lowertriangle( ) — Extract lower or upper triangle

Description Syntax Remarks and examples Conformability Diagnostics Also see

Description

`lowertriangle()` returns the lower triangle of A.

`uppertriangle()` returns the upper triangle of A.

`_lowertriangle()` replaces A with its lower triangle.

`_uppertriangle()` replaces A with its upper triangle.

Syntax

numeric matrix `lowertriangle(`*numeric matrix A* [`,` *numeric scalar d*]`)`
numeric matrix `uppertriangle(`*numeric matrix A* [`,` *numeric scalar d*]`)`

void `_lowertriangle(`*numeric matrix A* [`,` *numeric scalar d*]`)`
void `_uppertriangle(`*numeric matrix A* [`,` *numeric scalar d*]`)`

where argument *d* is optional.

Remarks and examples

Remarks are presented under the following headings:

Optional argument d

Optional argument *d* specifies the treatment of the diagonal. Specifying *d*`>=.`, or not specifying *d* at all, means no special treatment; if

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

then

$$\texttt{lowertriangle(}A\texttt{, .)} = \begin{bmatrix} 1 & 0 & 0 \\ 4 & 5 & 0 \\ 7 & 8 & 9 \end{bmatrix}$$

If a nonmissing value is specified for d, however, that value is substituted for each element of the diagonal, for example,

$$\texttt{lowertriangle(}A\texttt{, 1)} = \begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 7 & 8 & 1 \end{bmatrix}$$

Nonsquare matrices

`lowertriangle()` and `uppertriangle()` may be used with nonsquare matrices. If

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix}$$

then

$$\texttt{lowertriangle(}A\texttt{)} = \begin{bmatrix} 1 & 0 & 0 \\ 5 & 6 & 0 \\ 9 & 10 & 11 \end{bmatrix}$$

and

$$\texttt{uppertriangle(}A\texttt{)} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & 6 & 7 & 8 \\ 0 & 0 & 11 & 12 \end{bmatrix}$$

`_lowertriangle()` and `_uppertriangle()`, however, may not be used with nonsquare matrices.

Conformability

`lowertriangle(`A`, `d`)`:

A:	$r \times c$	
d:	1×1	(optional)
result:	$r \times \min(r, c)$	

`uppertriangle(`A`, `d`)`:

A:	$r \times c$	
d:	1×1	(optional)
result:	$\min(r, c) \times c$	

`_lowertriangle(`A`, `d`)`, `_uppertriangle(`A`, `d`)`:

input:

A:	$n \times n$	
d:	1×1	(optional)

output:

A:	$n \times n$	

Diagnostics

None.

Also see

[M-4] **Manipulation** — Matrix manipulation