

table twoway — Two-way tabulation

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

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

[Menu](#)
[Stored results](#)

[Syntax](#)
[References](#)

Description

In this entry, we discuss how to use `table` to create a two-way tabulation, including frequencies, percentages, and proportions.

Quick start

Table of frequencies, with rows defined by categories of `a1` and columns defined by categories of `a2`

```
table a1 a2
```

Same as above, but treat missing values like other values

```
table a1 a2, missing
```

Table with the percentage of observations in each cell

```
table a1 a2, statistic(percent)
```

For each category of `a1`, report the percentage of observations across levels of `a2`

```
table a1 a2, statistic(percent, across(a2))
```

Report frequencies and the proportion of observations across categories of `a1`, enclosed within parentheses

```
table a1 a2, statistic(frequency) ///  
    statistic(proportion, across(a1)) sformat("%s" proportion)
```

Menu

Statistics > Summaries, tables, and tests > Tables of frequencies, summaries, and command results

Syntax

Basic two-way tabulation

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

Customized two-way tabulation

```
table [(rowspec)] [(colspec)] [if] [in] [weight] [, options]
```

rowspec may be empty or may include *rowvar*, **result**, or *rowvar* and **result**, where **result** refers to the requested statistics.

colspec may be empty or may include *colvar*, **result**, or *colvar* and **result**, where **result** refers to the requested statistics.

<i>options</i>	Description
Main	
totals (<i>totals</i>)	report only the specified totals
nototals	suppress the marginal totals
Statistics	
u nderline stat (<i>stat</i> [, <i>statopts</i>])	statistic to be reported; default is statistic (frequency) when no weights are specified and statistic (sumw) otherwise
Formats	
n format(<i>%fmt</i> [<i>results</i>]) [, <i>basestyle</i>])	specify numeric format
s format(<i>sfmt</i> [<i>results</i>])	specify string format
Options	
missing	treat numeric missing values of <i>rowvar</i> and <i>colvar</i> like other values
zerocounts	report 0 for empty cell counts
name (<i>cname</i>)	collect results into a collection named <i>cname</i>
append	append results to an existing collection
replace	replace results of an existing collection
label (<i>filename</i>)	specify the collection labels
style (<i>filename</i> [, <i>override</i>])	specify the collection style
markvar (<i>newvar</i>)	create <i>newvar</i> that identifies observations used in the tabulation

fweights, **awweights**, **iweights**, and **pweights** are allowed; see [U] 11.1.6 **weight**.

strL variables are not allowed; see [U] 12.4.8 **strL**.

markvar() does not appear in the dialog box.

Options

Main

`totals(totals)` and `nototals` control which totals are to be displayed in the table. By default, all totals are reported.

`totals(totals)` specifies which margin totals to display in the reported table. `totals` can contain `rowvar`, `colvar`, and their interaction. Interactions can be specified by using the `#` operator.

`nototals` prevents `table` from displaying any totals.

Statistics

`statistic(stat[, statopts])` specifies the statistic to be displayed. `statistic()` may be repeated to request multiple statistics.

Available statistics are

<i>stat</i>	Definition
<code>frequency</code>	frequency
<code>sumw</code>	sum of weights
<code>proportion</code>	proportion
<code>percent</code>	percentage
<code>rawproportion</code>	proportion ignoring optionally specified weights
<code>rawpercent</code>	percentage ignoring optionally specified weights

The following options may be specified in combination with statistics `proportion`, `percent`, `rawproportion`, and `rawpercent`:

<i>statopts</i>	Definition
<code>across(rowvar)</code>	percentages or proportions across rows
<code>across(colvar)</code>	percentages or proportions across columns
<code>total</code>	compute overall percentages or proportions

Formats

`nformat(%fmt [results][, basestyle])` changes the numeric format, such as the number of decimal places, for specified results. If `results` are not specified, the numeric format is changed for all results.

`results` may be any statistic named in option `statistic()` (that is, any `stat`).

This option is repeatable, and when multiple formats apply to one result, the rightmost specification is applied.

This option does not affect the format of numeric layout variables (`rowspec` and `colspec`). The default format of these variables is taken from the dataset.

`basestyle` indicates that the format be applied to results that do not already have their own format instead of overriding the format for all results.

`sformat(sfmt [results])` changes the string format for specified results. You can, for instance, add symbols or text to the values reported in the table by modifying the string format.

`sfmt` may contain a mix of text and `%s`. Here `%s` refers to the numeric value that is formatted as specified using `nformat()`. The text will be placed around the numeric values in your table as it is placed around `%s` in this option. For instance, to place parentheses around the percent statistics, you can specify `sformat("(%s)" percent)`.

results may be any statistic named in option `statistic()` (that is, any *stat*).

Two text characters must be specified using a special character sequence if you want them to be displayed in your table. To include %, type %%. To include \, type \\ . For instance, to place a percent sign following percent statistics, you can specify `sformat("%s%" percent)`.

This option is repeatable, and when multiple formats apply to one result, the rightmost specification is applied.

Options

`missing` specifies that numeric missing values of *rowvar* or *colvar* be treated as valid categories. By default, observations with a numeric missing value in *rowvar* or *colvar* are omitted.

`zerocounts` specifies that `table` report a 0 in empty cells for the `frequency` statistic.

`name(cname)` specifies that a collection named *cname* be associated with the collected statistics and results. The default is `name(Table)`.

`append` specifies that `table` append its collection information into the collection named in `name()`.

`replace` permits `table` to overwrite an existing collection. This option is implied for `name(Table)` when `append` is not specified.

`label(filename)` specifies the *filename* containing the collection labels to use for your table. Labels in *filename* will be loaded for the table, and any labels not specified in *filename* will be taken from the labels defined in `c(collect_label)`. The default is to use only the collection labels set in `c(collect_label)`; see [TABLES] [set collect_label](#).

`style(filename [, override])` specifies the *filename* containing the collection styles to use for your table. The default collection styles will be discarded, and only the collection styles in *filename* will be applied.

If you prefer the default collection styles but also want to apply any styles in *filename*, specify `override`. If there are conflicts between the default collection styles and those in *filename*, the ones in *filename* will take precedence.

The default is to use only the collection styles set in `c(table_style)`; see [TABLES] [set table_style](#).

The following option is available with `table` but is not shown in the dialog box:

`markvar(newvar)` generates an indicator variable that identifies the observations used in the tabulation.

Remarks and examples

[stata.com](http://www.stata.com)

Remarks are presented under the following headings:

Tabulation of two variables
Tabulation, including percentages
Customizing results

Tabulation of two variables

To obtain a two-way tabulation that reports the number of observations across the levels of two categorical variables, we need to specify only the names of the categorical variables following `table`.

To demonstrate, we use data from the Second National Health and Nutrition Examination Survey (NHANES II) (McDowell et al. 1981) and create a two-way tabulation of self-reported health status (`hlthstat`) by region of the USA (`region`).

```
. use https://www.stata-press.com/data/r18/nhanes21
(Second National Health and Nutrition Examination Survey)
. table hlthstat region
```

	Region				Total
	NE	MW	S	W	
Health status					
Excellent	562	730	546	569	2,407
Very good	558	721	651	661	2,591
Good	631	735	807	765	2,938
Fair	257	419	532	462	1,670
Poor	77	167	317	168	729
Total	2,085	2,772	2,853	2,625	10,335

We can examine the missing values as well by adding the missing option.

```
. table hlthstat region, missing
```

	Region				Total
	NE	MW	S	W	
Health status					
Excellent	562	730	546	569	2,407
Very good	558	721	651	661	2,591
Good	631	735	807	765	2,938
Fair	257	419	532	462	1,670
Poor	77	167	317	168	729
.	1	1			2
Blank but applicable	10	1		3	14
Total	2,096	2,774	2,853	2,628	10,351

We find that 16 individuals have a missing health status, and the majority of these are from individuals in the Northeast. The empty cells correspond to regions in which all the individuals have a nonmissing health status; we can fill in these empty cells with 0s:

```
. table hlthstat region, missing zerocounts
```

	Region				Total
	NE	MW	S	W	
Health status					
Excellent	562	730	546	569	2,407
Very good	558	721	651	661	2,591
Good	631	735	807	765	2,938
Fair	257	419	532	462	1,670
Poor	77	167	317	168	729
.	1	1	0	0	2
Blank but applicable	10	1	0	3	14
Total	2,096	2,774	2,853	2,628	10,351

Tabulation, including percentages

Instead of frequencies, we can request that table report the percentage of observations in each cell of the table by specifying the `statistic(percent)` option.

```
. table hlthstat region, statistic(percent)
```

	NE	MW	Region S	W	Total
Health status					
Excellent	5.44	7.06	5.28	5.51	23.29
Very good	5.40	6.98	6.30	6.40	25.07
Good	6.11	7.11	7.81	7.40	28.43
Fair	2.49	4.05	5.15	4.47	16.16
Poor	0.75	1.62	3.07	1.63	7.05
Total	20.17	26.82	27.61	25.40	100.00

We see that 5.44% of all observations correspond to individuals in excellent health who live in the Northeast.

Rather than looking at overall percentages, we might want to examine the distribution of observations within each health status level across the four regions. To do this, we can add the `across(region)` option.

```
. table hlthstat region, statistic(percent, across(region))
```

	NE	MW	Region S	W	Total
Health status					
Excellent	23.35	30.33	22.68	23.64	100.00
Very good	21.54	27.83	25.13	25.51	100.00
Good	21.48	25.02	27.47	26.04	100.00
Fair	15.39	25.09	31.86	27.66	100.00
Poor	10.56	22.91	43.48	23.05	100.00
Total	20.17	26.82	27.61	25.40	100.00

Of individuals reporting excellent health, 23.35% live in the Northeast, while 30.33% live in the Midwest, 22.68% live in the South, and 23.64% live in the West.

We can also look at the distribution of observations across health status categories within each region.

```
. table hlthstat region, statistic(percent, across(hlthstat))
```

	NE	MW	Region S	W	Total
Health status					
Excellent	26.95	26.33	19.14	21.68	23.29
Very good	26.76	26.01	22.82	25.18	25.07
Good	30.26	26.52	28.29	29.14	28.43
Fair	12.33	15.12	18.65	17.60	16.16
Poor	3.69	6.02	11.11	6.40	7.05
Total	100.00	100.00	100.00	100.00	100.00

Of individuals living in the South, 11.11% report having poor health. This is notably higher than the percentage of individuals reporting poor health in the other regions.

It is often helpful to see both frequencies and percentages in the same table. To do this, we can add the `statistic(frequency)` option to our command.

```
. table hlthstat region, statistic(frequency)
> statistic(percent, across(hlthstat))
```

	Region				Total
	NE	MW	S	W	
Health status					
Excellent					
Frequency	562	730	546	569	2,407
Percent	26.95	26.33	19.14	21.68	23.29
Very good					
Frequency	558	721	651	661	2,591
Percent	26.76	26.01	22.82	25.18	25.07
Good					
Frequency	631	735	807	765	2,938
Percent	30.26	26.52	28.29	29.14	28.43
Fair					
Frequency	257	419	532	462	1,670
Percent	12.33	15.12	18.65	17.60	16.16
Poor					
Frequency	77	167	317	168	729
Percent	3.69	6.02	11.11	6.40	7.05
Total					
Frequency	2,085	2,772	2,853	2,625	10,335
Percent	100.00	100.00	100.00	100.00	100.00

Customizing results

There are several ways that we can customize the results of our two-way tabulation.

For instance, in some cases, we may prefer to omit the row and column totals. We can specify the `nototals` option to suppress these totals.

```
. table hlthstat region, statistic(frequency)
> statistic(percent, across(hlthstat)) nototals
```

	Region			
	NE	MW	S	W
Health status				
Excellent				
Frequency	562	730	546	569
Percent	26.95	26.33	19.14	21.68
Very good				
Frequency	558	721	651	661
Percent	26.76	26.01	22.82	25.18
Good				
Frequency	631	735	807	765
Percent	30.26	26.52	28.29	29.14
Fair				
Frequency	257	419	532	462
Percent	12.33	15.12	18.65	17.60
Poor				
Frequency	77	167	317	168
Percent	3.69	6.02	11.11	6.40

Or perhaps we want to see row totals or column totals but not both. We can include the `totals(region)` option to display only the region totals.

```
. table hlthstat region, statistic(frequency)
> statistic(percent, across(hlthstat)) totals(region)
```

	Region			
	NE	MW	S	W
Health status				
Excellent				
Frequency	562	730	546	569
Percent	26.95	26.33	19.14	21.68
Very good				
Frequency	558	721	651	661
Percent	26.76	26.01	22.82	25.18
Good				
Frequency	631	735	807	765
Percent	30.26	26.52	28.29	29.14
Fair				
Frequency	257	419	532	462
Percent	12.33	15.12	18.65	17.60
Poor				
Frequency	77	167	317	168
Percent	3.69	6.02	11.11	6.40
Total				
Frequency	2,085	2,772	2,853	2,625
Percent	100.00	100.00	100.00	100.00

Once we have the statistics we want in our table, we can format the way that they appear. If, for instance, we want to add a percent sign to each of our percentages, we can specify the `sformat("%s%" percent)` option. The `sformat()` option specifies that we want to add string characters to the numbers in the table. Within it, we refer to the numeric values as `%s` and place any string characters we want around this. The percent sign is unique because it already has special meaning in this context. Therefore, we must type two percent signs, `%%`, to display one. Finally, by adding `percent` within the `sformat()` option, we specify that we want to apply this format only to the percent statistics.


```
. table hlthstat region, statistic(frequency)
> statistic(percent, across(hlthstat)) totals(region)
> sformat("%s%" percent)
```

	Region			
	NE	MW	S	W
Health status				
Excellent				
Frequency	562	730	546	569
Percent	26.95%	26.33%	19.14%	21.68%
Very good				
Frequency	558	721	651	661
Percent	26.76%	26.01%	22.82%	25.18%
Good				
Frequency	631	735	807	765
Percent	30.26%	26.52%	28.29%	29.14%
Fair				
Frequency	257	419	532	462
Percent	12.33%	15.12%	18.65%	17.60%
Poor				
Frequency	77	167	317	168
Percent	3.69%	6.02%	11.11%	6.40%
Total				
Frequency	2,085	2,772	2,853	2,625
Percent	100.00%	100.00%	100.00%	100.00%

Now that we have added the percent sign, we could argue that the labels Frequency and Percent are unnecessary. If we remove these statistic names from the row labels, we might also want to right-align the remaining labels in row headers. Finally, for readability, we could insert blank lines between levels of hlthstat. We could use the `collect` suite of commands to make these style changes. Fortunately, however, one of our [predefined styles](#), `table-tab2`, includes these style changes, and we can select it using the `style()` option.

```
. table hlthstat region, statistic(frequency)
> statistic(percent, across(hlthstat)) totals(region)
> sformat("%s%" percent) style(table-tab2)
```

	Region			
	NE	MW	S	W
Health status				
Excellent	562	730	546	569
	26.95%	26.33%	19.14%	21.68%
Very good	558	721	651	661
	26.76%	26.01%	22.82%	25.18%
Good	631	735	807	765
	30.26%	26.52%	28.29%	29.14%
Fair	257	419	532	462
	12.33%	15.12%	18.65%	17.60%
Poor	77	167	317	168
	3.69%	6.02%	11.11%	6.40%
Total	2,085	2,772	2,853	2,625
	100.00%	100.00%	100.00%	100.00%

You can learn more about the predefined styles described at [TABLES] **Predefined styles**. If none of these provide the exact style you want for your table, you can further customize the results by using the `collect` suite of commands. To learn more, see [TABLES] **Intro**.

If you wish to include this table in a paper, on a webpage, or in another format, you can easily export it in L^AT_EX, Word, Excel, HTML, and a variety of other formats by using `collect export`.

Stored results

`table` stores the following in `r()`:

Scalars

`r(N)` number of observations

References

Huber, C. 2021. Customizable tables in Stata 17, part 1: The new `table` command. *The Stata Blog: Not Elsewhere Classified*. <https://blog.stata.com/2021/06/07/customizable-tables-in-stata-17-part-1-the-new-table-command/>.

McDowell, A., A. Engel, J. T. Massey, and K. Maurer. 1981. Plan and operation of the Second National Health and Nutrition Examination Survey, 1976–1980. *Vital and Health Statistics* 1(15): 1–144.

Also see

[R] **table** — Table of frequencies, summaries, and command results

[R] **table intro** — Introduction to tables of frequencies, summaries, and command results

[R] **table multiway** — Multiway tables

[R] **table oneway** — One-way tabulation

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

[TABLES] **Intro** — Introduction

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. StataNow and NetCourseNow are trademarks of StataCorp LLC. 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.



For suggested citations, see the FAQ on [citing Stata documentation](#).