

**dtable** — Create a table of descriptive statistics

[Description](#)[Quick start](#)[Menu](#)[Syntax](#)[Options](#)[Remarks and examples](#)[Methods and formulas](#)[Appendix](#)[Acknowledgments](#)[Reference](#)[Also see](#)

## Description

The `dtable` command allows you to easily create a table of descriptive (summary) statistics, commonly known as “Table 1”. Optionally, you can add  $p$ -values from test statistics comparing groups on summary statistics. You can also create a table complete with a title, notes, and more, and then export it to a variety of file types.

## Quick start

Create a table describing continuous variables `x1` and `x2` and factor variables `f1` and `f2`

```
dtable x1 x2 i.f1 i.f2
```

Same as above, but report statistics for each level of variable `group`, with and without the total sample

```
dtable x1 x2 i.f1 i.f2, by(group)
dtable x1 x2 i.f1 i.f2, by(group, nototals)
```

Same as above, but test each variable for equality between groups using linear regressions for continuous variables and Pearson  $\chi^2$  tests for factor variables

```
dtable x1 x2 i.f1 i.f2, by(group, tests)
```

Same as above, but rather than putting sample frequencies in the first row, put them in the column header under each group label

```
dtable x1 x2 i.f1 i.f2, by(group, tests) sample(, place(seplabels))
```

Create a table of the median and interquartile interval for variables `x1-x4`

```
dtable, continuous(x1-x4, statistics(q2 iqi)) ///
define(iqi = q1 q3) sformat("[%s]" iqi)
```

Same as above, but add the Kruskal–Wallis test for equality between levels of variable `group`

```
dtable, continuous(x1-x4, statistics(q2 iqi) test(kwallis)) ///
define(iqi = q1 q3) sformat("[%s]" iqi) by(group, tests)
```

Create a table to describe variables using survey weights

```
dtable x* i.f*, svy
```

Same as above, but restrict the sample to the subpopulation where `group` equals 4

```
dtable x* i.f*, svy subpop(if group==4)
```

## Menu

Statistics > Summaries, tables, and tests > Table of descriptive statistics

## Syntax

<i>options</i>	Description
<b>Main</b>	
<code>by(<i>varname</i> [, <i>by_opts</i>])</code>	group results by categories of <i>varname</i>
<code>column(<i>col_opts</i>)</code>	change column header styles and titles
<b>Survey</b>	
<code>svy</code>	use survey settings
<code>subpop([<i>varname</i>] [<i>if</i>])</code>	identify a survey subpopulation
<b>Sample</b>	
<code>nosample</code>	suppress sample frequency statistics
<code>sample([<i>title</i>] [, <i>samp_opts</i>])</code>	select sample frequency statistics
<b>Continuous</b>	
<code>continuous([<i>varlist<sub>c</sub></i>] [, <i>cont_opts</i>])</code>	select continuous variable statistics
<b>Factors</b>	
<code>factor([<i>varlist<sub>f</sub></i>] [, <i>fact_opts</i>])</code>	select factor-variable statistics
<b>Composite</b>	
<code>define(<i>name</i> = <i>elements</i> [, <i>def_opts</i>])</code>	define a composite result
<b>Formats</b>	
<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
<b>Title</b>	
<code>title(<i>string</i>)</code>	add table title
<code>titlestyles(<i>text_styles</i>)</code>	change table title styles
<b>Notes</b>	
<code>note(<i>string</i>)</code>	add table note
<code>notestyles(<i>text_styles</i>)</code>	change table note styles
<b>Export</b>	
<code>export(<i>filename.suffix</i> [, <i>export_opts</i>])</code>	export table
<b>Options</b>	
<code>[no] varlabel</code>	display or suppress variable names or labels
<code>[no] fvlablel</code>	display or suppress factor values or labels
<code>[no] listwise</code>	specify how to handle missing values
<code>halign(<i>hvalue</i>)</code>	specify horizontal alignment of item cells
<code>name(<i>cname</i>)</code>	work with collection <i>cname</i> ; default is <code>name(DTable)</code>
<code>replace</code>	replace the 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>warn</code>	show collect warnings

---

warn does not appear in the dialog box.

<i>by_opts</i>	Description
[no] tests	show or suppress tests across groups
[no] testnotes	show or suppress notes about tests across groups
[no] totals	show or suppress statistics for the total sample
[no] missing	show or suppress numeric missing values in <i>varname</i>

<i>col_opts</i>	Description
summary([ <i>title</i> ] [, <i>hcol_opts</i> ])	change header styles for the summary column
by([ <i>by_style</i> ] [, <i>by_hcol_opts</i> ])	change header styles for the by() variable columns; <i>by_style</i> may be label, name, or hide
test([ <i>title</i> ] [, <i>hcol_opts</i> ])	change header styles for the test column
total([ <i>title</i> ] [, <i>hcol_opts</i> ])	change header styles for the total sample column

<i>hcol_opts</i>	Description
[no] hide	show or hide the column header title
halign( <i>hvalue</i> )	specify horizontal alignment of column header cells
<i>text_styles</i>	change column header text styles

nohide and hide do not appear in the dialog box for summary().

<i>by_hcol_opts</i>	Description
[no] fvlabel	display by() variable values or value labels
halign( <i>hvalue</i> )	specify horizontal alignment of column header cells
<i>text_styles</i>	change column header text styles

<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 <sup>A</sup> T <sub>E</sub> 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
underline( <i>upattern</i> )	specify underline pattern for text

## 4 **dtable** — Create a table of descriptive statistics

---

<i>samp_opts</i>	Description
<b>statistics</b> ( <i>nstats</i> )	select sample frequency statistics
<b>place</b> ( <i>place</i> )	select location for sample frequency statistics; <i>place</i> may be <b>items</b> , <b>inlabels</b> , or <b>seplabels</b>
<i>text_styles</i>	change row header text styles

---

<i>cont_opts</i>	Description
<b>statistics</b> ( <i>cstats</i> )	select continuous variable statistics
<b>test</b> ( <i>cctest</i> )	select continuous variable test
<i>text_styles</i>	change row header text styles

---

<i>fact_opts</i>	Description
<b>statistics</b> ( <i>fstats</i> )	select factor-variable statistics
<b>test</b> ( <i>ftest</i> )	select factor-variable test
<i>text_styles</i>	change row header text styles

---

<i>def_opts</i>	Description
<b>delimiter</b> ( <i>char</i> )	use character as delimiter between <i>elements</i>
<b>[no]trim</b>	preserve or trim extra spaces from numeric formats
<b>[no]override</b>	preserve or override trim property of elements
<b>replace</b>	redefine <i>name</i> if it already exists

---

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

---

<i>export_opts</i>	Description
<b>as</b> ( <i>fileformat</i> )	specify document type
<b>replace</b>	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
<b>noisily</b>	show the putdocx commands used to export to the Microsoft Word file
<b>dofile</b> ( <i>filename</i> [, <b>replace</b> ])	save the putdocx commands used for exporting to the named do-file

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

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

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

`noopen` does not appear in the dialog box.

## 6 **dtable** — Create a table of descriptive statistics

<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

*varlist* may contain `i.` notation, including level selection. Time-series operators and interactions are not allowed.

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

*results* is a list of names taken from *nstats*, *cstats*, *ctest*, *fstats*, *frest*, and composite results.

*hvalue* may be `left`, `center`, or `right`.

*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, L<sup>A</sup>T<sub>E</sub>X, 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, L<sup>A</sup>T<sub>E</sub>X, 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, L<sup>A</sup>T<sub>E</sub>X, and HTML files.

*upattern* may be any of the patterns listed in the [Appendix](#). For example, `underline(none)` removes the underline from the text, and `underline(single)` underlines the text. All other *upatterns* are available only when publishing items from a collection to Microsoft Word.

*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 L<sup>A</sup>T<sub>E</sub>X macro to render text for L<sup>A</sup>T<sub>E</sub>X output. Example L<sup>A</sup>T<sub>E</sub>X macro names are `textbf`, `textsf`, `textrm`, and `texttt`. Custom L<sup>A</sup>T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X:

```
\latex {text}
```

*spec* 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.

*sfmt* is the specification for a string format in option `sformat()` and may contain a mix of text and `%s`. Here `%s` refers to the statistic value that is formatted as specified using `nformat()`. The text will be placed around the statistic as it is placed around `%s` in this option. For instance, to place parentheses around the statistic, you can specify `sformat("(%s")`.

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

## Options

Main

`by(varname [, by_opts])` specifies that the statistics be displayed separately for each unique value of *varname*; *varname* may be numeric or string.

*by\_opts* are the following:

`tests` and `notests` specify whether to show the tests comparing groups. `tests` displays the tests comparing groups; `notests` suppresses them.

When `tests` is combined with the `missing` suboption, the test will be performed with the missing values composing an additional group. By default, missing values are excluded from tests.

`testnotes` and `notestnotes` specify whether to show a note detailing each test comparing groups. `testnotes` displays the notes; `notestnotes` suppresses them.

`totals` and `nototals` specify whether to show the statistics for the total sample. `totals` displays the statistics for the total sample; `nototals` suppresses them.

`missing` and `nomissing` specify whether to show numeric missing values in *varname*. `missing` displays the numeric missing values; `nomissing` suppresses them.

When `missing` is combined with the `tests` suboption, missing values will also be included in the test performed. By default, missing values are excluded from tests.

`column(col_opts)` specifies the column header styles and titles. *col\_opts* are one or more of the following.

`summary([title][, [no]hide halign(hvalue) text_styles])` specifies the column header and styles when global option `by()` is not specified. The default *title* is `Summary`.

`hide` and `nohide` specify whether to show the title. `hide` suppresses the title; `nohide` displays the title.

`halign(hvalue)` specifies how the column header cell is horizontally aligned. *hvalue* may be `left`, `center`, or `right`.

*text\_styles* affect the text styles for the column header.

`by([by_style] [, [no]fvlabel halign(hvalue) text_styles])` specifies the `by()` variable's column header and styles.

*by\_style* specifies the column header style for the `by()` variable. *by\_style* may be `label`, `name`, or `hide`.

`label` specifies that the `by()` variable's label be used. If the `by()` variable does not have a label, then the name is used.

`name` specifies that the `by()` variable's name be used.

`hide` specifies that the `by()` variable not be shown in the column header.

`fvlabel` and `nofvlabel` specify whether value labels for the levels of the `by()` variable should be displayed. `fvlabel` displays the value labels; `nofvlabel` displays the values.

`halign(hvalue)` specifies how the column header cells are horizontally aligned. *hvalue* may be `left`, `center`, or `right`.

*text\_styles* affect the text styles for the column header.

`test([title] [, [no]hide halign(hvalue) text_styles])` specifies the column header and text styles for the test column. The default *title* is `Test`.

`hide` and `nohide` specify whether to show the title for the test column. `hide` suppresses the title; `nohide` displays the title.

`halign(hvalue)` specifies how the test column header cell is horizontally aligned. *hvalue* may be `left`, `center`, or `right`.

*text\_styles* affect the text styles for the test column header.

`total([title] [, [no]hide halign(hvalue) text_styles])` specifies the column header and text styles for the total sample. The default *title* is `Total`.

`hide` and `nohide` specify whether to show the title for the total sample. `hide` suppresses the title; `nohide` displays the title.

`halign(hvalue)` specifies how the total sample column header cell is horizontally aligned. *hvalue* may be `left`, `center`, or `right`.

*text\_styles* affect the text styles for the total sample column header.

*text\_styles* are `font([fontfamily] [, font_opts])`, `smcl(smcl)`, `latex(latex)`, and `shading(sspec)`.

`font([fontfamily] [, size(# [unit]) color(color) variant(variant) [no]bold [no]italic [no]strikeout [no]underline underline(upattern))])` specifies the font style. These font style properties are applicable when exporting the table to Microsoft Word, Microsoft Excel, PDF, HTML, and L<sup>A</sup>T<sub>E</sub>X 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 text. `strikeout` adds a strikeout mark to the text; `nostrikeout` changes the text back to normal.

`underline`(*upattern*), `underline`, and `nounderline` specify how to underline the text.

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

`smcl`(*smcl*) specifies how to render the text 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 text for L<sup>A</sup>T<sub>E</sub>X output. This style property is applicable only when publishing items from a collection to a L<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X files. The foreground color and fill pattern are applicable when exporting the table to Microsoft Word and Microsoft Excel.

---

#### Survey

`svy` specifies that `dtable` use the survey design information from `svyset`. This means all statistics will be computed using the specified survey weights, where applicable. This option changes the list of supported tests in `continuous()` and `factor()`.

`subpop`( [*varname*] [*if*] ) specifies that estimates be computed for the single subpopulation identified by the observations for which *varname*  $\neq$  0 and that also meet the `if` conditions. Typically, *varname* = 1 defines the subpopulation, and *varname* = 0 indicates observations not belonging to the subpopulation. For observations whose subpopulation status is uncertain, *varname* should be set to a missing value; such observations are dropped from the estimation sample. This option implies the `svy` option.

See [\[SVY\] Subpopulation estimation](#) for more information.

## Sample

`nosample` and `sample()` control the display of sample frequency statistics.

`nosample` prevents reporting any sample frequency statistic.

`sample([title][, statistics(nstats) place(place) text_styles])` specifies sample frequency statistics, their row header title, and how they are composed in the table.

*title* specifies the row header for the sample frequency statistics. The default *title* is `N`.

`statistics(nstats)` specifies the sample frequency statistics to report.

*nstats* may include the following sample frequency statistics and any composite result defined from them.

<i>nstats</i>	Definition
<u>frequency</u>	frequency
<u>sumw</u>	sum of weights
<u>percent</u>	percentage
<u>proportion</u>	proportion
<u>rawpercent</u>	unweighted percentage
<u>rawproportion</u>	unweighted proportion

Without weights, the default sample frequency statistic is `frequency`.

With weights, the default sample frequency statistic is `sumw`.

With option `by()`, `percent` is added to the default sample frequency statistic.

`place(place)` controls where the sample frequency statistics are reported in the table.

`place(items)`, the default, specifies that the sample frequency statistics be reported in the first row of items in the table.

`place(inlabels)` specifies that the sample frequency statistics be added to the column header labels.

`place(seplabels)` specifies that the sample frequency statistics be stacked in the column header as separate labels.

*text\_styles* specifies the text styles for *title*.

## Continuous

`continuous([varlistc][, statistics(cstats) test(ctest) text_styles])` specifies continuous variables, their row header text styles, their statistics, and their test. If *varlist<sub>c</sub>* is not specified, then the specified text styles, statistics, and test become the default for all continuous variables in *varlist*.

This option is repeatable, and when multiple text styles, statistics, or tests apply to a variable, the rightmost specification is applied.

`statistics(cstats)` specifies the statistics to report for *varlist<sub>c</sub>*.

*cstats* may contain the following continuous variable statistics and any composite result defined from them.

<i>cstats</i>	Description
<code>mean</code>	mean
<code>semean</code>	standard error of the mean
<code>sebinomial</code>	standard error of the mean, binomial
<code>sepoisson</code>	standard error of the mean, Poisson
<code>variance</code>	variance
<code>sd</code>	standard deviation
<code>skewness</code>	skewness
<code>kurtosis</code>	kurtosis
<code>cv</code>	coefficient of variation
<code>svycv</code>	coefficient of variation (svy)
<code>geomean</code>	geometric mean
<code>geosd</code>	geometric standard deviation
<code>count</code>	number of nonmissing values
<code>median</code>	median
<code>p#</code>	#th percentile
<code>q1</code>	first quartile
<code>q2</code>	second quartile
<code>q3</code>	third quartile
<code>iqr</code>	interquartile range
<code>min</code>	minimum value
<code>max</code>	maximum value
<code>range</code>	range
<code>first</code>	first value
<code>last</code>	last value
<code>firstnm</code>	first nonmissing value
<code>lastnm</code>	last nonmissing value
<code>total</code>	total
<code>rawtotal</code>	unweighted total

The default continuous variable statistics are `mean` and `sd`.

`test(ctest)` specifies that `ctest` be computed for the continuous variables in `varlistc`. This option has no effect if the `by()` option is not specified.

`ctest` is one of the following:

<i>ctest</i>	Description
<code>regress</code>	main effects test from a linear regression
<code>poisson</code>	main effects test from a Poisson regression
<code>lnormal</code>	main effects test from a log-normal regression
<code>kwallis</code>	Kruskal–Wallis rank test
<code>kwallis_ties</code>	Kruskal–Wallis rank test, adjusted for ties
<code>none</code>	suppress the test

`kwallis` and `kwallis_ties` are not allowed with `weights` or the `svy` option.

The default test for continuous variables is `regress`.

For nonsurvey data, `regress` is equivalent to a one-way ANOVA, and when the `by()` variable has two levels, `regress` is equivalent to a pooled *t* test.

`text_styles` specifies the row header text styles for `varlistc`.

#### Factors

`factor([varlistf] [, statistics(fstats) test(ftest) text_styles])` specifies factor variables, their row header text styles, their statistics, and their test. If `varlistf` is not specified, then the specified text styles, statistics, and test become the default for all factor variables in `varlist`.

This option is repeatable, and when multiple text styles, statistics, or tests apply to a variable, the rightmost specification is applied.

`statistics(fstats)` specifies the statistics to report for `varlistf`.

`fstats` may contain the following factor-variable statistics and any composite result defined from them.

<i>fstats</i>	Definition
<code>fvfrequency</code>	frequency of each factor-variable level
<code>fvpercent</code>	percentage within each factor-variable level
<code>fvproportion</code>	proportion within each factor-variable level
<code>fvrawfrequency</code>	unweighted frequency of each factor-variable level
<code>fvrawpercent</code>	unweighted percentage within each factor-variable level
<code>fvrawproportion</code>	unweighted proportion within each factor-variable level

The default factor-variable statistics are `fvfrequency` and `fvpercent`.

`test(ftest)` specifies that *ftest* be computed for the factor variables in *varlist<sub>f</sub>*. This option has no effect if the `by()` option is not specified.

*ftest* is one of the following:

<i>ftest</i>	Description
<code>pearson</code>	Pearson's $\chi^2$ test
<code>fisher</code>	Fisher's exact test
<code>lrchi2</code>	likelihood-ratio $\chi^2$ test
<code>gamma</code>	Goodman and Kruskal's gamma
<code>kendall</code>	Kendall's $\tau_b$
<code>cramer</code>	Cramér's $V$
<code>svylr</code>	survey-adjusted likelihood-ratio test
<code>svywald</code>	survey-adjusted Wald test
<code>svyllwald</code>	survey-adjusted log-linear Wald test
<code>none</code>	suppress the test

`fisher`, `lrchi2`, `gamma`, `kendall`, and `cramer` are not allowed with `aweight`s, `iweight`s, `pweight`s, or the `svy` option.

`svylr`, `svywald`, and `svyllwald` are allowed only with the `svy` option.

The default test for factor variables is `pearson`.

`text_styles` specifies the row header text styles for *varlist<sub>f</sub>*.

#### Composite

`define(name=elements[, def_opts])` defines a composite result. Composite results give more control over how statistics are composed and formatted in a table cell.

*name* is the name you choose for the new composite result.

*elements* is a list of statistics and other composite results.

*def\_opts* are `delimiter(char)`, `[no]trim`, `[no]override`, and `replace`.

`delimiter(char)` changes the delimiter between elements. The default is `delimiter(" ")`.

`notrim` and `trim` control the handling of extra spaces caused by numeric formats applied to the elements. `notrim` preserves the extra spaces; `trim`, the default, removes the extra spaces.

`nooverride` and `override` control handling of the trim property when an element is a composite result. `nooverride`, the default, does not change the trim property of elements; `override` applies the specified trim property to all elements.

`replace` permits you to redefine *name* if it already exists.

#### 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* is a list of names taken from *nstats*, *cstats*, *ctest*, *fstats*, *ftest*, and composite results.

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 factor variables specified in `varlist` or the `factor()` option. 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)" fvpercent percent)`.

*results* is a list of names taken from *nstats*, *cstats*, *ctest*, *fstats*, *ftest*, and composite results.

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%%" fvpercent 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 L<sup>A</sup>T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X output. This style property is applicable only when publishing items from a collection to a L<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X output. This style property is applicable only when publishing items from a collection to a L<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X 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, `dtable` determines the format from the suffix of the file being created.

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

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

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

`tableonly` specifies that only the table be exported to the specified HTML or  $\text{\LaTeX}$  document. By default, `dtable` produces complete HTML and  $\text{\LaTeX}$  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 `dtable` define the styles in *cssfile* instead of *filename* when you export to HTML.

`prefix(prefix)` specifies that `dtable` 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 on the dialog box. For more information about this option, see [RPT] [putexcel](#).

## Options

`varlabel` and `novarlabel` specify whether variable labels for the continuous and factor variables should be displayed. `varlabel` displays the variable label; `novarlabel` displays variable names.

`fvlablel` and `nofvlablel` specify whether value labels for the factor variables should be displayed. `fvlablel` displays value labels; `nofvlablel` displays the values.

`listwise` and `nolistwise` specify how to handle missing values across the continuous and factor variables. `listwise` handles missing values through listwise deletion, meaning that the entire observation is omitted from the sample if any continuous or factor variable is missing for that



observation; `nolistwise` specifies that `dtable` omit an observation only if all the continuous and factor variables are missing for that observation.

`halign(hvalue)` specifies how the item cells are horizontally aligned. *hvalue* may be `left`, `center`, or `right`.

`name(cname)` specifies the collection for `dtable` to work with. The default is `name(DTable)`.

`replace` permits `dtable` to overwrite the existing collection. This option is implied for `name(DTable)`.

`label(filename)` specifies the *filename* containing the collection labels to use for your table. Labels in *filename* will be loaded for the table, and default labels will be used for any labels not specified in *filename*.

`style(filename[, override])` specifies the *filename* containing the collection styles to use for your table. This might be a style you saved with `collect style save` or a `predefined style` shipped with Stata. The `dtable` collection styles will be discarded, and only the collection styles in *filename* will be applied. Note that the layout specification saved in *filename* will not be applied; `dtable` will always use its predefined layout.

If you prefer the `dtable` 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(dtable_style)`; see [TABLES] `set dtable_style`.

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

`warn` specifies that `dtable` display warnings from `collect`. By default, these warnings are suppressed.

## Remarks and examples

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

Remarks are presented under the following headings:

- [Introduction](#)
- [A first example](#)
- [Describe variables across groups](#)
- [Survey data](#)
- [Save your style choices for next time](#)
- [Composite results](#)
- [The default style](#)

## Introduction

`dtable` allows you to easily create a table of summary statistics and export it to a variety of file types, without any knowledge of the collection system. You can specify the continuous and factor variables that you want described in your table. You can also customize the table by formatting the results, changing the variable-specific statistics, adding a variable to identify sample groups, testing each variable for equality between groups, adding a title and notes, and more.

In most cases, you will use `dtable` to easily create a table and export it to another format. However, you can customize the table beyond the options that are available with `dtable`. When you issue a `dtable` command, the results are stored in a collection called `DTable`. This collection is replaced with each new `dtable` command, unless you specify a different name for the collection in the `name()` option. You can make additional changes to the collection with the `collect` suite of commands. To learn more about the `collect` commands, see [TABLES] [Intro](#) and the entries discussed therein.

## A first example

In its simplest specification, you type `dtable` and a list of variables you want described in your table. For our first example, we use data from the Second National Health and Nutrition Examination Survey (NHANES II) (McDowell et al. 1981). Our table describes continuous variables for systolic blood pressure, age, weight, and factor variables for race and health status.

```
. use https://www.stata-press.com/data/r18/nhanes2l
(Second National Health and Nutrition Examination Survey)
. dtable bpsystol age weight i.race i.hlthstat
```

Summary	
N	10,351
Systolic blood pressure	130.882 (23.333)
Age (years)	47.580 (17.215)
Weight (kg)	71.898 (15.356)
Race	
White	9,065 (87.6%)
Black	1,086 (10.5%)
Other	200 (1.9%)
Health status	
Excellent	2,407 (23.3%)
Very good	2,591 (25.1%)
Good	2,938 (28.4%)
Fair	1,670 (16.2%)
Poor	729 (7.1%)

The default continuous variable statistics are the mean and standard deviation. The default factor-variable statistics are the factor-level frequencies and percentages. You can change the statistics reported for each continuous and factor variable; see the `continuous()` and `factor()` options for the list of supported statistics.

Many options are also available to change the formatting and style of your table. Here we set the format of means and standard deviations to show two decimal places, remove the first row of the table with the overall sample size, and change the column header to be more descriptive of the table results.

```
. dtable bpsystol age weight i.race i.hlthstat,
> nformat(%16.2fc mean sd)
> nosample
> column(summary(M(SD) / n(%)))
```

M(SD) / n(%)	
Systolic blood pressure	130.88 (23.33)
Age (years)	47.58 (17.21)
Weight (kg)	71.90 (15.36)
Race	
White	9,065 (87.6%)
Black	1,086 (10.5%)
Other	200 (1.9%)
Health status	
Excellent	2,407 (23.3%)
Very good	2,591 (25.1%)
Good	2,938 (28.4%)
Fair	1,670 (16.2%)
Poor	729 (7.1%)

Additionally, you can complete your table with a title and notes and even export it to another format. We now add a title and export our table to the file `mydtable.html`.

```
. dtable bpsystol age weight i.race i.hlthstat,
>       nformat(%16.2fc mean sd)
>       nosample
>       column(summary(M(SD) / n(%)))
>       title(Table 1)
>       export(mydtable.html)
```

Table 1

	M(SD) / n(%)
Systolic blood pressure	130.88 (23.33)
Age (years)	47.58 (17.21)
Weight (kg)	71.90 (15.36)
Race	
White	9,065 (87.6%)
Black	1,086 (10.5%)
Other	200 (1.9%)
Health status	
Excellent	2,407 (23.3%)
Very good	2,591 (25.1%)
Good	2,938 (28.4%)
Fair	1,670 (16.2%)
Poor	729 (7.1%)

(collection DTable exported to file `mydtable.html`)

We could also export this table to a Microsoft Word, Microsoft Excel, L<sup>A</sup>T<sub>E</sub>X, Markdown, SMCL, PDF, or plain-text file by specifying the appropriate [file extension](#).

## Describe variables across groups

Describing variables across samples in your dataset is easy with the `by()` option. In the following, we revisit our original table, but we add the `by()` option to get summary statistics from the urban, rural, and total samples.

```
. dtable bpsystol age weight i.race i.hlthstat, by(rural) sample(N (%))
```

	Urban	Rural Rural	Total
N (%)	6,548 (63.3%)	3,803 (36.7%)	10,351 (100.0%)
Systolic blood pressure	130.460 (23.526)	131.607 (22.980)	130.882 (23.333)
Age (years)	46.835 (17.484)	48.862 (16.666)	47.580 (17.215)
Weight (kg)	71.427 (15.363)	72.708 (15.314)	71.898 (15.356)
Race			
White	5,419 (82.8%)	3,646 (95.9%)	9,065 (87.6%)
Black	968 (14.8%)	118 (3.1%)	1,086 (10.5%)
Other	161 (2.5%)	39 (1.0%)	200 (1.9%)
Health status			
Excellent	1,609 (24.6%)	798 (21.0%)	2,407 (23.3%)
Very good	1,713 (26.2%)	878 (23.1%)	2,591 (25.1%)
Good	1,878 (28.7%)	1,060 (27.9%)	2,938 (28.4%)
Fair	950 (14.5%)	720 (19.0%)	1,670 (16.2%)
Poor	389 (5.9%)	340 (9.0%)	729 (7.1%)

Next, we add the `tests` suboption in the `by()` option to report tests of equality between samples and the `nototals` option to suppress the column of statistics from the total sample. We also specify that we would like sample statistics to be in the column headers rather than the first row of the table using the `sample()` option. Here we also add the `column(by(hide))` option to suppress the redundant label from variable `rural` and add notes that detail what is being reported.

```
. dtable bpsystol age weight i.race i.hlthstat,
> by(rural, nototals tests)
> column(by(hide))
> sample(, place(seplabels))
> note(Mean (Standard deviation): p-value from a pooled t-test.)
> note(Frequency (Percent%): p-value from Pearson test.)
note: using test regress across levels of rural for bpsystol, age, and
weight.
note: using test pearson across levels of rural for race and hlthstat.
```

	Urban 6,548 (63.3%)	Rural 3,803 (36.7%)	Test
Systolic blood pressure	130.460 (23.526)	131.607 (22.980)	0.016
Age (years)	46.835 (17.484)	48.862 (16.666)	<0.001
Weight (kg)	71.427 (15.363)	72.708 (15.314)	<0.001
Race			
White	5,419 (82.8%)	3,646 (95.9%)	<0.001
Black	968 (14.8%)	118 (3.1%)	
Other	161 (2.5%)	39 (1.0%)	
Health status			
Excellent	1,609 (24.6%)	798 (21.0%)	<0.001
Very good	1,713 (26.2%)	878 (23.1%)	
Good	1,878 (28.7%)	1,060 (27.9%)	
Fair	950 (14.5%)	720 (19.0%)	
Poor	389 (5.9%)	340 (9.0%)	

Mean (Standard deviation): p-value from a pooled t-test.

Frequency (Percent%): p-value from Pearson test.

For continuous variables, the `Test` column reports the  $p$ -value from linear regressions, which are equivalent to pooled  $t$  tests because `rural` has two categories. For factor variables, the `Test` column reports the  $p$ -values from Pearson  $\chi^2$  tests.

## Survey data

`dtable` supports survey data by using the survey data characteristics in your dataset when you specify the `svy` option. Your data must be `svyset` prior to calling `dtable` with the `svy` option. Our data already have survey data characteristics.

```
. svyset
Sampling weights: finalwgt
                  VCE: linearized
                  Single unit: missing
                  Strata 1: strata
Sampling unit 1: psu
                  FPC 1: <zero>
```

The following table reports the survey weighted frequency, mean, standard deviation, factor-level frequency, and factor-level percent of the specified variables.

```
. dtable bpsystol age weight i.race i.hlthstat, svy
```

Summary	
N	117,157,513
Systolic blood pressure	126.946 (21.401)
Age (years)	42.253 (15.502)
Weight (kg)	71.901 (15.433)
Race	
White	102,999,549 (87.9%)
Black	11,189,236 (9.6%)
Other	2,968,728 (2.5%)
Health status	
Excellent	32,187,335 (27.5%)
Very good	32,176,310 (27.5%)
Good	32,715,395 (28.0%)
Fair	14,380,261 (12.3%)
Poor	5,537,956 (4.7%)

Option `by()` is allowed with `svy`. `dtable` also has options for tests of equality between groups that are allowed with `svy`. These tests account for the survey data characteristics in your data. In the following, we revisit the urban and rural summary table, but we add the `svy` option, a title, and some notes. We also add the `column(by(hide))` option to suppress the redundant label from variable `rural` and add the `test()` suboption to rename the Test column.

```
. dtable bpsystol age weight i.race i.hlthstat, svy
>   by(rural, nototals tests)
>   column(by(hide) test(p-value))
>   title(Survey data summary)
>   note(Mean (Standard deviation): p-value from linear regression.)
>   note(Frequency (Percent%): p-value from Pearson test.)
>   note(Statistics computed using the survey weights.)
>   note(Tests adjusted for the survey design.)
note: using test regress across levels of rural for bpsystol, age, and
weight.
note: using test pearson across levels of rural for race and hlthstat.
Survey data summary
```

	Urban	Rural	p-value
N	79,965,794 (68.3%)	37,191,719 (31.7%)	
Systolic blood pressure	126.607 (21.438)	127.675 (21.305)	0.406
Age (years)	41.805 (15.662)	43.215 (15.112)	0.024
Weight (kg)	71.322 (15.371)	73.144 (15.493)	<0.001
Race			
White	67,579,394 (84.5%)	35,420,155 (95.2%)	<0.001
Black	9,936,159 (12.4%)	1,253,077 (3.4%)	
Other	2,450,241 (3.1%)	518,487 (1.4%)	
Health status			
Excellent	22,781,784 (28.5%)	9,405,551 (25.3%)	<0.001
Very good	22,867,496 (28.6%)	9,308,814 (25.1%)	
Good	22,089,942 (27.7%)	10,625,453 (28.6%)	
Fair	8,892,926 (11.1%)	5,487,335 (14.8%)	
Poor	3,229,798 (4.0%)	2,308,158 (6.2%)	

Mean (Standard deviation): p-value from linear regression.

Frequency (Percent%): p-value from Pearson test.

Statistics computed using the survey weights.

Tests adjusted for the survey design.

## Save your style choices for next time

If you find yourself repeating certain options each time you use `dtable`, you can save most of those option choices for future tables. Use `collect style save` to store your `dtable` option choices to a collection style file and then specify this file in the `style()` option the next time you use `dtable`. If you store the collection style file in your `PERSONAL` folder, you can set that file as the default style for `dtable`.

In the following, we use `dtable` to create a table with some option choices, save our style to a file in the `PERSONAL` directory, and set that style file as the default for `dtable`.

```
. sysuse auto
(1978 automobile data)
. dtable mpg turn i.rep78,
>   sample(Sample, statistic(frequency) place(seplabels))
>   sformat("n=%s" frequency)
>   by(foreign, tests nototals missing)
>   continuous(, test(kwallis))
>   factor(, test(pearson))
>   title(Table 1. Summary by group with a test.)
>   note(Kruskal-Wallis test for continuous variables.)
>   note(Pearson's test for factor variables.)
note: using test kwallis across levels of foreign for mpg and turn.
note: using test pearson across levels of foreign for rep78.
Table 1. Summary by group with a test.
```

	Car origin		Test
	Domestic n=52	Foreign n=22	
Mileage (mpg)	19.827 (4.743)	24.773 (6.611)	0.002
Turn circle (ft.)	41.442 (3.968)	35.409 (1.501)	<0.001
Repair record 1978			
1	2 (4.2%)	0 (0.0%)	<0.001
2	8 (16.7%)	0 (0.0%)	
3	27 (56.2%)	3 (14.3%)	
4	9 (18.8%)	9 (42.9%)	
5	2 (4.2%)	9 (42.9%)	

```
Kruskal-Wallis test for continuous variables.
Pearson's test for factor variables.
```

```
. personal dir
your personal ado-directory is C:\ado\personal\
(output omitted)
. collect style save "C:\ado\personal\mydtable"
(style from DTable saved to file C:\ado\personal\mydtable.stjson)
. set dtable_style mydtable
. display c(dtable_style)
mydtable
```

Now these style choices are the defaults for tables created in the future. Here we create a similarly styled table but from different data.

```
. use https://www.stata-press.com/data/r18/lbw
(Hosmer & Lemeshow data)
. dtable age i.smoke i.ht, by(low)
note: using test kwallis across levels of low for age.
note: using test pearson across levels of low for smoke and ht.
Table 1. Summary by group with a test.
```

	Birthweight<2500g		Test
	0 n=130	1 n=59	
Age of mother	23.662 (5.585)	22.305 (4.511)	0.248
Smoked during pregnancy			
Nonsmoker	86 (66.2%)	29 (49.2%)	0.026
Smoker	44 (33.8%)	30 (50.8%)	
Has history of hypertension			
0	125 (96.2%)	52 (88.1%)	0.036
1	5 (3.8%)	7 (11.9%)	

Kruskal-Wallis test for continuous variables.

Pearson's test for factor variables.

To reset the dtable style back to its original default, type

```
. set dtable_style dtable
```

You can add the `permanently` option to set the default dtable style for future Stata sessions.

## Composite results

dtable's default style does not define any composite results, so you will need to define your own with the `define()` option. Stata has a collection style named `dtable_composites` that defines the following composite results for you.

<i>composite</i>	<i>elements</i>	Description
<code>idi</code>	<code>p10 p90</code>	Interdecile interval
<code>iqi</code>	<code>q1 q3</code>	Interquartile interval
<code>rangei</code>	<code>min max</code>	Range interval

All of these results are bound in square brackets without a delimiter between their elements. To get access to these composite results, use the `style(dtable_composites, override)` option. You can then specify `idi`, `iqi`, and `rangei` in the `statistic()` suboption of `continuous()`, just as you would any other statistic.

You may want different formatting for these composite results, or you may want to create composite results based on other statistics. The `define()` option allows you to customize your own composite results. In the following, we use `dtable` to create a table that defines our own interval results with some other option choices and then save the collection's style to a file.

```

. sysuse auto
(1978 automobile data)

. dtable, define(idi = p10 p90, delimiter("; "))
>   define(iqi = q1 q3, delimiter("; "))
>   define(rangei = min max, delimiter("; "))
>   sample(Sample, statistic(frequency) place(seplabels))
>   sformat("n=%s" frequency)
>   sformat("%s" idi)
>   sformat("[%s]" iqi)
>   sformat("<%s>" rangei)
>   continuous(mpg, statistic(p50 idi))
>   continuous(turn, statistic(q2 iqi))
>   continuous(trunk, statistic(mean rangei))
>   title(Table 1. Location and interval.)
>   note(Median {Interdecile interval})
>   note(Median [Interquartile interval])
>   note(Mean <Range interval>)

```

Table 1. Location and interval.

	Summary n=74
Mileage (mpg)	20.000 {14.000; 29.000}
Turn circle (ft.)	40.000 [36.000; 43.000]
Trunk space (cu. ft.)	13.757 <5.000; 23.000>

```

Median {Interdecile interval}
Median [Interquartile interval]
Mean <Range interval>

```

```

. collect style save dtable-intervals
(style from DTable saved to file dtable-intervals.stjson)

```

Now we can use this style and simply reference our composite results without having to redefine them.

```

. dtable, continuous(mpg, statistic(p50 idi))
>   continuous(turn, statistic(q2 iqi))
>   continuous(trunk, statistic(mean rangei))
>   style(dtable-intervals)

```

Table 1. Location and interval.

	Summary n=74
Mileage (mpg)	20.000 {14.000; 29.000}
Turn circle (ft.)	40.000 [36.000; 43.000]
Trunk space (cu. ft.)	13.757 <5.000; 23.000>

```

Median {Interdecile interval}
Median [Interquartile interval]
Mean <Range interval>

```



## The default style

Most of dtable's option defaults are controllable in its collection style. The following options are not part of dtable's collection style:

```
svy
subpop()
export()
name()
replace
label()
style()
warn
```

When the by() option is not specified, the default column() header options are

```
summary("Summary", nohide
        halign(center)
        font(, nobold
            noitalic
            nostrikeout
            nounderline)
        smcl(text))
```

When a variable is specified in the by() option, its default options are

```
notest
testnotes
totals
nomissing
```

and its default column() header options are

```
by(label, nohide
    fvlabel
    halign(center)
    font(, nobold
        noitalic
        nostrikeout
        nounderline)
    smcl(text))
total("Total", nohide
    halign(center)
    font(, nobold
        noitalic
        nostrikeout
        nounderline)
    smcl(text))
test("Test", nohide
    halign(center)
    font(, nobold
        noitalic
        nostrikeout
        nounderline)
    smcl(text))
```

The default sample frequency title is “N”. When weights are not specified, the default sample frequency statistic is `frequency`; otherwise, `sumw` is reported. When the `by()` option is specified, `percent` is added to the default sample frequency statistic. The other default `sample()` options are

```
font(, nobold
      noitalic
      nostrikeout
      nounderline)
smcl(text)
place(items)
```

The defaults for the `continuous()` option are

```
statistic(mean sd)
test(regress)
font(, nobold
      noitalic
      nostrikeout
      nounderline)
smcl(text)
```

The defaults for the `factor()` option are

```
statistic(fvfrequency fvpercent)
test(pearson)
font(, nobold
      noitalic
      nostrikeout
      nounderline)
smcl(text)
```

There are no predefined composite results in `dtable`'s default style.

The default formats are

```
nformat("%21.3fc", basestyle)
nformat("%21.0fc" N count sumw frequency fvfrequency fvrawfrequency)
nformat("%9.1fc" percent rawpercent fvpercent fvrawpercent)
sformat("%s%" percent rawpercent fvpercent fvrawpercent)
sformat("%s)" sd)
nformat("%6.3f" regress poisson lnnormal kwallis kwallis_ties pearson
        fisher lrchi2 svylr svywald svyllwald)
```

There is no default table title. The defaults for the `titlestyles()` option are

```
font(, nobold
      noitalic
      nostrikeout
      nounderline)
smcl(text)
```

There are no default table notes. The defaults for the `notestyles()` option are

```
font(, nobold
      noitalic
      nostrikeout
      nounderline)
smcl(text)
```

The remaining default options for `dtable` are

```
varlabel
fvlabel
nolistwise
halign(right)
```

## Methods and formulas

See *Methods and formulas* in [R] [table](#) for details on the statistics supported by `dtable`. When the `svy` or `subpop()` option is specified, the weights from `svyset` are used to compute statistics that support weights. In addition, the subpopulation variance  $\hat{\sigma}^2$  is computed as

$$\hat{\sigma}^2 = s^2 \frac{M}{n-1} \cdot \frac{n_s - 1}{M_s}$$

where  $s^2$  is the sample variance as documented in [R] [table](#),  $M$  is the estimated population size,  $n$  is the full sample size,  $M_s$  is the estimated subpopulation size, and  $n_s$  is the subpopulation sample size. The subpopulation standard deviation is computed as  $\hat{\sigma} = \sqrt{\hat{\sigma}^2}$  and coefficient of variation (cv) is computed as  $\hat{\sigma}/\bar{x}$ .

Let  $y$  be a continuous variable and `by(group, tests)` be in effect. For `test(regress)`, `dtable` reports the  $p$ -value from `test [#1]` after

```
regress y i.group [if] [in] [weight]
```

For `test(poisson)`, `dtable` reports the  $p$ -value from `test [#1]` after

```
poisson y i.group [if] [in] [weight]
```

For `test(lnormal)`, `dtable` reports the  $p$ -value from `test [#1]` after

```
gsem y <- i.group [if] [in] [weight], family(lognormal)
```

For `test(kwallis)` and `test(kwallis_ties)`, `dtable` reports the  $p$ -values corresponding with the  $\chi^2$  statistics reported by

```
kwallis y [if] [in], by(group)
```

When the `svy` or `subpop()` option is specified, the prefix

```
svy, subpop([varname] [if]) :
```

is used with `regress`, `poisson`, and `gsem`. For more details, see [R] [test](#), [R] [regress](#), [R] [poisson](#), [SEM] [gsem](#), [R] [kwallis](#), and [SVY] [svy](#).

Let `f` be a factor variable and `by(group, tests)` be in effect. For nonsurvey data, the factor-variable test across groups is performed using

```
tabulate f group [if] [in] [weight], opt
```

where *opt* is an option of `tabulate` corresponding with the specified test (*ftest*) using the following table:

<i>ftest</i>	<i>opt</i>
pearson	chi2
fisher	exact
lrchi2	lrchi2
gamma	gamma
kendall	taub
cramer	V

For survey data, `dtable` reports the *p*-value from

```
svy, subpop([varname] [if]): tabulate f group [if] [in], opt
```

where *opt* is an option of `svy: tabulate` corresponding with the specified test (*ftest*) using the following table:

<i>ftest</i>	<i>opt</i>
pearson	pearson
svylr	lr
svywald	wald
svyllwald	llwald

`pearson`, `fisher`, `lrchi2`, `svylr`, `svywald`, and `svyllwald` report a *p*-value. `gamma`, `kendall`, and `cramer` report the computed measure of association. For more details, see [R] [tabulate twoway](#) and [SVY] [svy: tabulate twoway](#).

# Appendix

## Colors

*bgcolor*, *fgcolor*, and *color*

---

aliceblue	darkslategray	lightsalmon	palevioletred
antiquewhite	darkturquoise	lightseagreen	papayawhip
aqua	darkviolet	lightskyblue	peachpuff
aquamarine	deeppink	lightslategray	peru
azure	deepskyblue	lightsteelblue	pink
beige	dimgray	lightyellow	plum
bisque	dodgerblue	lime	powderblue
black	firebrick	limegreen	purple
blanchedalmond	floralwhite	linen	red
blue	forestgreen	magenta	rosybrown
blueviolet	fuchsia	maroon	royalblue
brown	gainsboro	mediumaquamarine	saddlebrown
burlywood	ghostwhite	mediumblue	salmon
cadetblue	gold	mediumorchid	sandybrown
chartreuse	goldenrod	mediumpurple	seagreen
chocolate	gray	mediumseagreen	seashell
coral	green	mediumslateblue	sienna
cornflowerblue	greenyellow	mediumspringgreen	silver
cornsilk	honeydew	mediumturquoise	skyblue
crimson	hotpink	mediumvioletred	slateblue
cyan	indianred	midnightblue	slategray
darkblue	indigo	mintcream	snow
darkcyan	ivory	mistyrose	springgreen
darkgoldenrod	khaki	moccasin	steelblue
darkgray	lavender	navajowhite	tan
darkgreen	lavenderblush	navy	teal
darkkhaki	lawngreen	oldlace	thistle
darkmagenta	lemonchiffon	olive	tomato
darkolivegreen	lightblue	olivedrab	turquoise
darkorange	lightcoral	orange	violet
darkorchid	lightcyan	orangered	wheat
darkred	lightgoldenrodyellow	orchid	white
darksalmon	lightgray	palegoldenrod	whitesmoke
darkseagreen	lightgreen	palegreen	yellow
darkslateblue	lightpink	paleturquoise	yellowgreen

---

## Underline patterns

*upattern*


---

none	dashLong
single	dashLongHeavy
words	dotDash
double	dashDotHeavy
thick	dotDotDash
dotted	dashDotDotHeavy
dottedHeavy	wave
dash	wavyHeavy
dashedHeavy	wavyDouble

---

## Shading patterns

*fpattern*


---

nil	pct20
clear	pct25
solid	pct30
horzStripe	pct35
vertStripe	pct37
reverseDiagStripe	pct40
diagStripe	pct45
horzCross	pct50
diagCross	pct55
thinHorzStripe	pct60
thinVertStripe	pct62
thinReverseDiagStripe	pct65
thinDiagStripe	pct70
thinHorzCross	pct75
thinDiagCross	pct80
pct5	pct85
pct10	pct87
pct12	pct90
pct15	pct95

---

## Acknowledgments

We thank Phil Clayton of the ANZDATA Registry at the Royal Adelaide Hospital, Mark Chatfield of the University of Queensland, and Niels Henrik Bruun of Region Nordjylland for writing similar commands for creating tables of descriptive statistics. We also thank Joanna Dipnall of Monash University, Australia, for suggesting a unified framework for tables.

## Reference

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 intro](#) — Introduction to tables of frequencies, summaries, and command results

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

[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](#).