

rank() — Rank of matrix[Description
Diagnostics](#)[Syntax
Also see](#)[Remarks and examples](#)[Conformability](#)

Description

`rank(A)` and `rank(A, tol)` return the rank of $A: m \times n$.

Syntax

real scalar `rank(numeric matrix A)`

real scalar `rank(numeric matrix A, real scalar tol)`

Remarks and examples

[stata.com](#)

The row rank of a matrix $A: m \times n$ is the number of rows of A that are linearly independent. The column rank is the number of columns that are linearly independent. The terms row rank and column rank, however, are used merely for emphasis. The ranks are equal, and the result is simply called the rank of A .

`rank()` calculates the rank by counting the number of nonzero singular values of the SVD of A , where nonzero is interpreted relative to a tolerance. `rank()` uses the same tolerance as `pinv()` (see [M-5] [pinv\(\)](#)) and as `svsolve()` (see [M-5] [svsolve\(\)](#)), and optional argument `tol` is specified in the same way as with those functions.

Thus if you were going to use `rank()` before calculating an inverse using `pinv()`, it would be better to skip `rank()` altogether and proceed to the `pinv()` step, because `pinv()` will return the rank, calculated as a by-product of calculating the inverse. Using `rank()` ahead of time, the SVD would be calculated twice.

`rank()` in general duplicates calculations; and, worse, if you are not planning on using `pinv()` or `svsolve()` but rather are planning on using some other function, the rank returned by `rank()` may disagree with the implied rank of whatever numerical method you subsequently use because each numerical method has its own precision and tolerances.

All that said, `rank()` is useful in interactive and pedagogical situations.

Conformability

`rank(A, tol):`

<i>A:</i>	$m \times n$	
<i>tol:</i>	1×1	(optional)
<i>result:</i>	1×1	

Diagnostics

`rank(A)` returns missing if A contains missing values.

Also see

[M-5] **svd()** — Singular value decomposition

[M-5] **fullsvd()** — Full singular value decomposition

[M-5] **pinv()** — Moore–Penrose pseudoinverse

[M-4] **matrix** — Matrix functions