

diag() — Create diagonal matrix

[Description](#) [Syntax](#) [Remarks and examples](#) [Conformability](#)
[Diagnostics](#) [Also see](#)

Description

`diag()` creates diagonal matrices.

`diag(Z)`, Z a matrix, extracts the principal diagonal of Z to create a new matrix. Z must be square.

`diag(z)`, z a vector, creates a new matrix with the elements of z on its diagonal.

Syntax

numeric matrix `diag(numeric matrix Z)`

numeric matrix `diag(numeric vector z)`

Remarks and examples

Do not confuse `diag()` with its functional inverse, `diagonal()`; see [M-5] [diagonal\(\)](#). `diag()` creates a matrix from a vector (or matrix); `diagonal()` extracts the diagonal of a matrix into a vector.

Use of `diag()` should be avoided because it wastes memory. The [colon operators](#) will allow you to use vectors directly:

| Desired calculation | Equivalent |
|----------------------------|----------------------------|
| <code>diag(v)*X</code> , | |
| v is a column | $v:*X$ |
| v is a row | $v':*X$ |
| v is a matrix | <code>diagonal(v)*X</code> |
| <code>X*diag(v)</code> | |
| v is a column | $X:*v'$ |
| v is a row | $X:*v$ |
| v is a matrix | $X:*diagonal(v)'$ |

In the above table, it is assumed that v is real. If v might be complex, the transpose operators that appear must be changed to `transposeonly()` calls, because we do not want the conjugate. For instance, $v':*X$ would become `transposeonly(v):*X`.

Conformability

`diag(Z)`:

Z: $m \times n$
result: $\min(m, n) \times \min(m, n)$

`diag(z)`:

z: $1 \times n$ or $n \times 1$
result: $n \times n$

Diagnostics

None.

Also see

[M-5] [_diag\(\)](#) — Replace diagonal of a matrix

[M-5] [diagonal\(\)](#) — Extract diagonal into column vector

[M-5] [isdiagonal\(\)](#) — Whether matrix is diagonal

[M-4] [Manipulation](#) — Matrix manipulation