

# 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 help    | Introduction and first session               |
| How           | Obtaining help in Stata                      |
| How           | How Mata works                               |
| Interactive   | Using Mata interactively                     |
| LAPACK        | Linear algebra package (LAPACK) routines     |
| Limits        | Limits and memory utilization                |
| Naming        | Advice on naming functions and variables     |
| Permutation   | An aside on permutation matrices and vectors |
| Returned args | 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                |
| lmbuild       | Easily create function library               |
| 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] Categorical guide to Mata functions**

|              |   |
|--------------|---|
| Intro        | Categorical guide to Mata functions                 |
| Dates        | Date and time 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] Alphabetical index to Mata functions

Intro ..... Alphabetical index to 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  
 AssociativeArray() ..... Associative arrays (class)  
 ascii() ..... Manipulate ASCII and byte codes  
 uchar() ..... Convert code point to Unicode character  
 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

|                                  |  |
|----------------------------------|--|
| <code>displayflush()</code>      | Flush terminal-output buffer                                   |
| <code>Dmatrix()</code>           | Duplication matrix   |
| <code>_docx*()</code>            | Generate Office Open XML (.docx) file                          |
| <code>dsign()</code>             | FORTRAN-like DSIGN() function                                  |
| <code>e()</code>                 | Unit vectors   |
| <code>editmissing()</code>       | Edit matrix for missing values                                 |
| <code>edittoint()</code>         | Edit matrix for roundoff error (integers)                      |
| <code>edittozero()</code>        | Edit matrix for roundoff error (zeros)                         |
| <code>editvalue()</code>         | Edit (change) values in matrix                                 |
| <code>eigensystem()</code>       | Eigenvectors and eigenvalues                                   |
| <code>eigensystemselect()</code> | Compute selected eigenvectors and eigenvalues                  |
| <code>eltype()</code>            | Element type, organizational type, and type name 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 %fmt  |
| <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’s 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 byte 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>isascii()</code>           | Whether string scalar contains only ASCII codes                |
| <code>isdiagonal()</code>        | Whether matrix is diagonal                                     |
| <code>isfleeting()</code>        | Whether argument is temporary                                  |

|                 |   |
|-----------------|---|
| isreal()        | Storage type of matrix                                    |
| isrealvalues()  | Whether matrix contains only real values                  |
| issamefile()    | Whether two file paths are pointing to the same file      |
| issymmetric()   | Whether matrix is symmetric (Hermitian)                   |
| isview()        | Whether matrix is view                                    |
| J()             | Matrix of constants                                       |
| Kmatrix()       | Commutation matrix  |
| lapack()        | Linear algebra package (LAPACK) functions                 |
| ldl()           | Bunch–Kaufman decomposition                               |
| LinearProgram() | Linear programming  |
| liststruct()    | List structure’s contents                                 |
| Lmatrix()       | Elimination matrix  |
| logit()         | 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                                   |
| mvnnormal()     | Compute multivariate normal distributions and derivatives |
| _negate()       | Negate real matrix  |
| norm()          | Matrix and vector norms                                   |
| normal()        | Cumulatives, reverse cumulatives, and densities           |
| optimize()      | Function optimization                                     |
| panelsetup()    | Panel-data processing                                     |
| panelsum()      | Panel sums  |
| pathjoin()      | File path manipulation                                    |
| Pdf*()          | Create a PDF file   |
| 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                             |
| Quadrature()    | Numerical integration                                     |

|                               |   |
|-------------------------------|---|
| <code>range()</code>          | Vector over specified range                                     |
| <code>rank()</code>           | Rank of matrix  |
| <code>Re()</code>             | Extract real or imaginary part                                  |
| <code>reldif()</code>         | Relative/absolute difference                                    |
| <code>rows()</code>           | Number of rows and number of columns                            |
| <code>rowshape()</code>       | Reshape matrix  |
| <code>runiform()</code>       | Uniform and nonuniform pseudorandom variates                    |
| <code>runningsum()</code>     | Running sum of vector   |
| <code>schurd()</code>         | Schur decomposition   |
| <code>select()</code>         | Select rows, columns, or indices                                |
| <code>setbreakintr()</code>   | Break-key processing  |
| <code>sign()</code>           | Sign and complex quadrant functions                             |
| <code>sin()</code>            | Trigonometric and hyperbolic functions                          |
| <code>sizeof()</code>         | Number of bytes consumed by object                              |
| <code>solve_tol()</code>      | Tolerance used by solvers and inverters                         |
| <code>solve_lower()</code>    | Solve $AX=B$ for $X$ , $A$ triangular                           |
| <code>solvenl()</code>        | Solve systems of nonlinear equations                            |
| <code>sort()</code>           | 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_frame*()</code>      | Data frame manipulation   |
| <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_updata()</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   |

|                    |  |
|--------------------|--|
| stataversion()     | Version of Stata being used  |
| strdup()           | String duplication   |
| strlen()           | Length of string in bytes  |
| ustrlen()          | Length of Unicode string in Unicode characters                       |
| udstrlen()         | Length of Unicode string in display columns                          |
| strmatch()         | Determine whether string matches pattern                             |
| strofreal()        | Convert real to string   |
| strpos()           | Find substring in string   |
| ustrpos()          | Find substring in Unicode string                                     |
| strreverse()       | Reverse string   |
| ustrreverse()      | Reverse Unicode string   |
| strtoname()        | Convert a string to a Stata 13 compatible name                       |
| ustrtoname()       | Convert a Unicode string to a Stata name                             |
| strtoreal()        | Convert string to real   |
| strtrim()          | Remove blanks  |
| ustrtrim()         | Remove Unicode whitespace characters                                 |
| strupper()         | Convert ASCII letter to uppercase (lowercase)                        |
| ustrupper()        | Convert Unicode string to uppercase, lowercase, or titlecase         |
| subinstr()         | Substitute text  |
| usubinstr()        | Replace Unicode substring  |
| sublowertriangle() | Return a matrix with zeros above a diagonal                          |
| _substr()          | Substitute into string   |
| _usubstr()         | Substitute into Unicode string                                       |
| substr()           | Extract substring  |
| usubstr()          | Extract Unicode substring  |
| udsubstr()         | Extract Unicode substring based on display columns                   |
| sum()              | Sums   |
| svd()              | Singular value decomposition   |
| svsolve()          | Solve $AX=B$ for $X$ using singular value decomposition              |
| swap()             | 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   |
| urlencode()        | Convert URL into percent-encoded ASCII format                        |
| ustrcompare()      | Compare or sort Unicode strings                                      |
| ustrfix()          | Replace invalid UTF-8 sequences in Unicode string                    |
| ustrnormalize()    | Normalize Unicode string   |
| ustrsplit()        | Split string into parts based on a Unicode regular expression        |
| ustrtto()          | Convert a Unicode string to or from a string in a specified encoding |
| ustrunescape()     | Convert escaped hex sequences to Unicode strings                     |
| ustrword()         | Obtain Unicode word from Unicode string                              |
| 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 .....