

## utility — Matrix utility functions

[Contents](#)[Description](#)[Remarks and examples](#)[Also see](#)**Contents**

[M-5] Manual entry	Function	Purpose
<b>Complex</b>		
<b>Re()</b>	Re() Im()	real part imaginary part
<b>C()</b>	C()	make complex
<b>Shape &amp; type</b>		
<b>rows()</b>	rows() cols() length()	number of rows number of columns number of elements of vector
<b>eltype()</b>	eltype() orgtype() classname() structname()	element type of object organizational type of object class name of a Mata class scalar struct name of a Mata struct scalar
<b>isreal()</b>	isreal() iscomplex() isstring() ispointer()	object is real matrix object is complex matrix object is string matrix object is pointer matrix
<b>isrealvalues()</b>	isrealvalues()	whether matrix contains only real values
<b>isview()</b>	isview()	whether matrix is view

**Properties**

<b>issymmetric()</b>	issymmetric() issymmetriconly()	whether matrix is symmetric (Hermitian) whether matrix is mechanically symmetric
<b>isdiagonal()</b>	isdiagonal()	whether matrix is diagonal
<b>diag0cnt()</b>	diag0cnt()	count 0s on diagonal

Selection
-----------

---

<b>select()</b>	<code>select()</code> <code>st_select()</code> <code>selectindex()</code>	select rows or columns select rows or columns of view select indices
-----------------	---	--

Missing values
----------------

---

<b>missing()</b>	<code>missing()</code> <code>rowmissing()</code> <code>colmissing()</code> <code>nonmissing()</code> <code>rownonmissing()</code> <code>colnonmissing()</code> <code>hasmissing()</code>	count of missing values count of missing values, by row count of missing values, by column count of nonmissing values count of nonmissing values, by row count of nonmissing values, by column whether matrix has missing values
<b>missingof()</b>	<code>missingof()</code>	appropriate missing value

Range, sums, & cross products
-------------------------------

---

<b>minmax()</b>	<code>rowmin()</code> <code>colmin()</code> <code>min()</code> <code>rowmax()</code> <code>colmax()</code> <code>max()</code> <code>rowminmax()</code> <code>colminmax()</code> <code>minmax()</code> <code>rowmaxabs()</code> <code>colmaxabs()</code>	minimum, by row minimum, by column minimum, overall maximum, by row maximum, by column maximum, overall minimum and maximum, by row minimum and maximum, by column minimum and maximum, overall <code>rowmax(abs())</code> <code>colmax(abs())</code>
<b>minindex()</b>	<code>minindex()</code> <code>maxindex()</code>	indices of minimums indices of maximums
<b>sum()</b>	<code>rowsum()</code> <code>colsum()</code> <code>sum()</code> <code>quadrowsum()</code> <code>quadcolsum()</code> <code>quadsum()</code>	sum of each row sum of each column overall sum quad-precision sum of each row quad-precision sum of each column quad-precision overall sum

---

 Range, sums, & cross products, *continued*


---

<b>runningsum()</b>	runningsum() quadrunningsum()	running sum of vector quad-precision runningsum()
<b>cross()</b>	cross()	$X'X$ , $X'Z$ , etc.
<b>crossdev()</b>	crossdev()	$(X: -x)'(X: -x)$ , $(X: -x)'(Z: -z)$ , etc.
<b>quadcross()</b>	quadcross() quadcrossdev()	quad-precision cross() quad-precision crossdev()

---

 Programming
 

---

<b>reldif()</b>	reldif() mreldif() mreldifsym() mreldifre()	relative difference max. relative difference between matrices max. relative difference from symmetry max. relative difference from real
<b>all()</b>	all() any() allof() anyof()	$\text{sum}(!L) == 0$ $\text{sum}(L) != 0$ $\text{all}(P == s)$ $\text{any}(P == s)$
<b>panelsetup()</b>	panelsetup() panelstats() panelsubmatrix() panelsubview()	initialize panel-data processing summary statistics on panels obtain matrix for panel $i$ obtain view matrix for panel $i$
<b>_negate()</b>	_negate()	fast negation of matrix

---

 Constants & tolerances
 

---

<b>mindouble()</b>	mindouble() maxdouble() smallestdouble()	minimum nonmissing value maximum nonmissing value smallest $e > 0$
<b>epsilon()</b>	epsilon()	unit roundoff error
<b>floatround()</b>	floatround()	round to float precision
<b>solve_tol()</b>	solve_tol()	tolerance used by solvers and inverters

---

## Description

Matrix utility functions tell you something about the matrix, such as the number of rows or whether it is diagonal.

## Remarks and examples

[stata.com](#)

There is a thin line between utility and manipulation; also see

[M-4] [manipulation](#)      Matrix manipulation functions

## Also see

[M-4] [intro](#) — Categorical guide to Mata functions