

tabstat — Compact table of summary statistics

[Description](#)
[Options](#)
[Also see](#)

[Quick start](#)
[Remarks and examples](#)

[Menu](#)
[Acknowledgments](#)

[Syntax](#)
[Reference](#)

Description

`tabstat` displays summary statistics for a series of numeric variables in one table. It allows you to specify the list of statistics to be displayed. Statistics can be calculated (conditioned on) another variable. `tabstat` allows substantial flexibility in terms of the statistics presented and the format of the table.

Quick start

Mean of `v1` displayed using `v1`'s display format

```
tabstat v1, format
```

Same as above, but use format with 2 significant digits and a comma

```
tabstat v1, format(%9.2fc)
```

Nonmissing observations, mean, standard error, and coefficient of variation for `v1`

```
tabstat v1, statistics(n mean semean cv)
```

Quartiles and interquartile range of `v1` and `v2`

```
tabstat v1 v2, statistics(q iqr)
```

Same as above, but report statistics separately for each level of `catvar`

```
tabstat v1 v2, by(catvar) statistics(q iqr)
```

Same as above, but display a separate column for each statistic

```
tabstat v1 v2, by(catvar) statistics(q iqr) columns(statistics)
```

Menu

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

Syntax

```
tabstat varlist [if] [in] [weight] [, options]
```

| <i>options</i> | Description |
|--|---|
| Main | |
| <code>by(<i>varname</i>)</code> | group statistics by variable |
| <code>statistics(<i>statname</i> [...])</code> | report specified statistics |
| Options | |
| <code>labelwidth(#)</code> | width for <code>by()</code> variable labels; default is <code>labelwidth(16)</code> |
| <code>varwidth(#)</code> | variable width; default is <code>varwidth(12)</code> |
| <code>columns(<i>variables</i>)</code> | display variables in table columns; the default |
| <code>columns(<i>statistics</i>)</code> | display statistics in table columns |
| <code>format[<i>(%fmt)</i>]</code> | display format for statistics; default format is <code>%9.0g</code> |
| <code>casewise</code> | perform casewise deletion of observations |
| <code>nototal</code> | do not report overall statistics; use with <code>by()</code> |
| <code>missing</code> | report statistics for missing values of <code>by()</code> variable |
| <code>noseparator</code> | do not use separator line between <code>by()</code> categories |
| <code>longstub</code> | make left table stub wider |
| <code>save</code> | store summary statistics in <code>r()</code> |

`by` is allowed; see [D] [by](#).

`aweight`s and `fweight`s are allowed; see [U] [11.1.6 weight](#).

Options

Main

`by(varname)` specifies that the statistics be displayed separately for each unique value of *varname*; *varname* may be numeric or string. For instance, `tabstat height` would present the overall mean of height. `tabstat height, by(sex)` would present the mean height of males, and of females, and the overall mean height. Do not confuse the `by()` option with the `by` prefix (see [D] [by](#)); both may be specified.

`statistics(statname [...])` specifies the statistics to be displayed; the default is equivalent to specifying `statistics(mean)`. (`stats()` is a synonym for `statistics()`.) Multiple statistics may be specified and are separated by white space, such as `statistics(mean sd)`. Available statistics are

| <i>statname</i> | Definition | <i>statname</i> | Definition |
|-----------------|--|-----------------|--------------------------------------|
| <u>mean</u> | mean | p1 | 1st percentile |
| <u>count</u> | count of nonmissing observations | p5 | 5th percentile |
| <u>n</u> | same as count | p10 | 10th percentile |
| <u>sum</u> | sum | p25 | 25th percentile |
| <u>max</u> | maximum | <u>median</u> | median (same as p50) |
| <u>min</u> | minimum | p50 | 50th percentile (same as median) |
| <u>range</u> | range = max - min | p75 | 75th percentile |
| <u>sd</u> | standard deviation | p90 | 90th percentile |
| <u>variance</u> | variance | p95 | 95th percentile |
| <u>cv</u> | coefficient of variation (sd/mean) | p99 | 99th percentile |
| <u>semean</u> | standard error of mean (sd/ \sqrt{n}) | iqr | interquartile range = p75 - p25 |
| <u>skewness</u> | skewness | q | equivalent to specifying p25 p50 p75 |
| <u>kurtosis</u> | kurtosis | | |

Options

`labelwidth(#)` specifies the maximum width to be used within the stub to display the labels of the `by()` variable. The default is `labelwidth(16)`. $8 \leq \# \leq 32$.

`varwidth(#)` specifies the maximum width to be used within the stub to display the names of the variables. The default is `varwidth(12)`. `varwidth()` is effective only with `columns(statistics)`. Setting `varwidth()` implies `longstub`. $8 \leq \# \leq 32$.

`columns(variables | statistics)` specifies whether to display variables or statistics in the columns of the table. `columns(variables)` is the default when more than one variable is specified.

`format` and `format(%fmt)` specify how the statistics are to be formatted. The default is to use a `%9.0g` format.

`format` specifies that each variable's statistics be formatted with the variable's display format; see [\[D\] format](#).

`format(%fmt)` specifies the format to be used for all statistics.

The column width is the maximum width of these formats. The minimum column width is nine display characters.

`casewise` specifies casewise deletion of observations. Statistics are to be computed for the sample that is not missing for any of the variables in `varlist`. The default is to use all the nonmissing values for each variable.

`nototal` is for use with `by()`; it specifies that the overall statistics not be reported.

`missing` specifies that missing values of the `by()` variable be treated just like any other value and that statistics should be displayed for them. The default is not to report the statistics for the `by()=missing` group. If the `by()` variable is a string variable, `by()=""` is considered to mean missing.

`noseparator` specifies that a separator line between the `by()` categories not be displayed.

`longstub` specifies that the left stub of the table be made wider so that it can include names of the statistics or variables in addition to the categories of `by(varname)`. The default is to describe the statistics or variables in a header. `longstub` is ignored if `by(varname)` is not specified.

`save` specifies that the summary statistics be returned in `r()`. The overall (unconditional) statistics are returned in matrix `r(StatTotal)` (rows are statistics, columns are variables). The conditional statistics are returned in the matrices `r(Stat1)`, `r(Stat2)`, ..., and the names of the corresponding variables are returned in the macros `r(name1)`, `r(name2)`, ...

Remarks and examples

This command is probably most easily understood by going through a series of examples.

▷ Example 1

We have data on the price, weight, mileage rating, and repair record of 22 foreign and 52 domestic 1978 automobiles. We want to summarize these variables for the different origins of the automobiles.

```
. use https://www.stata-press.com/data/r18/auto
(1978 automobile data)
. tabstat price weight mpg rep78, by(foreign)
Summary statistics: Mean
Group variable: foreign (Car origin)
```

| foreign | price | weight | mpg | rep78 |
|----------|----------|----------|----------|----------|
| Domestic | 6072.423 | 3317.115 | 19.82692 | 3.020833 |
| Foreign | 6384.682 | 2315.909 | 24.77273 | 4.285714 |
| Total | 6165.257 | 3019.459 | 21.2973 | 3.405797 |

More summary statistics can be requested via the `statistics()` option. The group totals can be suppressed with the `nototal` option.

```
. tabstat price weight mpg rep78, by(foreign) stat(mean sd min max) nototal
Summary statistics: Mean, SD, Min, Max
Group variable: foreign (Car origin)
```

| foreign | price | weight | mpg | rep78 |
|----------|----------|----------|----------|----------|
| Domestic | 6072.423 | 3317.115 | 19.82692 | 3.020833 |
| | 3097.104 | 695.3637 | 4.743297 | .837666 |
| | 3291 | 1800 | 12 | 1 |
| | 15906 | 4840 | 34 | 5 |
| Foreign | 6384.682 | 2315.909 | 24.77273 | 4.285714 |
| | 2621.915 | 433.0035 | 6.611187 | .7171372 |
| | 3748 | 1760 | 14 | 3 |
| | 12990 | 3420 | 41 | 5 |

Although the header of the table describes the statistics running vertically in the “cells”, the table may become hard to read, especially with many variables or statistics. The `longstub` option specifies that a column be added describing the contents of the cells. The `format` option can be issued to specify that `tabstat` display the statistics by using the display format of the variables rather than the overall default `%9.0g`.

```
. tabstat price weight mpg rep78, by(foreign) stat(mean sd min max) long format
```

| foreign | Stats | price | weight | mpg | rep78 |
|----------|-------|---------|---------|---------|---------|
| Domestic | Mean | 6,072.4 | 3,317.1 | 19.8269 | 3.02083 |
| | SD | 3,097.1 | 695.364 | 4.7433 | .837666 |
| | Min | 3,291 | 1,800 | 12 | 1 |
| | Max | 15,906 | 4,840 | 34 | 5 |
| Foreign | Mean | 6,384.7 | 2,315.9 | 24.7727 | 4.28571 |
| | SD | 2,621.9 | 433.003 | 6.61119 | .717137 |
| | Min | 3,748 | 1,760 | 14 | 3 |
| | Max | 12,990 | 3,420 | 41 | 5 |
| Total | Mean | 6,165.3 | 3,019.5 | 21.2973 | 3.4058 |
| | SD | 2,949.5 | 777.194 | 5.7855 | .989932 |
| | Min | 3,291 | 1,760 | 12 | 1 |
| | Max | 15,906 | 4,840 | 41 | 5 |

We can specify a layout of the table in which the statistics run horizontally and the variables run vertically by specifying the `col(statistics)` option.

```
. tabstat price weight mpg rep78, by(foreign) stat(min mean max) col(stat) long
```

| foreign | Variable | Min | Mean | Max |
|----------|----------|------|----------|-------|
| Domestic | price | 3291 | 6072.423 | 15906 |
| | weight | 1800 | 3317.115 | 4840 |
| | mpg | 12 | 19.82692 | 34 |
| | rep78 | 1 | 3.020833 | 5 |
| Foreign | price | 3748 | 6384.682 | 12990 |
| | weight | 1760 | 2315.909 | 3420 |
| | mpg | 14 | 24.77273 | 41 |
| | rep78 | 3 | 4.285714 | 5 |
| Total | price | 3291 | 6165.257 | 15906 |
| | weight | 1760 | 3019.459 | 4840 |
| | mpg | 12 | 21.2973 | 41 |
| | rep78 | 1 | 3.405797 | 5 |

Finally, `tabstat` can also be used to enhance `summarize` so we can specify the statistics to be displayed. For instance, we can display the number of observations, the mean, the coefficient of variation, and the 25%, 50%, and 75% quantiles for a list of variables.

```
. tabstat price weight mpg rep78, stat(n mean cv q) col(stat)
```

| variable | N | mean | cv | p25 | p50 | p75 |
|----------|----|----------|----------|------|--------|------|
| price | 74 | 6165.257 | .478406 | 4195 | 5006.5 | 6342 |
| weight | 74 | 3019.459 | .2573949 | 2240 | 3190 | 3600 |
| mpg | 74 | 21.2973 | .2716543 | 18 | 20 | 25 |
| rep78 | 69 | 3.405797 | .290661 | 3 | 3 | 4 |

Because we did not specify the `by()` option, these statistics were not displayed for the subgroups of the data formed by the categories of the `by()` variable.

Video example

[Descriptive statistics in Stata](#)

Acknowledgments

The `tabstat` command was written by Jeroen Weesie and Vincent Buskens both of the Department of Sociology at Utrecht University, The Netherlands.

Reference

Donath, S. 2018. `baselinetable`: A command for creating one- and two-way tables of summary statistics. *Stata Journal* 18: 327–344.

Also see

[R] [summarize](#) — Summary statistics

[R] [table](#) — Table of frequencies, summaries, and command results

[R] [table summary](#) — Table of summary statistics

[R] [tabulate, summarize\(\)](#) — One- and two-way tables of summary statistics

[D] [collapse](#) — Make dataset of summary statistics

Stata, Stata Press, and Mata are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2023 StataCorp LLC, College Station, TX, USA. All rights reserved.

