

Contents

[M-0] Introduction to the Mata manual

intro Introduction to the Mata manual

[M-1] Introduction and advice

intro Introduction and advice
ado Using Mata with ado-files
first Introduction and first session
help Obtaining help in Stata
how How Mata works
interactive Using Mata interactively
LAPACK The LAPACK linear-algebra routines
limits Limits and memory utilization
naming Advice on naming functions and variables
permutation An aside on permutation matrices and vectors
returnedargs Function arguments used to return results
source Viewing the source code
tolerance Use and specification of tolerances

[M-2] Language definition

intro Language definition
break Break out of for, while, or do loop
class Object-oriented programming (classes)
comments Comments
continue Continue with next iteration of for, while, or do loop
declarations Declarations and types
do do ... while (exp)
errors Error codes
exp Expressions
for for (exp1; exp2; exp3) stmt
ftof Passing functions to functions
goto goto label
if if (exp) ... else ...
op_arith Arithmetic operators
op_assignment Assignment operator
op_colon Colon operators
op_conditional Conditional operator
op_increment Increment and decrement operators
op_join Row- and column-join operators
op_kronecker Kronecker direct-product operator
op_logical Logical operators
op_range Range operators
op_transpose Conjugate transpose operator

| | |
|------------|----------------------------------|
| optargs | Optional arguments |
| pointers | Pointers |
| pragma | Suppressing warning messages |
| reswords | Reserved words |
| return | return and return(exp) |
| semicolons | Use of semicolons |
| struct | Structures |
| subscripts | Use of subscripts |
| syntax | Mata language grammar and syntax |
| version | Version control |
| void | Void matrices |
| while | while (exp) stmt |

[M-3] Commands for controlling Mata

| | |
|---------------|--|
| intro | Commands for controlling Mata |
| end | Exit Mata and return to Stata |
| mata | Mata invocation command |
| mata clear | Clear Mata's memory |
| mata describe | Describe contents of Mata's memory |
| mata drop | Drop matrix or function |
| mata help | Obtain help in Stata |
| mata matsave | Save and restore matrices |
| mata memory | Report on Mata's memory usage |
| mata mlib | Create function library |
| mata mosave | Save function's compiled code in object file |
| mata rename | Rename matrix or function |
| mata set | Set and display Mata system parameters |
| mata stata | Execute Stata command |
| mata which | Identify function |
| namelists | Specifying matrix and function names |

[M-4] Index and guide to functions

| | |
|--------------|---|
| intro | Index and guide to functions |
| io | I/O functions |
| manipulation | Matrix manipulation |
| mathematical | Important mathematical functions |
| matrix | Matrix functions |
| programming | Programming functions |
| scalar | Scalar mathematical functions |
| solvers | Functions to solve $AX=B$ and to obtain A inverse |
| standard | Functions to create standard matrices |
| stata | Stata interface functions |
| statistical | Statistical functions |
| string | String manipulation functions |
| utility | Matrix utility functions |

[M-5] Mata functions

| | |
|-------|----------------|
| intro | Mata functions |
|-------|----------------|

| | |
|------------------|---|
| abbrev() | Abbreviate strings |
| abs() | Absolute value (length) |
| adosubdir() | Determine ado-subdirectory for file |
| all() | Element comparisons |
| args() | Number of arguments |
| asarray() | Associative arrays |
| ascii() | Manipulate ASCII codes |
| assert() | Abort execution if false |
| blockdiag() | Block-diagonal matrix |
| bufio() | Buffered (binary) I/O |
| byteorder() | Byte order used by computer |
| C() | Make complex |
| c() | Access c() value |
| callersversion() | Obtain version number of caller |
| cat() | Load file into string matrix |
| chdir() | Manipulate directories |
| cholesky() | Cholesky square-root decomposition |
| cholinv() | Symmetric, positive-definite matrix inversion |
| cholsolve() | Solve $AX=B$ for X using Cholesky decomposition |
| comb() | Combinatorial function |
| cond() | Condition number |
| conj() | Complex conjugate |
| corr() | Make correlation matrix from variance matrix |
| cross() | Cross products |
| crossdev() | Deviation cross products |
| cvpermute() | Obtain all permutations |
| date() | Date and time manipulation |
| deriv() | Numerical derivatives |
| designmatrix() | Design matrices |
| det() | Determinant of matrix |
| _diag() | Replace diagonal of a matrix |
| diag() | Create diagonal matrix |
| diag0cnt() | Count zeros on diagonal |
| diagonal() | Extract diagonal into column vector |
| dir() | File list |
| direxists() | Whether directory exists |
| direxternal() | Obtain list of existing external globals |
| display() | Display text interpreting SMCL |
| displayas() | Set display level |
| displayflush() | Flush terminal-output buffer |
| Dmatrix() | Duplication matrix |
| _docx*() | Generate Office Open XML (.docx) file |
| dsign() | FORTRAN-like DSIGN() function |
| e() | Unit vectors |
| editmissing() | Edit matrix for missing values |
| edittoint() | Edit matrix for roundoff error (integers) |
| edittozero() | Edit matrix for roundoff error (zeros) |
| editvalue() | Edit (change) values in matrix |
| eigensystem() | Eigenvectors and eigenvalues |

| | |
|-----------------------------------|--|
| <code>eigensystemsselect()</code> | Compute selected eigenvectors and eigenvalues |
| <code>eltype()</code> | Element type and organizational type of object |
| <code>epsilon()</code> | Unit roundoff error (machine precision) |
| <code>_equilrc()</code> | Row and column equilibration |
| <code>error()</code> | Issue error message |
| <code>errprintf()</code> | Format output and display as error message |
| <code>exit()</code> | Terminate execution |
| <code>exp()</code> | Exponentiation and logarithms |
| <code>factorial()</code> | Factorial and gamma function |
| <code>favorspeed()</code> | Whether speed or space is to be favored |
| <code>ferrortext()</code> | Text and return code of file error code |
| <code>fft()</code> | Fourier transform |
| <code>fileexists()</code> | Whether file exists |
| <code>_fillmissing()</code> | Fill matrix with missing values |
| <code>findexternal()</code> | Find, create, and remove external globals |
| <code>findfile()</code> | Find file |
| <code>floatround()</code> | Round to float precision |
| <code>fmtwidth()</code> | Width of <code>%fmt</code> |
| <code>fopen()</code> | File I/O |
| <code>fullsvd()</code> | Full singular value decomposition |
| <code>geigensystem()</code> | Generalized eigenvectors and eigenvalues |
| <code>ghessenbergd()</code> | Generalized Hessenberg decomposition |
| <code>ghk()</code> | Geweke–Hajivassiliou–Keane (GHK) multivariate normal simulator |
| <code>ghkfast()</code> | GHK multivariate normal simulator using pregenerated points |
| <code>gschurd()</code> | Generalized Schur decomposition |
| <code>halton()</code> | Generate a Halton or Hammersley set |
| <code>hash1()</code> | Jenkins' one-at-a-time hash function |
| <code>hessenbergd()</code> | Hessenberg decomposition |
| <code>Hilbert()</code> | Hilbert matrices |
| <code>I()</code> | Identity matrix |
| <code>inbase()</code> | Base conversion |
| <code>indexnot()</code> | Find character not in list |
| <code>invorder()</code> | Permutation vector manipulation |
| <code>invsym()</code> | Symmetric real matrix inversion |
| <code>invtokens()</code> | Concatenate string rowvector into string scalar |
| <code>isdiagonal()</code> | Whether matrix is diagonal |
| <code>isfleeting()</code> | Whether argument is temporary |
| <code>isreal()</code> | Storage type of matrix |
| <code>isrealvalues()</code> | Whether matrix contains only real values |
| <code>issymmetric()</code> | Whether matrix is symmetric (Hermitian) |
| <code>isview()</code> | Whether matrix is view |
| <code>J()</code> | Matrix of constants |
| <code>Kmatrix()</code> | Commutation matrix |
| <code>lapack()</code> | LAPACK linear-algebra functions |
| <code>liststruct()</code> | List structure's contents |
| <code>Lmatrix()</code> | Elimination matrix |
| <code>logit()</code> | Log odds and complementary log-log |

| | |
|-----------------|---|
| lowertriangle() | Extract lower or upper triangle |
| lud() | LU decomposition |
| luinv() | Square matrix inversion |
| lusolve() | Solve $AX=B$ for X using LU decomposition |
| makesymmetric() | Make square matrix symmetric (Hermitian) |
| matexpsym() | Exponentiation and logarithms of symmetric matrices |
| matpowersym() | Powers of a symmetric matrix |
| mean() | Means, variances, and correlations |
| mindouble() | Minimum and maximum nonmissing value |
| minindex() | Indices of minimums and maximums |
| minmax() | Minimums and maximums |
| missing() | Count missing and nonmissing values |
| missingof() | Appropriate missing value |
| mod() | Modulus |
| moptimize() | Model optimization |
| more() | Create –more– condition |
| –negate() | Negate real matrix |
| norm() | Matrix and vector norms |
| normal() | Cumulatives, reverse cumulatives, and densities |
| optimize() | Function optimization |
| panelsetup() | Panel-data processing |
| pathjoin() | File path manipulation |
| pinv() | Moore–Penrose pseudoinverse |
| polyeval() | Manipulate and evaluate polynomials |
| printf() | Format output |
| qrd() | QR decomposition |
| qrinv() | Generalized inverse of matrix via QR decomposition |
| qrsolve() | Solve $AX=B$ for X using QR decomposition |
| quadcross() | Quad-precision cross products |
| range() | Vector over specified range |
| rank() | Rank of matrix |
| Re() | Extract real or imaginary part |
| reldif() | Relative/absolute difference |
| rows() | Number of rows and number of columns |
| rowshape() | Reshape matrix |
| runiform() | Uniform and nonuniform pseudorandom variates |
| runningsum() | Running sum of vector |
| schurd() | Schur decomposition |
| select() | Select rows, columns, or indices |
| setbreakintr() | Break-key processing |
| sign() | Sign and complex quadrant functions |
| sin() | Trigonometric and hyperbolic functions |
| sizeof() | Number of bytes consumed by object |
| solve_tol() | Tolerance used by solvers and inverters |
| sovelower() | Solve $AX=B$ for X , A triangular |
| solvenl() | Solve systems of nonlinear equations |
| sort() | Reorder rows of matrix |

| | |
|---------------------------------|---|
| <code>soundex()</code> | Convert string to soundex code |
| <code>spline3()</code> | Cubic spline interpolation |
| <code>sqrt()</code> | Square root |
| <code>st_addobs()</code> | Add observations to current Stata dataset |
| <code>st_addvar()</code> | Add variable to current Stata dataset |
| <code>st_data()</code> | Load copy of current Stata dataset |
| <code>st_dir()</code> | Obtain list of Stata objects |
| <code>st_dropvar()</code> | Drop variables or observations |
| <code>st_global()</code> | Obtain strings from and put strings into global macros |
| <code>st_isfmt()</code> | Whether valid <code>%fmt</code> |
| <code>st_isname()</code> | Whether valid Stata name |
| <code>st_local()</code> | Obtain strings from and put strings into Stata macros |
| <code>st_macroexpand()</code> | Expand Stata macros in string |
| <code>st_matrix()</code> | Obtain and put Stata matrices |
| <code>st_numscalar()</code> | Obtain values from and put values into Stata scalars |
| <code>st_nvar()</code> | Numbers of variables and observations |
| <code>st_rclear()</code> | Clear <code>r()</code> , <code>e()</code> , or <code>s()</code> |
| <code>st_store()</code> | Modify values stored in current Stata dataset |
| <code>st_subview()</code> | Make view from view |
| <code>st_tempname()</code> | Temporary Stata names |
| <code>st_tsrevar()</code> | Create time-series <code>op.varname</code> variables |
| <code>st_update()</code> | Determine or set data-have-changed flag |
| <code>st_varformat()</code> | Obtain/set format, etc., of Stata variable |
| <code>st_varindex()</code> | Obtain variable indices from variable names |
| <code>st_varname()</code> | Obtain variable names from variable indices |
| <code>st_varrename()</code> | Rename Stata variable |
| <code>st_vartype()</code> | Storage type of Stata variable |
| <code>st_view()</code> | Make matrix that is a view onto current Stata dataset |
| <code>st_viewvars()</code> | Variables and observations of view |
| <code>st_vlexists()</code> | Use and manipulate value labels |
| <code>stata()</code> | Execute Stata command |
| <code>stataversion()</code> | Version of Stata being used |
| <code>strdup()</code> | String duplication |
| <code>strlen()</code> | Length of string |
| <code>strmatch()</code> | Determine whether string matches pattern |
| <code>stofreal()</code> | Convert real to string |
| <code>strpos()</code> | Find substring in string |
| <code>strreverse()</code> | Reverse string |
| <code>strtoname()</code> | Convert a string to a Stata name |
| <code>strtoreal()</code> | Convert string to real |
| <code>strtrim()</code> | Remove blanks |
| <code>strupper()</code> | Convert string to uppercase (lowercase) |
| <code>subinstr()</code> | Substitute text |
| <code>sublowertriangle()</code> | Return a matrix with zeros above a diagonal |
| <code>_substr()</code> | Substitute into string |
| <code>substr()</code> | Extract substring |
| <code>sum()</code> | Sums |
| <code>svd()</code> | Singular value decomposition |
| <code>svsolve()</code> | Solve $AX=B$ for X using singular value decomposition |
| <code>swap()</code> | Interchange contents of variables |

Toeplitz() Toeplitz matrices
 tokenget() Advanced parsing
 tokens() Obtain tokens from string
 trace() Trace of square matrix
 _transpose() Transposition in place
 transposeonly() Transposition without conjugation
 trunc() Round to integer

 uniqrows() Obtain sorted, unique values
 unitcircle() Complex vector containing unit circle
 unlink() Erase file

 valofexternal() Obtain value of external global
 Vandermonde() Vandermonde matrices
 vec() Stack matrix columns

 xl() Excel file I/O class

[M-6] Mata glossary of common terms

Glossary

 Subject and author index