

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	
<code>totals(<i>totals</i>)</code>	report only the specified totals
<code>nototals</code>	suppress the marginal totals
Statistics	
<code><u>statistic</u>(<i>stat</i> [, <i>statopts</i>])</code>	statistic to be reported; default is <code>statistic(frequency)</code> when no weights are specified and <code>statistic(sumw)</code> otherwise
Formats	
<code>nformat(<i>%fmt</i> [<i>results</i>] [, <i>basestyle</i>])</code>	specify numeric format
<code>sformat(<i>sfmt</i> [<i>results</i>])</code>	specify string format
Title	
<code>title(<i>string</i>)</code>	add table title
<code><u>titlestyles</u>(<i>text_styles</i>)</code>	change table title styles
Notes	
<code>note(<i>string</i>)</code>	add table note
<code><u>notestyles</u>(<i>text_styles</i>)</code>	change table note styles
Export	
<code>export(<i>filename.suffix</i> [, <i>export_opts</i>])</code>	export table
Options	
<code><u>missing</u></code>	treat numeric missing values of <i>rowvar</i> and <i>colvar</i> like other values
<code><u>zerocounts</u></code>	report 0 for empty cell counts
<code>name(<i>cname</i>)</code>	collect results into a collection named <i>cname</i>
<code>append</code>	append results to an existing collection
<code>replace</code>	replace results of an existing collection
<code>label(<i>filename</i>)</code>	specify the collection labels
<code>style(<i>filename</i> [, <i>override</i>])</code>	specify the collection style
<code>markvar(<i>newvar</i>)</code>	create <i>newvar</i> that identifies observations used in the tabulation

fweights, aweights, 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.

<i>text_styles</i>	Description
font(<i>[fontfamily]</i> [<i>, font_opts</i>])	specify font style
smcl(<i>smcl</i>)	specify formatting for SMCL files
latex(<i>latex</i>)	specify L ^A T _E X macro
shading(<i>sspec</i>)	set background color, foreground color, and fill pattern

<i>font_opts</i>	Description
size(# [<i>unit</i>])	specify font size
color(<i>color</i>)	specify font color
variant(<i>variant</i>)	specify font variant and capitalization
[no]bold	specify whether to format text as bold
[no]italic	specify whether to format text as italic
[no]strikeout	specify whether to strike out text
[no]underline	specify whether to underline text

<i>suffix</i>	<i>fileformat</i>	Output format
docx	as(docx)	Microsoft Word
html	as(html)	HTML 5 with CSS
pdf	as(pdf)	PDF
xlsx	as(xlsx)	Microsoft Excel 2007/2010 or newer
xls	as(xls)	Microsoft Excel 1997/2003
tex	as(tex)	L ^A T _E X
smcl	as(smcl)	SMCL
txt	as(txt)	plain text
markdown	as(markdown)	Markdown
md	as(md)	Markdown

<i>export_opts</i>	Description
as(<i>fileformat</i>)	specify document type
replace	overwrite existing file
<i>docx_options</i>	available when exporting to .docx files
<i>html_options</i>	available when exporting to .html files
<i>pdf_options</i>	available when exporting to .pdf files
<i>excel_options</i>	available when exporting to .xls and .xlsx files
<i>tex_options</i>	available when exporting to .tex files
<i>smcl_option</i>	available when exporting to .smcl files
<i>txt_option</i>	available when exporting to .txt files
<i>md_option</i>	available when exporting to .markdown and .md files

<i>docx_options</i>	Description
<code>noisily</code>	show the putdocx commands used to export to the Microsoft Word file
<code>dofile(filename[, replace])</code>	save the putdocx commands used for exporting to the named do-file

<i>html_options</i>	Description
<code>append</code>	append to an existing file
<code>tableonly</code>	export only the table to the specified file
<code>cssfile(cssfile)</code>	define the styles in <i>cssfile</i> instead of <i>filename</i>
<code>prefix(prefix)</code>	use <i>prefix</i> to identify style classes

<i>pdf_options</i>	Description
<code>noisily</code>	show the putpdf commands used to export to the PDF file
<code>dofile(filename[, replace])</code>	save the putpdf commands used for exporting to the named do-file

<i>excel_options</i>	Description
<code>noisily</code>	show the putexcel commands used to export to the Excel file
<code>dofile(filename[, replace])</code>	save the putexcel commands used for exporting to the named do-file
<code>sheet(sheetname[, replace])</code>	specify the worksheet to use; the default sheet name is Sheet1
<code>cell(cell)</code>	specify the Excel upper-left cell as the starting position to export the table; the default is cell(A1)
<code>modify</code>	modify Excel file
<code>noopen</code>	do not open Excel file in memory

`noopen` does not appear in the dialog box.

<i>tex_options</i>	Description
<code>append</code>	append to an existing file
<code>tableonly</code>	export only the table to the specified file

<i>smcl_option</i>	Description
<code>append</code>	append to an existing file

<i>txt_option</i>	Description
<code>append</code>	append to an existing file

<i>md_option</i>	Description
<code>append</code>	append to an existing file

fontfamily specifies a valid font family.

unit may be *in* (inch), *pt* (point), or *cm* (centimeter). An inch is equivalent to 72 points and 2.54 centimeters. The default is *pt*.

variant may be *allcaps*, *smallcaps*, or *normal*.

variant (*allcaps*) changes the text to all uppercase letters; applicable when publishing items from a collection to Microsoft Word, PDF, \LaTeX , and HTML files.

variant (*smallcaps*) changes the text to use large capitals for uppercase letters and smaller capitals for lowercase letters; applicable when publishing items from a collection to Microsoft Word, \LaTeX , and HTML files.

variant (*normal*) changes the font variant back to normal and leaves the capitalization unchanged from the original text; applicable when publishing items from a collection to Microsoft Word, PDF, \LaTeX , and HTML files.

smcl specifies the name of the SMCL directive to render text for SMCL output. The supported SMCL directives are *input*, *error*, *result*, and *text*.

latex specifies the name of a \LaTeX macro to render text for \LaTeX output. Example \LaTeX macro names are *textbf*, *textsf*, *textrm*, and *texttt*. Custom \LaTeX macros are also allowed. If *text* is to be rendered in a cell, title, or note, then *latex* is translated to the following when you export to \LaTeX :

```
\latex {text}
```

sspec is

```
[ background(bgcolor) foreground(fgcolor) pattern(fpattern) ]
```

bgcolor specifies the background color.

fgcolor specifies the foreground color.

fpattern specifies the fill pattern. A complete list of fill patterns is shown in the [Appendix](#).

bgcolor, *fgcolor*, and *color* may be one of the colors listed in the [Appendix](#); a valid RGB value in the form *### ## #*, for example, 171 248 103; or a valid RRGGBB hex value in the form *#####*, for example, ABF867.

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</code> (<i>rowvar</i>)	percentages or proportions across rows
<code>across</code> (<i>colvar</i>)	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.

Title

`title(string)` adds the text *string* as a title to the table.

`titlestyles(text_styles)` changes the style for the table title. *text_styles* are the following:

`font([fontfamily] [, size(# [unit]) color(color) variant(variant) [no]bold [no]italic [no]strikeout [no]underline])` specifies the font style. These font style properties are applicable when exporting the table to Microsoft Word, Microsoft Excel, PDF, HTML, and \LaTeX files, unless otherwise specified.

fontfamily specifies a valid font family. This font style property is applicable when publishing items from a collection to Microsoft Word, Microsoft Excel, PDF, and HTML files.

`size(# [unit])` specifies the font size as a number optionally followed by units. This font style property is applicable when publishing items from a collection to Microsoft Word, Microsoft Excel, PDF, and HTML files.

`color(color)` specifies the text color.

`variant(variant)` specifies the font variant and capitalization.

`bold` and `nobold` specify the font weight. `bold` changes the font weight to bold; `nobold` changes the font weight back to normal.

`italic` and `noitalic` specify the font style. `italic` changes the font style to italic; `noitalic` changes the font style back to normal.

`strikeout` and `nostrikeout` specify whether to add a strikeout mark to the title. `strikeout` adds a strikeout mark to the title; `nostrikeout` changes the title back to normal.

`underline` and `nounderline` specify whether to underline the table title. `underline` adds a single line under the title; `nounderline` removes the underline.

Only one of `strikeout` or `underline` is allowed when publishing to HTML files.

`smcl(smcl)` specifies how to render the table title for SMCL output. This style property is applicable only when publishing items from a collection to a SMCL file.

`latex(latex)` specifies how to render the table title for \LaTeX output. This style property is applicable only when publishing items from a collection to a \LaTeX file.

`shading(sspec)` sets the background color, foreground color, and fill pattern. The background color is applicable when exporting the table to Microsoft Word, Microsoft Excel, PDF, HTML, and \LaTeX files. The foreground color and fill pattern are applicable when exporting the table to Microsoft Word and Microsoft Excel.

Notes

`note(string)` adds the text *string* as a note to the table. `note()` may be specified multiple times to add multiple notes. Each note is placed on a new line.

`notestyles(text_styles)` changes the style for the table notes. *text_styles* are the following:

`font([fontfamily] [, size(# [unit]) color(color) variant(variant) [no]bold [no]italic [no]strikeout [no]underline])` specifies the font style. These font style properties are applicable when exporting the table to Microsoft Word, Microsoft Excel, PDF, HTML, and L^AT_EX files, unless otherwise specified.

fontfamily specifies a valid font family. This font style property is applicable when publishing items from a collection to Microsoft Word, Microsoft Excel, PDF, and HTML files.

`size(# [unit])` specifies the font size as a number optionally followed by units. This font style property is applicable when publishing items from a collection to Microsoft Word, Microsoft Excel, PDF, and HTML files.

`color(color)` specifies the text color.

`variant(variant)` specifies the font variant and capitalization.

`bold` and `nobold` specify the font weight. `bold` changes the font weight to bold; `nobold` changes the font weight back to normal.

`italic` and `noitalic` specify the font style. `italic` changes the font style to italic; `noitalic` changes the font style back to normal.

`strikeout` and `nostrikeout` specify whether to add a strikeout mark to the notes. `strikeout` adds a strikeout mark to the note; `nostrikeout` changes the note back to normal.

`underline` and `nounderline` specify whether to underline the table notes. `underline` adds a single line under the notes; `nounderline` removes the underline.

Only one of `strikeout` or `underline` is allowed when publishing to HTML files.

`smcl(smcl)` specifies how to render the table notes for SMCL output. This style property is applicable only when publishing items from a collection to a SMCL file.

`latex(latex)` specifies how to render the table notes for L^AT_EX output. This style property is applicable only when publishing items from a collection to a L^AT_EX file.

`shading(sspec)` sets the background color, foreground color, and fill pattern. The background color is applicable when exporting the table to Microsoft Word, Microsoft Excel, PDF, HTML, and L^AT_EX files. The foreground color and fill pattern are applicable when exporting the table to Microsoft Word and Microsoft Excel.

Export

`export(filename.suffix[, export_opts])` exports the table to the specified file. *export_opts* are the following:

`as(fileformat)` specifies the file format to which the table is to be exported. This option is rarely specified because, by default, `table` determines the format from the suffix of the file being created.

`replace` permits `table` to overwrite an existing file.

`noisily` specifies that `table` show the commands used to export the table to Microsoft Word, Microsoft Excel, and PDF files. The `putdocx`, `putexcel`, or `putpdf` command used to export the table will be displayed.

`dofile(filename[, replace])` specifies that `table` save to *filename* the commands used to export the table to Microsoft Word, Microsoft Excel, and PDF files.

If *filename* already exists, it can be overwritten by specifying `replace`. If *filename* is specified without an extension, `.do` is assumed.

`append` specifies that `table` append the table to an existing file.

This option is applicable when you export the table to an HTML, a \LaTeX , a SMCL, a `txt`, or a Markdown file. When you export to HTML and \LaTeX files, the `append` option implies the `tableonly` option. Furthermore, when you export to HTML files, if the target CSS file already exists, `table` will also append to it.

`tableonly` specifies that only the table be exported to the specified HTML or \LaTeX document. By default, `table` produces complete HTML and \LaTeX documents.

When you export to an HTML file, if the `cssfile()` option is not specified, a CSS filename is constructed from *filename*, with the extension replaced with `.css`.

`cssfile(cssfile)` specifies that `table` define the styles in *cssfile* instead of *filename* when you export to HTML.

`prefix(prefix)` specifies that `table` use *prefix* to identify style classes when you export to HTML.

`sheet(sheetname [, replace])` saves to the worksheet named *sheetname*. For more information about this option, see [\[RPT\] putexcel](#).

`cell(cell)` specifies an Excel upper-left cell as the starting position to publish the table. The default is `cell(A1)`.

`modify` permits `putexcel` set to modify an Excel file. For more information about this option, see [\[RPT\] putexcel](#).

`noopen` prevents `putexcel` from opening the Excel file in memory for modification. It does not appear in the dialog box. For more information about this option, see [\[RPT\] putexcel](#).

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

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/r19/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)
```

	Region				Total
	NE	MW	S	W	
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))
```

	Region				Total
	NE	MW	S	W	
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))
```

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

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 \LaTeX , Word, Excel, HTML, and a variety of other formats by using the `export()` option.

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". In *Vital and Health Statistics*, ser. 1, no. 15. Hyattsville, MD: National Center for Health Statistics.

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 © 1985–2025 StataCorp LLC, College Station, TX, USA. All rights reserved.

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

