

**matlist** — Display a matrix and control its format

Description

Syntax

Style options

General options

Required options for the second syntax

Remarks and examples

Also see

## Description

`matlist` displays a matrix, allowing you to control the display format. Row and column names are used as the row and column headers. Equation names are displayed in a manner similar to estimation results.

Columns may have different formats, and lines may be shown between each column. You cannot format rows of the matrix differently.

`matlist` is an extension of the `matrix list` command (see [P] [matrix utility](#)).

## Syntax

*One common display format for every column*

```
matlist matrix_exp [ , style_options general_options ]
```

*Each column with its own display format*

```
matlist matrix_exp , cspec(cspec) rspec(rspec) [general_options]
```

*style\_options*

Description

---

<u>lines</u> ( <i>lstyle</i> )	lines style; default between headers/labels and data
<u>border</u> ( <i>bspec</i> )	border style; default is none
<u>border</u>	same as <code>border(all)</code>
<u>format</u> ( <i>%fmt</i> )	display format; default is <code>format(%9.0g)</code>
<u>twidth</u> (#)	row-label width; default is <code>twidth(12)</code>
<u>left</u> (#)	left indent for tables; default is <code>left(0)</code>
<u>right</u> (#)	right indent for tables; default is <code>right(0)</code>
<u>rightindent</u>	right indent data by one space relative to end of row lines

---

<i>lstyle</i>	Lines are drawn ...
<u>oneline</u>	between headers/labels and data; default with no equations
<u>eq</u>	between equations; default when equations are present
<u>rowtotal</u>	same as <u>oneline</u> plus line before last row
<u>coltotal</u>	same as <u>oneline</u> plus line before last column
<u>rctotal</u>	same as <u>oneline</u> plus line before last row and column
<u>rows</u>	between all rows; between row labels and data
<u>columns</u>	between all columns; between column header and data
<u>cells</u>	between all rows and columns
<u>none</u>	suppress all lines

---

<i>bspec</i>	Border lines are drawn ...
<u>none</u>	no border lines are drawn; the default
<u>all</u>	around all four sides
<u>rows</u>	at the top and bottom
<u>columns</u>	at the left and right
<u>left</u>	at the left
<u>right</u>	at the right
<u>top</u>	at the top
<u>bottom</u>	at the bottom

---

<i>general_options</i>	Description
<u>title</u> ( <i>string</i> )	title displayed above table
<u>tindent</u> (#)	indent title # spaces
<u>rowtitle</u> ( <i>string</i> )	title to display above row names
<u>rowtitleleft</u>	display row title flush left
<u>names</u> ( <u>rows</u> )	display row names
<u>names</u> ( <u>columns</u> )	display column names
<u>names</u> ( <u>all</u> )	display row and column names; the default
<u>names</u> ( <u>none</u> )	suppress row and column names
<u>nonames</u>	same as <u>names</u> ( <u>none</u> )
<u>showcoleq</u> ( <i>ceq</i> )	specify how column equation names are displayed
<u>roweqonly</u>	display only row equation names
<u>coleqonly</u>	display only column equation names
<u>colorcoleq</u> ( <u>txt</u>   <u>res</u> )	display mode (color) for column equation names; default is <u>txt</u>
<u>keepcoleq</u>	keep columns of the same equation together
<u>aligncolnames</u> ( <u>r</u> <u>align</u> )	right-align column names
<u>aligncolnames</u> ( <u>l</u> <u>align</u> )	left-align column names
<u>aligncolnames</u> ( <u>c</u> <u>enter</u> )	center column names
<u>noblank</u>	suppress blank line before tables
<u>nohalf</u>	display full matrix even if symmetric
<u>nodotz</u>	display missing value .z as blank
<u>underscore</u>	display underscores as blanks in row and column names
<u>linesize</u> (#)	override <u>linesize</u> setting

---

---

<i>ceq</i>	Equation names are displayed
<u>f</u> irst	over the first column only; the default
<u>e</u> ach	over each column
<u>c</u> ombined	centered over all associated columns
<u>l</u> combined	left-aligned over all associated columns
<u>r</u> combined	right-aligned over all associated columns

---

## Style options

`lines(lstyle)` specifies where lines are drawn in the display of *matrix\_exp*. The following values of *lstyle* are allowed:

`oneline` draws lines separating the row and column headers from the numerical entries. This is the default if the *matrix\_exp* has no equation names.

`eq` draws horizontal and vertical lines between equations. This is the default if the *matrix\_exp* has row or column equation names.

`rowtotal` is the same as `oneline` and has a line separating the last row (the totals) from the rest.

`coltotal` is the same as `oneline` and has a line separating the last column (the totals) from the rest.

`rctotal` is the same as `oneline` and has lines separating the last row and column (the totals) from the rest.

`rows` draws horizontal lines between all rows and one vertical line between the row-label column and the first column with numerical entries.

`columns` draws vertical lines between all columns and one horizontal line between the headers and the first numeric row.

`cells` draws horizontal and vertical lines between all rows and columns.

`none` suppresses all horizontal and vertical lines.

`border[bspec]` specifies the type of border drawn around the table. *bspec* is any combination of the following values:

`none` draws no outside border lines and is the default.

`all` draws all four outside border lines.

`rows` draws horizontal lines in the top and bottom margins.

`columns` draws vertical lines in the left and right margins.

`left` draws a line in the left margin.

`right` draws a line in the right margin.

`top` draws a line in the top margin.

`bottom` draws a line in the bottom margin.

`border` without an argument is equivalent to `border(all)`, or, equivalently, `border(left right top bottom)`.

`format(%fmt)` specifies the format for displaying the individual elements of the matrix. The default is `format(%9.0g)`. See [U] 12.5 **Formats: Controlling how data are displayed**.

`twidth(#)` specifies the width of the row-label column (first column). The default is `twidth(12)`.

`left(#)` specifies that the table be indented `#` spaces; the default is `left(0)`. To indent the title, see the `tindent()` option.

`right(#)` specifies that the right margin of the table be `#` spaces in from the page margin. The default is `right(0)`. The right margin affects the number of columns that are displayed before wrapping.

`rightindent` specifies that the data be indented by one space relative to the end of row lines. By default, the data are not indented except when there is a right border, in which case the data are indented by one space.

## General options

`title(string)` adds `string` as the title displayed before the matrix. `matlist` has no default title or header.

`tindent(#)` specifies the indentation for the title; the default is `tindent(0)`.

`rowtitle(string)` specifies that `string` be used as a column header for the row labels. This option is allowed only when both row and column labels are displayed.

`rowtitleleft` specifies that the row title be displayed flush left.

`names(rows | columns | all | none)` specifies whether the row and column names are displayed; the default is `names(all)`, which displays both.

`nonames` suppresses row and column names and is a synonym for `names(none)`.

`showcoleq(ceq)` specifies how column equation names are displayed. The following `ceq` are allowed:

`first` displays an equation name over the first column associated with that name; this is the default.

`each` displays an equation name over each column.

`combined` displays an equation name centered over all columns associated with that name.

`lcombined` displays an equation name left-aligned over all columns associated with that name.

`rcombined` displays an equation name right-aligned over all columns associated with that name.

If necessary, equation names are truncated to the width of the field in which the names are displayed. With `combined`, `lcombined`, and `rcombined`, the field comprises all columns and the associated separators for the equation.

`roweqonly` specifies that only row equation names be displayed in the output. This option may not be combined with `names(columns)`, `names(none)`, or `nonames`.

`coleqonly` specifies that only column equation names be displayed in the output. This option may not be combined with `names(rows)`, `names(none)`, or `nonames`.

`colorcoleq(txt | res)` specifies the mode (`color`) used for the column equation names that appear in the first displayed row. Specifying `txt` (the default) displays the equation name in the same color used to display text. Specifying `res` displays the name in the same color used to display results.

`keepcoleq` specifies that columns of the same equation be kept together if possible.

`aligncolnames(ralign | lalign | center)` specifies the alignment for the column names. `ralign` indicates alignment to the right, `lalign` indicates alignment to the left, and `center` indicates centering. `aligncolnames(ralign)` is the default.

`noblank` suppresses printing a blank line before the matrix. This is useful in programs.

`nohalf` specifies that, even if the matrix is symmetric, the full matrix be printed. The default is to print only the lower triangle in such cases.

`nodotz` specifies that `.z` missing values be listed as a field of blanks rather than as `.z`; see [U] 12.2.1 Missing values.

`underscore` converts underscores to blanks in row and column names.

`linesize(#)` specifies the width of the page for formatting the table. Specifying a value of `linesize()` wider than your screen width can produce truly ugly output on the screen, but that output can nevertheless be useful if you are logging output and later plan to print the log on a wide printer.

## Required options for the second syntax

`cspec(cspec)` specifies the formatting of the columns and the separators of the columns,

where `cspec` is `[sep [qual] %#s] sep nspec [nspec [...]]`

and where `sep` is `[o#] &| [o#]`

`qual` is

<i>qual</i>	Description
s	standard font
b	boldface font
i	italic font
t	text mode
e	error mode
c	command mode
L	left-aligned
R	right-aligned
C	centered
w#	field width #

`nspec` is `[qual] nfmt sep`

`nfmt` is `%#.#{f|g}`

The first (optional) part, `[sep [qual] %#s]`, of `cspec` specifies the formatting for the column containing row names. It is required if the row names are part of the display; see the `names()` option. The number of `nspecs` should equal the number of columns of `matname`.

In a separator specification, `sep`, `|` specifies that a vertical line be drawn. `&` specifies that no line be drawn. The number of spaces before and after the separator may be specified with `o#`; these default to one space, except that by default no spaces are included before the first column and after the last column.

Here are examples for a matrix with two columns (three columns when you count the column containing the row labels):

```
cspec(& %16s & %9.2f & %7.4f &)
```

specifies that the first column, containing row labels, be displayed using 16 characters; the second column, with format %9.2f; and the third column, with format %7.4f. No vertical lines are drawn. The number of spaces before and after the table is 0. Columns are separated with two spaces.

```
cspec(&o2 %16s o2&o2 %9.2f o2&o2 %7.4f o2&)
```

specifies more white space around the columns (two spaces everywhere, for a total of four spaces between columns).

```
cspec(|%16s|%9.2f|%7.4f|)
```

displays the columns in the same way as the first example but draws vertical lines before and after each column.

```
cspec(| b %16s | %9.2f & %7.4f |)
```

specifies that vertical lines be drawn before and after all columns, except between the two columns with numeric entries. The first column is displayed in the boldface font.

`rspec(rspec)` specifies where horizontal lines be drawn. *rspec* consists of a sequence of characters, optionally separated by white space. - (or synonym |) specifies that a line be drawn. & indicates that no line be drawn. When *matname* has *r* rows, *r* + 2 characters are required if column headers are displayed, and *r* + 1 characters are required otherwise. The first character specifies whether a line is to be drawn before the first row of the table; the second, whether a line is to be drawn between the first and second row, etc.; and the last character, whether a line is to be drawn after the last row of the table.

You cannot add blank lines before or after the horizontal lines.

For example, in a table with column headers and three numeric rows,

```
rspec(|&&|) or equivalently rspec(--&&-)
```

specifies that horizontal lines be drawn before the first and second rows and after the last row, but not elsewhere.

## Remarks and examples

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

Remarks are presented under the following headings:

*All columns with the same format*  
*Different formats for each column*  
*Other output options*

### All columns with the same format

The `matrix list` command displays Stata matrices but gives you little control over formatting; see [P] [matrix utility](#).

The `matlist` command, on the other hand, offers a wide array of options to give you more detailed control over the formatting of the output.

The output produced by `matlist` is a rectangular table of numbers with an optional row and column on top and to the left of the table. We distinguish two cases. In the first style, all numeric columns are to be displayed in the same format. In the second style, each column and each intercolumn divider is formatted individually.

## ▷ Example 1

We demonstrate with a simple  $3 \times 2$  matrix, A.

```
. matrix A = ( 1,2 \ 3,4 \ 5,6 )
. matrix list A
A[3,2]
   c1  c2
r1   1   2
r2   3   4
r3   5   6
```

Like `matrix list`, the `matlist` command displays one matrix but adopts a tabular display style.

```
. matlist A
```

	c1	c2
r1	1	2
r2	3	4
r3	5	6

Other border lines at the left, top, right, and bottom of the table may be specified with the `border()` option. For instance, `border(rows)` specifies a horizontal line at the top and bottom margins. `rowtitle()` specifies a row title. To make it easier to organize output with multiple matrices, you can use the `left()` option to left-indent the output.

```
. matlist A, border(rows) rowtitle(rows) left(4)
```

rows	c1	c2
r1	1	2
r2	3	4
r3	5	6

The `lines()` option specifies where internal lines are to be drawn. `lines(none)` suppresses all internal horizontal and vertical lines. `lines(cell)` displays lines between all rows and columns. `twidth()` specifies the width of the first column—the column containing the row names. By default, `matlist` shows row and column names obtained from the matrix resulting from `matrix_exp`. `names(rows)` specifies that the row names be shown, and the column names be suppressed. `names(none)` would suppress all row and column names. You may also display a title for the table, displayed in SMCL paragraph mode; see [P] [smcl](#). If the table is indented, the title will be shown with a hanging indent. The `tindent()` option allows you to indent the title as well. Finally, `matlist` allows a matrix expression—convenient for interactive use. Enclose the matrix expression in parentheses if the expression itself contains commas.

```
. matlist 2*A, border(all) lines(none) format(%6.1f) names(rows) twidth(8)
> left(4) title(Guess what, a title)
Guess what, a title
```

r1	2.0	4.0
r2	6.0	8.0
r3	10.0	12.0

`matlist` supports equations.

## ▷ Example 2

By default, `matlist` draws vertical and horizontal lines between equations.

```
. matrix E = ( 1 , 2 , 3 , 4 , 5 , 6 , 7 \
>             8 , 9 , 10 , 11 , 12 , 13 , 14 \
>             15 , 16 , 17 , 18 , 19 , 20 , 21 \
>             22 , 23 , 24 , 25 , 26 , 27 , 28 \
>             29 , 30 , 31 , 32 , 33 , 34 , 35 \
>             36 , 37 , 38 , 39 , 40 , 41 , 42 )
. matrix colnames E = A:a1 A:a2 B:b1 B:b2 C:c1 C:c2 C:c3
. matrix rownames E = D:d1 D:d2 E:e1 E:e2 F:f1 F:f2
. matlist E
```

		A		B		C	
		a1	a2	b1	b2	c1	
D	d1	1	2	3	4	5	
	d2	8	9	10	11	12	
E	e1	15	16	17	18	19	
	e2	22	23	24	25	26	
F	f1	29	30	31	32	33	
	f2	36	37	38	39	40	
		C					
		c2	c3				
D	d1	6	7				
	d2	13	14				
E	e1	20	21				
	e2	27	28				
F	f1	34	35				
	f2	41	42				

`matlist` wraps the columns, if necessary. The `keepcoleq` option keeps all columns of an equation together. By default, `matlist` shows the equation name left-aligned over the first column associated with the equation. Equation names are truncated, if necessary. We may also display equation names in the field created by combining the columns associated with the equation. In this wider field, truncation of equation names will be rare. The `showcoleq(combined)` option displays the equation names centered in this combined field. See the description of the `showcoleq()` option for other ways to format the column equation names. `border(right)` displays a vertical line to the right of the table. If the table is wrapped, a border line is shown to the right of each panel.



```
. matlist hadamard(E,E)', showcoleq(c) keepcoleq border(right) left(4)
```

		D		E	
		d1	d2	e1	e2
A	a1	1	64	225	484
	a2	4	81	256	529
B	b1	9	100	289	576
	b2	16	121	324	625
C	c1	25	144	361	676
	c2	36	169	400	729
	c3	49	196	441	784
		F			
		f1	f2		
A	a1	841	1296		
	a2	900	1369		
B	b1	961	1444		
	b2	1024	1521		
C	c1	1089	1600		
	c2	1156	1681		
	c3	1225	1764		

◀

## Different formats for each column

`matlist` allows you to format each column's display format (for example, `%8.2f` for the data columns), type style (for example, boldface font), and alignment. You may also specify whether a vertical line is to be drawn between the columns and the number of spaces before and after the line.

### ▶ Example 3

We illustrate the different formatting options with the example of a matrix of test results, one row per test, with the last row representing an overall test.

```
. matrix Htest = ( 12.30, 2, .00044642 \
>                 2.17, 1, .35332874 \
>                 8.81, 3, .04022625 \
>                 20.05, 6, .00106763 )
. matrix rnames Htest = trunk length weight overall
. matrix colnames Htest = chi2 df p
```

Again we can display the matrix `Htest` with `matrix list`,

```
. matrix list Htest
Htest[4,3]
      chi2      df      p
trunk  12.3      2  .00044642
length  2.17     1  .35332874
weight  8.81     3  .04022625
overall 20.05    6  .00106763
```

or with `matlist`,

```
. matlist Htest
      |      chi2      df      p
-----|-----
trunk |      12.3      2  .0004464
length|      2.17     1  .3533287
weight|      8.81     3  .0402262
overall|     20.05    6  .0010676
```

Neither of these displays of `Htest` is attractive because all columns are the same width and the numbers are formatted with the same display format. `matlist` can provide a better display of the matrix `Htest`.

```
. matlist Htest, rowtitle(Variables) title(Test results)
> cspec(o4& %12s | %8.0g & %5.0f & %8.4f o2&) rspec(&-&&--)
```

Test results

Variables	chi2	df	p
trunk	12.3	2	0.0004
length	2.17	1	0.3533
weight	8.81	3	0.0402
overall	20.05	6	0.0011

The `cspec()` and `rspec()` options may look somewhat intimidating at first, but they become clear if we examine their parts. The table for matrix `Htest` has four columns: one string column with the row names and three numeric columns with `chi2` statistics, degrees of freedom, and  $p$ -values. There are also five separators: one before the first column, three between the columns, and one after the last column. Thus the `cspec()` specification is made up of  $4 + 5 = 9$  elements that are explained in the next table.

Element	Purpose	Description
<code>o4&amp;</code>	before column 1	4 spaces/no vertical line
<code>%12s</code>	display format column 1	string display format <code>%12s</code>
<code> </code>	between columns 1 and 2	1 space/vertical line/1 space
<code>%8.0g</code>	display format column 2	numeric display format <code>%8.0g</code>
<code>&amp;</code>	between columns 2 and 3	1 space/no vertical line/1 space
<code>%5.0f</code>	display format column 3	numeric display format <code>%5.0f</code>
<code>&amp;</code>	between columns 3 and 4	1 space/no vertical line/1 space
<code>%8.4f</code>	display format column 4	numeric display format <code>%8.4f</code>
<code>o2&amp;</code>	after column 4	2 spaces/no vertical line

Vertical lines are drawn if the separator consists of a `|` character, whereas no vertical line is drawn with an `&` specification. By default, one space is displayed before and after the vertical line;

the exception is that, by default, no space is displayed before the first separator and after the last separator. More white space may be added by adding `o` specifications. For instance, `o3 | o2`, or more compactly `o3|o2`, specifies that three spaces be included before the vertical line and two spaces after the line.

The `rspec()` row formatting specification for a table with  $r$  rows (including the column headers) comprises a series of  $r + 1$  `-` and `&` characters, where

- `-` denotes that a horizontal line is to be drawn and
- `&` denotes that no horizontal line is to be drawn.

The table for matrix `Htest` has five rows: the column headers and four data rows. The specification `rspec(&-&&--)` is detailed in the next table.

Element	Purpose	Description
<code>&amp;</code>	before row 1	no line is drawn
<code>-</code>	between rows 1 and 2	a line is drawn
<code>&amp;</code>	between rows 2 and 3	no line is drawn
<code>&amp;</code>	between rows 3 and 4	no line is drawn
<code>-</code>	between rows 4 and 5	a line is drawn
<code>-</code>	after row 5	a line is drawn

Lines are drawn before and after the last row of the table for matrix `Htest` to emphasize that this row is an overall (total) test.

Further formatting is possible. For instance, we can specify that the second column (the first numeric column) be in the boldface font and text mode and that the last column be in italic and command mode. We simply insert appropriate qualifiers in the specification part for the respective columns.

```
. matlist Htest, rowt(Variables) title(Test results (again))
> cspec( o4&o2 %10s | b t %8.0g & %4.0f & i c %7.4f o2& )
> rspec( & - & & - & )
```

Test results (again)

Variables	chi2	df	p
trunk	<b>12.3</b>	2	<i>0.0004</i>
length	<b>2.17</b>	1	<i>0.3533</i>
weight	<b>8.81</b>	3	<i>0.0402</i>
overall	<b>20.05</b>	6	<i>0.0011</i>

In this manual, the boldface font is used for the `chi2` column and the italic font is used for the `p` column, but there is no difference due to the requested text mode and command mode. If we run this example interactively, both the font change and color change due to the requested mode can be seen depending on your Results window color scheme. Depending on your settings, the `chi2` column might display in the boldface font and the green color (text mode); the `df` column, in the default standard font and the yellow color (result mode); and the `p` column, in the italic font and the white color (command mode). Or it may look exactly as it does in this manual.

## Other output options

### ▷ Example 4

Finally, we illustrate two options for use with the extended missing value `.z` and with row and column names that contain underscores.

```
. matrix Z = ( .z, 1 \ .c, .z )
. matrix rnames Z = row_1 row_2
. matrix colnames Z = col1 col2
. matlist Z
```

	col1	col2
row_1	.z	1
row_2	.c	.z

The `nodotz` option displays `.z` as blanks. Underscores in row names are translated into spaces with the `underscore` option.

```
. matlist Z, nodotz underscore
```

	col1	col2
row 1		1
row 2	.c	

◀

## Also see

[P] **matrix** — Introduction to matrix commands

[P] **matrix utility** — List, rename, and drop matrices

[U] **14 Matrix expressions**

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

