Mx
Quick Reference

For availability, technical support, and comments (which are especially welcome) please use one of the following methods of contact, listed in order of preference:

email neale@ruby.vcu.edu *(internet)*
neale@vcuvax *(bitnet)*
fax 804 371 6577
phone 804 786 8590
snail mail Michael C. Neale
Department of Psychiatry
Box 710 MCV
Richmond VA 23298-0710, USA

refer to Mx *manual as:*

Neale, M.C. (1994). *Mx: Statistical Modeling.* Box 710 MCV, Richmond, VA 23298: Department of Psychiatry. 2nd edition.

---

1 Outline of Mx Scripts

1.1 Syntax Conventions

Mx reads three things: *keywords, parameters* and *numbers* A keyword must start on a new line. The first two letters of a keyword are sufficient to identify it, unless explicitely stated otherwise. Parameters pass information to commands, e.g. filenames or numbers. Comments can be put anywhere following !. Characters after column 120 and blank lines are ignored. The syntax described for commands follows these conventions:

- alternatives are represented by /
- optional parameters or keywords are enclosed by { and }
- items to be substituted according to the specific application are enclosed by < and >

1.2 Job Structure

Mx has been written for multiple groups, since genetically informative data generally comprise information on different types of relatives which form distinct groups. Several Mx programs can be run in one file. Minimum specifications per group are capitalized:

- TITLE
- DATA: indicate group type: data/calculation/constraint
- Read, select any observed data, labels
- MATRICES: define at least one matrix
- MODEL: define matrix formula: covariances/means/threshold/compute
- Specify numbers and parameters, starting values, equality constraints
- Request fit functions, statistical output and optimization options, multiple fit mode, save matrices and job specifications
- END

2 Reading Data

2.1 Title Line

The title line is purely for the user's reference, it is printed when Mx prints the parameter specifications and the parameter estimates for a group. The title is recognized by its location rather than a keyword at the start of a line.

2.2 Data command

Syntax:
```
Data {Calc/Const} {NGroups=n
     NInput!vars=n NObservations=n}
```

The data line specifies type of group: the default is data group, `Calc` forces calculation group; `Const` forces constraint group. The first group *must* specify number of groups `NG`. Other parameters are number of input variables `NI` and number of observations `NO`.

2.3 Covariance and Correlation Matrices

Syntax:
```
CMatrix/KMatrix/PMatrix {Full}
     {File=<filename>}
```

Several different commands are used to read summary statistics either from the input script or an external file, if `File=` appears. Covariance matrices (`CM`) and correlation matrices (`KM` or `PM` for matrices of polyserial or polychoric correlations) are by default symmetric. `Full` indicates that a full matrix is supplied.

2.4 Asymptotic Variances and Covariances

Syntax:
```
ACov/AVar/AInv
     {File=<filename>}
```

In order to use asymptotic or diagonally weighted least squares, it is necessary to read a weight matrix. Mx expects a PRELIS output type matrix of asymptotic covariances (`AC`) or asymptotic variances (`AV`). The inverse of the asymptotic matrix (`AI`) could be used also.

2.5 Raw and Variable Length Data

Syntax:
```
Rawdata/VLength
     {File=<filename>}
```

Two types of raw data can be read: a rectangular matrix of balanced raw data (`RA`) or a variable length file for unbalanced data (`VL`). A variable length file contains, per pedigree, the number of input variables k (on a separate line) and identification codes and observed data for k variables.

2.6 Contigency Tables

Syntax:
```
CTable <r> <c> {File=<filename>}
```

Frequency data are read in a contingency table (`CT`) which requires number of rows and columns on same line. Mx will automatically handle incomplete ascertainment (when flagged by a negative number for cells that are not ascertained).

2.7 Mean Vector

Syntax:
```
Means {File=<filename>}
```

The `Means` keyword reads a vector of means.

2.8 Labels and Select commands

Syntax:
```
Labels <label list>
Select <label list>/<numlist> /
Labels Row/Col <matrix name>
  <label list>
```

Labels may be read for the input variables, and either labels or variable numbers may be used to select variables. An / must end the select command. After supplying covariance or mean commands, labels may be provided for the rows or colums of defined matrices.

3 Building Models with Matrices

3.1 Matrices command

Syntax:
```
Matrices
<name> <type> <r> <c>
  {= <name> <group> / Free}
```

Matrices are defined by their name, type and dimensions (rows & colums) for use in current or subsequent groups. Possible types, structure, shape and number of free elements.

Type	Structure	Shape	# of free elements
`!ero`	Null	Any	0
`Unit`	Unit	Any	0
`Iden`	Identity	Square	0
`I!ero`	Identity | Zero	Any	0
`!Iden`	Zero | Identity	Any	0
`Diag`	Diagonal	Square	r
`SDiag`	Subdiagonal	Square	$r(r-1)/2$
`Stand`	Standardized	Square	$r(r-1)/2$
`Symm`	Symmetric	Square	$r(r+1)/2$
`Lower`	Lower triang.	Square	$r(r+1)/2$
`Full`	Full	Any	$r \times c$

Number of free elements indicates how many elements should be supplied. All changeable elements of matrices are initialized at zero and are fixed parameters, unless the `Free` keyword is used, in which case each changeable element is specified as a different free parameter.

Special codes for constraining a matrix to equal one previously computed or defined in group n. Note that none of the equalities may refer to groups that appear after the current group.

Symbol	Matrix quantity	Dimensions
`=%O`n	Observed cov./data	$\mathrm{NI}_n \times \mathrm{NI}_n$
`=%E`n	Expected cov. matrix	$\mathrm{NI}_n \times \mathrm{NI}_n$
`=%M`n	Expected mean vector	$1 \times \mathrm{NI}_n$
`=%P`n	Expected proportions	$\mathrm{NR}_n \times \mathrm{NC}_n$
`=%F`n	Function value	1×1

3.2 Covariances/Model/M2 command

Syntax: `Covariances {funct} <matrix name>`
`{operator <matrix name> ..} /`

The covariances command may be formed with any syntactically correct combination of matrices (specified by matrices command), operators and functions. The command may extend over several lines and must end in a /.

`Compute` is the recommended keyword for calculation groups, to make reading scripts easier for humans.

3.3 Means/M1 command

Syntax: `Means <matrix name> .. /`

Means will be used in covariance analysis if both a means model and observed means are supplied. Raw data analysis requires a model for means.

3.4 Threshold command

Syntax: `Threshold <matrix name> .. /`

Thresholds can be used only when fitting to contingency table data. Special restrictions apply to the dimensions of the matrix calculated in the threshold command. The resulting matrix must have 2 rows and at least d columns where $d = \max((r-1),(c-1))$ and the elements are the row and column thresholds respectively.

3.5 Matrix operators and functions

Unary and binary matrix operators are used to perform operations on or between matrices, defined by matrices command. Operations with lower priority are evaluated first, equal priority operations are carried out from left to right. Parentheses may be used to change order of evaluation.

Symbol	Operation	Priority
`~`	Inversion $^{-1}$	1
`'`	Transposition	1
`!`	Power Element	2
`*`	Multiplication	3
`.`	Dot product	3
`@`	Kronecker product $\otimes$	3
`%`	Element Division $\div$	3
`+`	Addition	4
`-`	Subtraction	4
`|`	Horizontal adhesion	5
`!`	Vertical adhesion	6

Matrix functions are defined by a keyword starting with a backslash and followed by an *argument* enclosed in brackets. This argument may be a single matrix name or a complex matrix formula. The expression within brackets will be evaluated prior to the function evaluation.

Keyword	Function	Restrictions
`!tr()`	Trace	$r = c$
`!det()`	Determinant	$r = c$
`!sum()`	Sum	None
`!prod()`	Product	None
`!max()`	Maximum	None
`!min()`	Minimum	None
`!abs()`	Absolute value	None
`!cos()`†	Cosine	None
`!sin()`†	Sin	None
`!tan()`†	Tan	None
`!exp()`	Exponent ($e^{\mathbf{A}}$)	None
`!ln()`	Natural logarithm	None
`!sqrt()`	Square root	None
`!d2v()`	Diagonal to Vector	None
`!v2d()`	Vector to Diagonal	r or $c = 1$
`!m2v()`	Matrix to Vector	None
`!vec()`	Matrix to Vector	None
`!vech()`	Lower triang to Vec	None
`!stnd()`	Standardize matrix	$r = c$
`!eval()`	Real eigenvalues	$r = c$
`!evec()`	Real eigenvectors	$r = c$
`!ival()`	Imaginary eigenval	$r = c$
`!ivec()`	Imaginary eigenvec	$r = c$

†add h (e.g., sinh) for hyperbolic function

3.6 Putting Numbers in Matrices

3.6.1 Matrix command

Syntax: `Matrix <matrix name>`
`{File=<filename>} <numlist>`

where `<numlist>` *is a free format list of numbers.*

The list of numbers must be equal to the number of free elements of that matrix.

3.6.2 Start and Value commands

Syntax: `Start/Value <value><elemlist>/All`

where `<elemlist>` *consists of matrix elements and may include the* `TO` *keyword*

The `TO` keyword operates differently for start and value, otherwise the commands are synonymous.

3.7 Putting Parameters in Matrices

3.7.1 Specification command

Syntax: `Specification <matrix name>`
`<numlist>`

where `<numlist>` *contains not necessarily distinct integers.*

Specify is a convenient method of defining constraints between parameters. If two elements are given the same value, then the same free parameter is assigned to both elements. A zero indicates that the element does not have a free parameter.

3.7.2 Pattern command

Syntax: `Pattern <matrix name>`
`{File=<filename>} <numlist>`

where `<numlist>` *is a list of* `1`*'s and* `0`*'s*

The pattern command requests a different free parameter for every element with a 1. A zero fixes the corresponding matrix element.

3.7.3 Free, Fix and Equate commands

Syntax: `Free/Fix/Equate <elemlist>`

where `<elemlist>` *is a list of matrix elements*

Equate passes the value and the parameter specification of the first matrix element in the list to the remaining elements in the list. Elements of matrices in the current group may be specified with 2 subscripts; elements in previous groups must be specified with 3 subscripts.

3.7.4 Boundary command

Syntax: `Bound low high <parlist> /All`

where `<parlist>` *is a list of parameter names or a list of parameter specification numbers.*

Free parameters in the list will be bounded to lie between low and high. Negative parameter numbers will bound non-linear constraints.

4 Output Options

4.1 Output, Options and End commands

Syntax: `Options {options}`
`End` *or*
`Output {options}`

The Options and Output line allow the specification of a wide variety of keywords to control the use of non-default fit functions, the amount of statistical output, optimization parameters, filenames for result matrices, etc.

4.2 Fit Functions

The default fit function for a group is set according to the type of data that are read. Note that the method may change between groups.

Input Data	Default Fit Funtion
`CM KM` or `PMatrix`	ML
`CM KM` or `PMatrix` with `ACov`	AWLS
`CM KM` or `PMatrix` with `AVar`	DWLS
`Rawdata` or `VLength`	RM
`CTable`	ML_N

4.3 Statistical Output Options

- Appearance: `NDecimals`, number of decimal places, `Width`, number of colums
- Suppressing Output: `NO!output`, no output for this group
- Residuals: `RSiduals` prints observed matrix, expected matrix and `(O-E)` residuals
- Adjusting Degrees of Freedom: `DFreedom`
- Power Calculations: `Power`
- Confidence Intervals: `CInterval`
- Standard Errors: `SErrors`
- Randomizing Starting Values: `THard`
- Jiggling Parameter Start Values: `Jiggle`

4.4 Optimization Parameters

- `NAG=n` creates output file (`NAGDUMP.OUT`) if $n > 0$
- `Iterations=n` alters maximum numbers of iterations
- Others

4.5 Fitting Submodels

4.5.1 Multiple Fit Option

Syntax: `Option Multiple`

The multiple keyword may be used as an option in the last group of an Mx script. Following optimization, the program is in multiple fit mode, and will accept commands to alter the model specification. Parameters of the model not changed will restart estimation at the previous solution. The only commands that may be used in multiple fit mode to modify matrices are `SP, PA, MA, VA, ST, EQ, FI, FR`. An `End` line is necessary to end the group. A number identifying the group in which a matrix is to be found *must* be placed directly after the keywords `SP, PA` and `MA`, *before* the letter indicating the matrix.

4.5.2 Drop command

Syntax: `Drop {@value} <parlist>`

where `<parlist>` *is a list of parameter numbers and* `{@value}` *an optional value to fix at.*

Drop fixes all occurences of a parameter with that number to zero or to a specified value.

4.5.3 Save and Get command

Syntax: `Save <filename>`
`Get <filename>`

where `<filename>` *is the name of the file to be saved or retrieved - the extension .mxs is recommended.*

When the multiple fit option is implemented, the entire input job specification, date, and estimates may be stored in binary format for rapid retrieval and estimation in subsequent fitting of submodels.

4.5.4 Mx command

Syntax: `M_a= <filename>`

where `a` *is the single-letter name of the matrix to be written, or one of* `%E %M %P %V`.

Mx will write matrices to files, including `%E`, expected covariance matrices, `%M`, expected mean matrices, `%P`, a series of columns of information about the likelihood of individual pedigrees, and `%V`, a variable length file.

4.5.5 Draw command

Syntax: `Draw= <filename>`

Iff you specify a model with three matrices according to RAM theory, then RAMpath graphics files may be written and saved to a file, as input for RAMpath to draw a path diagram.

4.5.6 Error Messages

When Mx runs into something it doesn't understand, it tries to tell you as soon as possible. Sometimes this will be after the error itself, so check the earlier input for warnings & mistakes if the cause is not immediately obvious. The *Just WHAT is this command...* message usually comes when Mx has been given too many numbers or labels for a matrix. You may get a "non-numeric characters" warning if you supply too few numbers, or Mx may run into the end of the file in the vain search for enough numbers. By and large, the error messages are supposed to inform and amuse a little to make them less user-hostile. Just one example.... if you really cannot find another reason to use Mx:

63 *The matrices you wish to subtract are not conformable for subtraction.* If this message doesn't add up to you, go back to elementary school.

5 Using Mx under different Operating Systems

Currently Mx is available for VAX VMS, several unix systems (e.g. Vax BSD4.3, Sun-OS, IBM (RS-6000) AIX, Dec Ultrix, HP), and MSDOS 3.0/ WINDOWS (or higher) for IBM compatible 386SX and above. Porting to other environments presents little difficulty, since the code is written in standard FORTRAN. Versions for most operating systems can be downloaded via binary ftp to sapphire.gen.vcu.edu

5.1 VAX VMS

We recommend that input files have the naming convention `cutename.mx` where `cutename` is a name of your choice. It allows you to run Mx either interactively or as a batch job under VMS with the following syntax:

Syntax: `$ mx cutename`

With this syntax, the output will be in a file called `cutename.mxo`.

5.2 UNIX

Mx may be used very simply in UNIX by typing

Syntax: `mx <inputfile >outputfile &`

where the parameter `&` let you run jobs in batch.

5.3 MSDOS/ WINDOWS

To run Mx on a PC, create an input script `cutename.mx` and type

Syntax: `mx cutename.mx {cutename.mxo}`

if you are running DOS. If you use WINDOWS, you can either associate files with the `.mx` extension with the `mx.exe` file and double click the input file or launch the input file on the Mx icon. Feedback of function evaluations is printed on the screen and an outputfile `cutename.mxo` is automatically created.

6 Examples of Mx Scripts

Single Group Example

```
TITLE: Factor Model
DATA NGroups=1 NInput_vars=3 NObs=100
Cmatrix Symmetric
1.0
 .5  .8
 .4  .2  .7
MATRICES
A Full 3 1
U Diag 3 3 Free
COVARIANCE  A*A' + U /
Specification A
4 5 0
Start .5 A 1 1 - A 2 1
Options ML
END
```

Model Fitting To Covariances

```
!______________________________________
!  double cholesky model.
!______________________________________
This group calculates, no fitting
!  Genetic factors
Data Calculation NGroups=5
Matrices
 H Lower 2 2 Free
Compute H*H' /
End
!______________________________________
Title 2nd Group also for calculation
Data CA
Matrices
 E Lower 2 2 Free
Compute E*E' /  ! Specific Envt.
End
!     Now get to actual data
!______________________________________
Group 3: Unmatched twins
Data NInput_vars=2 NObservations=449
 KMatrix Symmetric
 1.
 -.22 1.
 ACov File=unm.asy
MAtrices
 G Symm 2 2 = %E1
 E Symm 2 2 = %E2
Covariances (G + E) /
End
!______________________________________
!______________________________________
Group 4: MZ twins with cotwins
DAta NInput_vars=6 NObservations=456
 CMatrix File=mz.cov
 Select 1 2 4 5/
Matrices
 G Symm 2 2 = %E1
 E Symm 2 2 = %E2
Covariances (g+e | g_
             g   | g+e) /
End
!______________________________________
Group 5: DZ twins with cotwins
Data NInput_vars=6 NObservations=357
 CMatrix File=dz.cov
 Labels Ex1 Ne1 De1 Ex2 Ne2 De2
 Select ex1 ne1 ex2 ne2 /
Matrices
 G Symm 2 2 = %E1
 E Symm 2 2 = %E2
 H Diag 1 1
Covariances
 (g+e | h@g _
  h@g | g+e) /
 Matrix Half .5
 Start .5 ALL
 Boundary -1 1 All
Options RSiduals Iterations=200
End
```

Categorical Data Analysis

```
PACE MODEL:  MZ twins
Data Ninput_vars=2 NGroups=3
CTable 2 2
30  20
19  60
Matrices
A Full 2 6
B Full 2 2
I Iden 2 2
P Symm 6 6
T Full 2 1
Thresholds T /
Covariances (I-B)~*A*P*A'*(I-B)~/
Specify A
1 2 3 0 0 0
0 0 0 1 2 3
Labels Row A At1 Ct1 Et1 A2 C2 E2
Labels Col A Pt1 Pt2
Labels Row P At1 Ct1 Et1 A2 C2 E2
Labels Col P At1 Ct1 Et1 A2 C2 E2
Start .6 A 1 1 - A 2 6
```

```
Specification T 4 5
Pattern B
0 1
1 0
Equate B 2 1 B 1 2
Boundary -.99 .99 6
Matrix P
1
0 1
0 0 1
1 0 0 1
0 1 0 0 1
0 0 0 0 0 1
Options RSidual
End

DZ twins in PACE model
Data Ninput_vars=2
CTable 2 2
30  30
29  50
Matrices
A Full 2 6 =A1
B Full 2 2 =B1
I Iden 2 2
P Stand 6 6
T Full 2 1 =T1
Thresholds T /
Covariances  (I-B)~* A*P*A'* (I-B)~ /
Value .5 P 4 1
Value 1 P 5 2
Options RSiduals
End

Constraint group: a*a+c*c+e*e=1
Data Constraint Ninput=1
Matrices
S Full 1 3
I Identity 1 1
Constraint I - S*S' /
Specification S
1 2 3
Options Multiple None Iterations=300
End
Save pace.mxs
! No common environment
Drop 2
End
! No interaction
Get pace.mxs
Value 0 B 1 1 1 - B 1 2 2
Options NAG=30
End
```

Variable Pedigree Size ML Example

```
Estimate means and cholesky
Data NInput_vars=3 NGroups=1
VLength File=[neale]unbalanced.raw
Matrices
M Full 1 3 FRee
S Lower 3 3 FRee
Means M /  ! Means model
Covariances S*S' /
Start .7 s 1 1 - s 3 3
Options RM MX%E=unbal.cov THard=-1
End
```

`First few lines of unbalanced.raw`

```
2
1 2   0.5550   -1.1114
3
1 2 3   1.6442   -0.1728  3.69
2
1 3  -0.2145   5.01
```

Index