<div style="border: 1px solid black; padding: 10px;">

all( ) — Element comparisons

</div>

Description

$\mathtt{all}(L)$ is equivalent to $\mathtt{sum(!}L\mathtt{)==0}$ but is significantly faster.

$\mathtt{any}(L)$ is equivalent to $\mathtt{sum}(L)\mathtt{!=0}$ but is slightly faster.

$\mathtt{allof}(P, s)$ returns 1 if every element of P equals s and returns 0 otherwise. $\mathtt{allof}(P, s)$ is faster and consumes less memory than the equivalent construction $\mathtt{all}(P\mathtt{:==}s)$.

$\mathtt{anyof}(P, s)$ returns 1 if any element of P equals s and returns 0 otherwise. $\mathtt{anyof}(P, s)$ is faster and consumes less memory than the equivalent $\mathtt{any}(P\mathtt{:==}s)$.

Syntax

>*real scalar* $\mathtt{all}$(*real matrix L*)

>*real scalar* $\mathtt{any}$(*real matrix L*)

>*real scalar* $\mathtt{allof}$(*transmorphic matrix P*, *transmorphic scalar s*)

>*real scalar* $\mathtt{anyof}$(*transmorphic matrix P*, *transmorphic scalar s*)

Remarks and examples

These functions are fast, so their use is encouraged over alternative constructions.

$\mathtt{all()}$ and $\mathtt{any()}$ are typically used with logical expressions to detect special cases, such as

```
if (any(x :< 0)) {
        ...
}
```

or

```
if (all(x :>= 0)) {
        ...
}
```

$\mathtt{allof()}$ and $\mathtt{anyof()}$ are used to look for special values:

```
if (allof(x, 0)) {
        ...
}
```

or

```
    if (anyof(x, 0)) {
        ...
    }
```

Do not use `allof()` and `anyof()` to check for missing values—for example, `anyof(x, .)`—because to really check, you would have to check not only `.` but also `.a`, `.b`, ..., `.z`. Instead use `missing()`; see [M-5] **missing( )**.

Conformability

`all(L), any(L)`:

L:	$r \times c$
result:	1×1

`allof(P, s), anyof(P, s)`:

P:	$r \times c$
s:	1×1
result:	1×1

Diagnostics

`all(L)` and `any(L)` treat missing values in *L* as true.

`all(L)` and `any(L)` return 0 (false) if *L* is $r \times 0$, $0 \times c$, or 0×0.

`allof(P, s)` and `anyof(P, s)` return 0 (false) if *P* is $r \times 0$, $0 \times c$, or 0×0.

Also see

[M-4] **Utility** — Matrix utility functions