Title stata.com

Matrix functions

Contents Functions Reference Also see

Contents

cholesky(M)	the Cholesky decomposition of the matrix: if $R =$ cholesky(S), then $RR^T = S$
colnumb(M,s)	the column number of M associated with column name s; *missing* if the column cannot be found
colsof(M)	the number of columns of M
corr(M)	the correlation matrix of the variance matrix
det(M)	the determinant of matrix M
diag(v)	the square, diagonal matrix created from the row or column vector
diag0cnt(M)	the number of zeros on the diagonal of M
el(s,i,j)	s[floor(i),floor(j)], the i, j element of the matrix named s; *missing* if i or j are out of range or if matrix s does not exist
get(*systemname*)	a copy of Stata internal system matrix *systemname*
hadamard(M,N)	a matrix whose i, j element is $M[i,j] \cdot N[i,j]$ (if M and N are not the same size, this function reports a conformability error)
I(n)	an $n \times n$ identity matrix if n is an integer; otherwise, a round(n) $\times$ round(n) identity matrix
inv(M)	the inverse of the matrix M
invsym(M)	the inverse of M if M is positive definite
issymmetric(M)	1 if the matrix is symmetric; otherwise, 0
J(r,c,z)	the $r \times c$ matrix containing elements z
matmissing(M)	1 if any elements of the matrix are missing; otherwise, 0
matuniform(r,c)	the $r \times c$ matrices containing uniformly distributed pseudorandom numbers on the interval $(0, 1)$
mreldif(X,Y)	the relative difference of X and Y, where the relative difference is defined as $\max_{i,j}\left\{\lvert x_{ij} - y_{ij}\rvert/(\lvert y_{ij}\rvert + 1)\right\}$
nullmat(*matname*)	use with the row-join (,) and column-join (\) operators in programming situations
rownumb(M,s)	the row number of M associated with row name s; *missing* if the row cannot be found
rowsof(M)	the number of rows of M
sweep(M,i)	matrix M with ith row/column swept
trace(M)	the trace of matrix M
vec(M)	a column vector formed by listing the elements of M, starting with the first column and proceeding column by column
vecdiag(M)	the row vector containing the diagonal of matrix M

Functions

We divide the basic matrix functions into two groups, according to whether they return a matrix or a scalar:

Matrix functions returning a matrix
Matrix functions returning a scalar

Matrix functions returning a matrix

In addition to the functions listed below, see [P] **matrix svd** for singular value decomposition, [P] **matrix symeigen** for eigenvalues and eigenvectors of symmetric matrices, and [P] **matrix eigenvalues** for eigenvalues of nonsymmetric matrices.

`cholesky(`M`)`
Description: the Cholesky decomposition of the matrix: if $R =$ `cholesky(`S`)`, then $RR^T = S$

R^T indicates the transpose of R. Row and column names are obtained from M.
Domain: $n \times n$, positive-definite, symmetric matrices
Range: $n \times n$ lower-triangular matrices

`corr(`M`)`
Description: the correlation matrix of the variance matrix

Row and column names are obtained from M.
Domain: $n \times n$ symmetric variance matrices
Range: $n \times n$ symmetric correlation matrices

`diag(`v`)`
Description: the square, diagonal matrix created from the row or column vector

Row and column names are obtained from the column names of M if M is a row vector or from the row names of M if M is a column vector.
Domain: $1 \times n$ and $n \times 1$ vectors
Range: $n \times n$ diagonal matrices

`get(`*systemname*`)`
Description: a copy of Stata internal system matrix *systemname*

This function is included for backward compatibility with previous versions of Stata.
Domain: existing names of system matrices
Range: matrices

`hadamard(`M`,`N`)`
Description: a matrix whose i, j element is $M[i,j] \cdot N[i,j]$ (if M and N are not the same size, this function reports a conformability error)
Domain M: $m \times n$ matrices
Domain N: $m \times n$ matrices
Range: $m \times n$ matrices

`I(`n`)`
Description: an $n \times n$ identity matrix if n is an integer; otherwise, a `round(`n`)` $\times$ `round(`n`)` identity matrix
Domain: real scalars 1 to `matsize`
Range: identity matrices

`inv(`M`)`

Description: the inverse of the matrix M

If M is singular, this will result in an error.

The function `invsym()` should be used in preference to `inv()` because `invsym()` is more accurate. The row names of the result are obtained from the column names of M, and the column names of the result are obtained from the row names of M.

Domain: $n \times n$ nonsingular matrices

Range: $n \times n$ matrices

`invsym(`M`)`

Description: the inverse of M if M is positive definite

If M is not positive definite, rows will be inverted until the diagonal terms are zero or negative; the rows and columns corresponding to these terms will be set to 0, producing a g2 inverse. The row names of the result are obtained from the column names of M, and the column names of the result are obtained from the row names of M.

Domain: $n \times n$ symmetric matrices

Range: $n \times n$ symmetric matrices

`J(`r`,`c`,`z`)`

Description: the $r \times c$ matrix containing elements z

Domain r: integer scalars 1 to `matsize`

Domain c: integer scalars 1 to `matsize`

Domain z: scalars $-$8e+307 to 8e+307

Range: $r \times c$ matrices

`matuniform(`r`,`c`)`

Description: the $r \times c$ matrices containing uniformly distributed pseudorandom numbers on the interval $(0, 1)$

Domain r: integer scalars 1 to `matsize`

Domain c: integer scalars 1 to `matsize`

Range: $r \times c$ matrices

`nullmat(`*matname*`)`

Description: use with the row-join (`,`) and column-join (`\`) operators in programming situations

Consider the following code fragment, which is an attempt to create the vector $(1, 2, 3, 4)$:

```
forvalues i = 1/4 {
        mat v = (v, `i')
}
```

The above program will not work because, the first time through the loop, `v` will not yet exist, and thus forming `(v, `i')` makes no sense. `nullmat()` relaxes that restriction:

```
forvalues i = 1/4 {
        mat v = (nullmat(v), `i')
}
```

The `nullmat()` function informs Stata that if `v` does not exist, the function row-join is to be generalized. Joining nothing with `` `i' `` results in `` (`i') ``. Thus the first time through the loop, `v` $= (1)$ is formed. The second time through, `v` does exist, so `v` $= (1, 2)$ is formed, and so on.

`nullmat()` can be used only with the `,` and `\` operators.

Domain: matrix names, existing and nonexisting

Range: matrices including null if *matname* does not exist

`sweep(`M`,`i`)`

Description: matrix M with ith row/column swept

The row and column names of the resultant matrix are obtained from M, except that the nth row and column names are interchanged. If $B =$ `sweep(`A`,`k`)`, then

$$B_{kk} = \frac{1}{A_{kk}}$$

$$B_{ik} = -\frac{A_{ik}}{A_{kk}}, \qquad i \neq k$$

$$B_{kj} = \frac{A_{kj}}{A_{kk}}, \qquad j \neq k$$

$$B_{ij} = A_{ij} - \frac{A_{ik}A_{kj}}{A_{kk}}, \qquad i \neq k, j \neq k$$

Domain M: $n \times n$ matrices

Domain i: integer scalars 1 to n

Range: $n \times n$ matrices

`vec(`M`)`

Description: a column vector formed by listing the elements of M, starting with the first column and proceeding column by column

Domain: matrices

Range: column vectors ($n \times 1$ matrices) `vecdiag(`M`)`

Description: the row vector containing the diagonal of matrix M

`vecdiag()` is the opposite of `diag()`. The row name is set to `r1`; the column names are obtained from the column names of M.

Domain: $n \times n$ matrices

Range: $1 \times n$ vectors

Matrix functions returning a scalar

`colnumb(`M`,`s`)`

Description: the column number of M associated with column name s; *missing* if the column cannot be found

Domain M: matrices

Domain s: strings

Range: integer scalars 1 to `matsize` or *missing*

`colsof(`M`)`

Description: the number of columns of M

Domain: matrices

Range: integer scalars 1 to `matsize`

`det(`M`)`

Description: the determinant of matrix M

Domain: $n \times n$ (square) matrices

Range: scalars −8e+307 to 8e+307

`diag0cnt(`M`)`

Description: the number of zeros on the diagonal of M

Domain: $n \times n$ (square) matrices

Range: integer scalars 0 to n

`el(`s`,`i`,`j`)`

Description: s`[floor(`i`),floor(`j`)]`, the i, j element of the matrix named s; *missing* if i or j are out of range or if matrix s does not exist

Domain s: strings containing matrix name

Domain i: scalars 1 to `matsize`

Domain j: scalars 1 to `matsize`

Range: scalars −8e+307 to 8e+307 or *missing*

`issymmetric(`M`)`

Description: `1` if the matrix is symmetric; otherwise, `0`

Domain M: matrices

Range: integers 0 and 1

`matmissing(`M`)`

Description: `1` if any elements of the matrix are missing; otherwise, `0`

Domain M: matrices

Range: integers 0 and 1

`mreldif(`X`,`Y`)`
Description: the relative difference of X and Y, where the relative difference is defined as $\max_{i,j}\{|x_{ij} - y_{ij}|/(|y_{ij}| + 1)\}$
Domain X: matrices
Domain Y: matrices with same number of rows and columns as X
Range: scalars $-$8e+307 to 8e+307

`rownumb(`M`,`s`)`
Description: the row number of M associated with row name s; *missing* if the row cannot be found
Domain M: matrices
Domain s: strings
Range: integer scalars 1 to `matsize` or *missing*

`rowsof(`M`)`
Description: the number of rows of M
Domain: matrices
Range: integer scalars 1 to `matsize`

`trace(`M`)`
Description: the trace of matrix M
Domain: $n \times n$ (square) matrices
Range: scalars $-$8e+307 to 8e+307

Jacques Salomon Hadamard (1865–1963) was born in Versailles, France. He studied at the Ecole Normale Supérieure in Paris and obtained a doctorate in 1892 for a thesis on functions defined by Taylor series. Hadamard taught at Bordeaux for 4 years and in a productive period published an outstanding theorem on prime numbers, proved independently by Charles de la Vallée Poussin, and worked on what are now called Hadamard matrices. In 1897, he returned to Paris, where he held a series of prominent posts. In his later career, his interests extended from pure mathematics toward mathematical physics. Hadamard produced papers and books in many different areas. He campaigned actively against anti-Semitism at the time of the Dreyfus affair. After the fall of France in 1940, he spent some time in the United States and then Great Britain.

Reference

Mazýa, V. G., and T. O. Shaposhnikova. 1998. *Jacques Hadamard, A Universal mathematician*. Providence, RI: American Mathematical Society.

Also see

[D] **egen** — Extensions to generate

[M-5] **intro** — Mata functions

[U] **13.3 Functions**

[U] **14.8 Matrix functions**