**Syntax**

```
numeric scalar  det(numeric matrix A)
numeric scalar  dettriangular(numeric matrix A)
```

**Description**

`det(A)` returns the determinant of `A`.

`dettriangular(A)` returns the determinant of `A`, treating `A` as if it were triangular (even if it is not).

**Remarks and examples**

Calculation of the determinant is made by obtaining the LU decomposition of `A` and then calculating the determinant of `U`:

\[
det(A) = det(PLU) = det(P) \times det(L) \times det(U) = \pm1 \times 1 \times det(U) = \pm det(U)
\]

Since `U` is (upper) triangular, `det(U)` is simply the product of its diagonal elements. See [M-5] `lud()`.

**Conformability**

`det(A), dettriangular(A):`

- `A`: `n` × `n`
- `result`: `1` × `1`

**Diagnostics**

- `det(A)` and `dettriangular(A)` return 1 if `A` is `0` × `0`.
- `det(A)` aborts with error if `A` is not square and returns missing if `A` contains missing values.
- `dettriangular(A)` aborts with error if `A` is not square and returns missing if any element on the diagonal of `A` is missing.
Both \( \det(A) \) and \( \text{dettriangular}(A) \) will return missing value if the determinant exceeds 8.99e+307.

Also see

[M-5] \( \text{lud()} \) — LU decomposition

[M-4] \( \text{matrix} \) — Matrix functions