

## Description

`matpowersym(A, p)` returns  $A^p$  for symmetric matrix or Hermitian matrix  $A$ . The matrix returned is real if  $A$  is real and complex if  $A$  is complex.

`_matpowersym(A, p)` does the same thing, but instead of returning the result, it stores the result in  $A$ .

## Syntax

*numeric matrix*    `matpowersym(numeric matrix A, real scalar p)`

*void*                `_matpowersym(numeric matrix A, real scalar p)`

## Remarks and examples

Do not confuse `matpowersym(A, p)` and  $A:\wedge p$ . If  $p==2$ , the first returns  $A*A$  and the second returns  $A$  with each element squared.

Powers can be positive, negative, integer, or noninteger. Thus `matpowersym(A, .5)` is a way to find the square-root matrix  $R$  such that  $R*R==A$ , and `matpowersym(A, -1)` is a way to find the inverse. For inversion, you could obtain the result more quickly using other routines.

Powers are obtained by extracting the eigenvectors and eigenvalues of  $A$ , raising the eigenvalues to the specified power, and then rebuilding the matrix. That is, first  $X$  and  $L$  are found such that

$$AX = X \times \text{diag}(L) \tag{1}$$

For symmetric (Hermitian) matrix  $A$ ,  $X$  is orthogonal, meaning  $X'X = XX' = I$ . Thus

$$A = X \times \text{diag}(L)X' \tag{2}$$

$A^p$  is then defined

$$A^p = X \times \text{diag}(L:\wedge p) \times X' \tag{3}$$

(1) is obtained via `symeigensystem()`; see [M-5] [eigensystem\(\)](#).

## Conformability

`matpowersym(A, p)`:

*A*:  $n \times n$   
*p*:  $1 \times 1$   
*result*:  $n \times n$

`_matpowersym(A, p)`:

*input*:

*A*:  $n \times n$   
*p*:  $1 \times 1$

*output*:

*A*:  $n \times n$

## Diagnostics

`matpowersym(A, p)` and `_matpowersym(A, p)` return missing results if *A* contains missing values.

Also:

1. These functions do not check that *A* is symmetric or Hermitian. If *A* is a real matrix, only the lower triangle, including the diagonal, is used. If *A* is a complex matrix, only the lower triangle and the real parts of the diagonal elements are used.
2. These functions return a matrix of the same storage type as *A*. That means that if *A* is real and  $A^p$  cannot be expressed as a real, a matrix of missing values is returned. If you want the generalized solution, code `matpowersym(C(A), p)`. This is the same rule as with scalars:  $(-1)^.5$  is missing, but  $C(-1)^.5$  is  $1i$ .
3. These functions are guaranteed to return a matrix that is numerically symmetric, Hermitian, or **symmetriconly** if theory states that the matrix should be symmetric, Hermitian, or symmetriconly.

Concerning theory, the returned result is not necessarily symmetric (Hermitian). The eigenvalues *L* of a symmetric (Hermitian) matrix are real. If  $L : ^p$  are real, then the returned matrix will be symmetric (Hermitian), but otherwise, it will not. Think of a negative eigenvalue and  $p = .5$ : this results in a complex eigenvalue for  $A^p$ . Then if the original matrix was real (the eigenvectors were real), the resulting matrix will be symmetriconly. If the original matrix was complex (the eigenvectors were complex), the resulting matrix will have no special structure.

## Also see

[M-5] `eigensystem()` — Eigenvectors and eigenvalues

[M-5] `matexpsym()` — Exponentiation and logarithms of symmetric matrices

[M-4] **Matrix** — Matrix functions

Stata, Stata Press, Mata, NetCourse, and NetCourseNow are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow is a trademark of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2025 StataCorp LLC, College Station, TX, USA. All rights reserved.



For suggested citations, see the FAQ on [citing Stata documentation](#).