0.1679	0.0833	0.2295	0.0826	0.1929

The supercolumn total is the total over `race` and is divided into its columns based on `smokes`. Here is the table with the column rather than the supercolumn totals:

```
. table workplace smokes race [fw=pop], c(mean prob) format(%9.4f) col
```

Dustiness of workplace	Race and Smokes					
	other			white		
	no	yes	Total	no	yes	Total
least	0.0108	0.0102	0.0104	0.0082	0.0163	0.0129
less	0.0200	0.0082	0.0135	0.0137	0.0143	0.0140
most	0.0821	0.1679	0.1393	0.0833	0.2295	0.1835

Here is the table with both column and supercolumn totals:

```
. table workplace smokes race [fw=pop], c(mean prob) format(%9.4f) sc col
```

Dustiness of workplace	Race and Smokes								
	other			white			Total		
	no	yes	Total	no	yes	Total	no	yes	Total
least	0.0108	0.0102	0.0104	0.0082	0.0163	0.0129	0.0090	0.0145	0.0122
less	0.0200	0.0082	0.0135	0.0137	0.0143	0.0140	0.0159	0.0123	0.0138
most	0.0821	0.1679	0.1393	0.0833	0.2295	0.1835	0.0826	0.1929	0.1570

`table` is struggling to keep this table from becoming too wide—notice how it divided the words in the title in the top-left stub. Here, if the table had more columns, or, if we demanded more digits, `table` would be forced to segment the table and present it in pieces, which it would do:

```
. table workplace smokes race [fw=pop], c(mean prob) format(%9.6f) sc col
```

Dustiness of workplace	Race and Smokes: other: no	other: yes	other: Total	white: no	white: yes	white: Total
least	0.010753	0.010152	0.010417	0.008155	0.016277	0.012949
less	0.020000	0.008163	0.013483	0.013661	0.014315	0.014035
most	0.082090	0.167910	0.139303	0.083333	0.229508	0.183521

Dustiness of workplace	Race and Smokes: Total: no	Total: yes	Total: Total
least	0.008990	0.014471	0.012174
less	0.015901	0.012262	0.013846
most	0.082569	0.192905	0.156951

Here three digits is probably enough, so here is the table including all the row, column, and supercolumn totals:

```
. table workplace smokes race [fw=pop], c(mean prob) format(%9.3f) sc col row
```

Dustiness of workplace	Race and Smokes: other: no	other: yes	other: Total	white: no	white: yes	white: Total	Total: no	Total: yes	Total: Total
least	0.011	0.010	0.010	0.008	0.016	0.013	0.009	0.014	0.012
less	0.020	0.008	0.013	0.014	0.014	0.014	0.016	0.012	0.014
most	0.082	0.168	0.139	0.083	0.230	0.184	0.083	0.193	0.157
Total	0.025	0.048	0.038	0.014	0.035	0.026	0.018	0.039	0.030

We can show multiple statistics:

```
. table workplace smokes race [fw=pop], c(mean prob  n prob) format(%9.3f) sc
> col row
```

Dustiness of workplace	Race and Smokes: other: no	other: yes	other: Total	white: no	white: yes	white: Total	Total: no	Total: yes	Total: Total
least	0.011	0.010	0.010	0.008	0.016	0.013	0.009	0.014	0.012
	465	591	1,056	981	1,413	2,394	1,446	2,004	3,450
less	0.020	0.008	0.013	0.014	0.014	0.014	0.016	0.012	0.014
	200	245	445	366	489	855	566	734	1,300
most	0.082	0.168	0.139	0.083	0.230	0.184	0.083	0.193	0.157
	134	268	402	84	183	267	218	451	669
Total	0.025	0.048	0.038	0.014	0.035	0.026	0.018	0.039	0.030
	799	1,104	1,903	1,431	2,085	3,516	2,230	3,189	5,419

◁

Four-way and higher-dimensional tables

▷ Example 4

Let's pretend that our byssinosis dataset also recorded each worker's sex (it does not, and we have made up this extra information). We obtain a four-way table just as we would a three-way table, but we specify the fourth variable as a superrow by including it in the `by()` option:

```
. use http://www.stata-press.com/data/r15/byssin1
(Byssinosis incidence)
. table workplace smokes race [fw=pop], by(sex) c(mean prob)  format(%9.3f) sc
> col row
```

Sex and Dustiness of workplace	Race and Smokes: other — no	other — yes	other — Total	white — no	white — yes	white — Total	Total — no	Total — yes	Total — Total
Female									
least	0.006	0.009	0.008	0.009	0.021	0.016	0.009	0.018	0.014
less	0.020	0.008	0.010	0.015	0.015	0.015	0.016	0.012	0.014
most	0.057	0.154	0.141				0.057	0.154	0.141
Total	0.017	0.051	0.043	0.011	0.020	0.016	0.012	0.032	0.024
Male									
least	0.013	0.011	0.012	0.006	0.007	0.006	0.009	0.008	0.009
less	0.020	0.000	0.019	0.000	0.013	0.011	0.016	0.013	0.014
most	0.091	0.244	0.136	0.083	0.230	0.184	0.087	0.232	0.167
Total	0.029	0.041	0.033	0.020	0.056	0.043	0.025	0.052	0.039

If our dataset also included work group and we wanted a five-way table, we could include both the sex and work-group variables in the `by()` option. You may include up to four variables in `by()`, and so produce up to 7-way tables.

◁

Video example

Combining cross-tabulations and descriptives in Stata

Methods and formulas

The contents of cells are calculated by `collapse` and are displayed by `tabdisp`; see [D] **collapse** and [P] **tabdisp**.

Also see

[R] **summarize** — Summary statistics

[R] **tabstat** — Compact table of summary statistics

[R] **tabulate oneway** — One-way table of frequencies

[R] **tabulate twoway** — Two-way table of frequencies

[D] **collapse** — Make dataset of summary statistics

[P] **tabdisp** — Display tables