

st_matrix() — Obtain and put Stata matrices

Syntax Diagnostics	Description Also see	Remarks and examples	Conformability
-----------------------	-------------------------	----------------------	----------------

Syntax

```

real matrix    st_matrix(string scalar name)
string matrix st_matrixrowstripe(string scalar name)
string matrix st_matrixcolstripe(string scalar name)

void           st_matrix(string scalar name, real matrix X)
void           st_matrix(string scalar name, real matrix X, string scalar hcat)
void           st_matrixrowstripe(string scalar name, string matrix s)
void           st_matrixcolstripe(string scalar name, string matrix s)

void           st_replacematrix(string scalar name, real matrix X)

string scalar st_matrix_hcat(name)

```

where

1. All functions allow *name* to be
 - a. global matrix name such as "mymatrix",
 - b. `r()` matrix such as "r(Z)", or
 - c. `e()` matrix such as "e(V)".
2. `st_matrix(name)` returns the contents of the specified Stata matrix. It returns $J(0,0,.)$ if the matrix does not exist.
3. `st_matrix(name, X)` sets or resets the contents of the specified Stata matrix. Row and column stripes are set to the default `r1`, `r2`, ..., and `c1`, `c2`, ...
4. `st_replacematrix(name, X)` is an alternative way to replace existing Stata matrices. The number of rows and columns of *X* must match the Stata matrix being replaced, and in return, the row and column stripes are not replaced.
5. `st_matrix(name, X)` deletes the specified Stata matrix if *value*== $J(0,0,.)$ (if *value* is 0×0).
6. Neither `st_matrix()` nor `st_replacematrix()` can be used to set, replace, or delete special Stata `e()` matrices `e(b)`, `e(V)`, or `e(Cns)`. Only Stata commands `ereturn post` and `ereturn repost` can be used to set these special matrices; see [P] [ereturn](#). Also see [M-5] [stata\(\)](#) for executing Stata commands from Mata.

7. `st_matrix(name, X, hcat)` sets or resets the specified Stata matrix and sets the hidden or historical status when setting a Stata `e()` or `r()` matrix. Allowed *hcat* values are "visible", "hidden", "historical", and a string scalar release number such as "10", "10.1", or any string release number matching "#[#][. [#]]". See [\[P\] return](#) for a description of hidden and historical stored results.
8. `st_matrix_hcat(name)` returns the *hcat* associated with a Stata `e()` or `r()` matrix.
9. `st_matrixrowstripe()` and `st_matrixcolstripe()` allow querying and resetting the row and column stripes of existing or previously created Stata matrices.

Description

`st_matrix(name)` returns the contents of Stata's matrix *name*, or it returns `J(0,0,.)` if the matrix does not exist.

`st_matrixrowstripe(name)` returns the row stripe associated with the matrix *name*, or it returns `J(0,2,"")` if the matrix does not exist.

`st_matrixcolstripe(name)` returns the column stripe associated with the matrix *name*, or it returns `J(0,2,"")` if the matrix does not exist.

`st_matrix(name, X)` sets or resets the contents of the Stata matrix *name* to be *X*. If the matrix did not previously exist, a new matrix is created. If the matrix did exist, the new contents replace the old. Either way, the row and column stripes are also reset to contain "r1", "r2", ..., and "c1", "c2",

`st_matrix(name, X)` deletes the Stata matrix *name* when *X* is 0×0 : `st_matrix(name, J(0,0,.))` deletes Stata matrix *name* or does nothing if *name* does not exist.

`st_matrixrowstripe(name, s)` and `st_matrixcolstripe(name, s)` change the contents to be *s* of the row and column stripe associated with the already existing Stata matrix *name*. In either case, *s* must be $n \times 2$, where *n* = the number of rows (columns) of the underlying matrix.

`st_matrixrowstripe(name, s)` and `st_matrixcolstripe(name, s)` reset the row and column stripe to be "r1", "r2", ..., and "c1", "c2", ..., when *s* is 0×2 (that is, `J(0,2,"")`).

`st_replacematrix(name, X)` resets the contents of the Stata matrix *name* to be *X*. The existing Stata matrix must have the same number of rows and columns as *X*. The row stripes and column stripes remain unchanged.

`st_matrix(name, X, hcat)` and `st_matrix_hcat(name)` are used to set and query the *hcat* corresponding to a Stata `e()` or `r()` matrix. They are also rarely used. See [\[R\] stored results](#) and [\[P\] return](#) for more information.

Remarks and examples

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

Remarks are presented under the following headings:

*Processing Stata's row and column stripes
Stata's matsize is irrelevant*

Also see [\[M-5\] st_global\(\)](#) and [\[M-5\] st_rclear\(\)](#).

Processing Stata's row and column stripes

Both row stripes and column stripes are presented in the same way: each row of *s* represents the *eq:op.name* associated with a row or column of the underlying matrix. The first column records *eq*, and the second column records *op.name*. For instance, given the following Stata matrix

		eq2:	eq2:
	L.	L.	L.
	turn	turn	turn
mpg	1	2	3
L.mpg	5	6	7
eq2:mpg	9	10	11
eq2:L.mpg	13	14	15

`st_matrixrowstripe(name)` returns the 4×2 string matrix

```
""      "mpg"
""      "L.mpg"
"eq2"   "mpg"
"eq2"   "L.mpg"
```

and `st_matrixcolstripe(name)` returns

```
""      "turn"
""      "L.turn"
"eq2"   "turn"
"eq2"   "L.turn"
```

Stata's matsize is irrelevant

Matrices in Stata are limited to `matsize` (see [R] [matsize](#)), a number between 10 and 11,000. Mata matrices have no such limits.

When getting a matrix, the `matsize` limit plays no role.

When putting a matrix, the `matsize` limit is ignored; meaning that, to use the matrix in Stata, the user may have to reset `matsize` or, if the matrix is too large, the user may not be able to use the matrix at all.

Conformability

`st_matrix(name)`:

```
name:      1 × 1
result:    m × n (0 × 0 if not found)
```

`st_matrixrowstripe(name)`:

```
name:      1 × 1
result:    m × 2 (0 × 2 if not found)
```

`st_matrixcolstripe(name)`:

```
name:      1 × 1
result:    n × 2 (0 × 2 if not found)
```

`st_matrix(name, X):`

name: 1 × 1
X: r × c (0 × 0 means delete)
result: void

`st_matrix(name, X, hcat):`

name: 1 × 1
X: r × c
hcat: 1 × 1
result: void

`st_matrixrowstripe(name, s):`

name: 1 × 1
s: r × 2 (0 × 2 means default "r1", "r2", ...)
result: void

`st_matrixcolstripe(name, s):`

name: 1 × 1
s: c × 2 (0 × 2 means default "c1", "c2", ...)
result: void

`st_replacematrix(name, X):`

name: 1 × 1
X: m × n (0 × 0 means delete)
result: void

`st_matrix_hcat(name):`

name: 1 × 1
result: 1 × 1

Diagnostics

`st_matrix(name)`, `st_matrixrowstripe(name)`, and `st_matrixcolstripe(name)` abort with error if *name* is malformed. Also,

1. `st_matrix(name)` returns J(0,0,.) if Stata matrix *name* does not exist.
2. `st_matrixrowstripe(name)` and `st_matrixcolstripe(name)` return J(0,2,"") if Stata matrix *name* does not exist. There is no possibility that matrix *name* might exist and not have row and column stripes.

`st_matrix(name, X)`, `st_matrixrowstripe(name, s)`, and `st_matrixcolstripe(name, s)` abort with error if *name* is malformed. Also,

1. `st_matrixrowstripe(name, s)` aborts with error if `rows(s)` is not equal to the number of rows of Stata matrix *name* and `rows(s) != 0`, or if `cols(s) != 2`.
2. `st_matrixcolstripe(name, s)` aborts with error if `cols(s)` is not equal to the number of columns of Stata matrix *name* and `cols(s) != 0`, or if `cols(s) != 2`.

`st_replacematrix(name, X)` aborts with error if Stata matrix *name* does not have the same number of rows and columns as *X*. `st_replacematrix()` also aborts with error if Stata matrix *name* does not exist and `X != J(0,0,.)`; `st_replacematrix()` does nothing if the matrix does not exist and `X == J(0,0,.)`. `st_replacematrix()` aborts with error if *name* is malformed.

`st_matrix(name, X, hcat)` aborts with error if *hcat* is not an allowed value.

`st_matrix_hcat(name)` returns "visible" when *name* is not a Stata `e()` or `r()` matrix and returns "" when *name* is an `e()` or `r()` value that does not exist.

Also see

[M-5] [st_rclear\(\)](#) — Clear `r()`, `e()`, or `s()`

[M-4] [stata](#) — Stata interface functions