

Title

intro — Introduction to the Mata manual

Contents

Section	Description
[M-1]	Introduction and advice
[M-2]	Language definition
[M-3]	Commands for controlling Mata
[M-4]	Index and guide to functions
[M-5]	Functions
[M-6]	Mata glossary of common terms

Description

Mata is a matrix programming language that can be used by those who want to perform matrix calculations interactively and by those who want to add new features to Stata.

This entry describes this manual and what has changed since Stata 11.

Remarks

This manual is divided into six sections. Each section is organized alphabetically, but there is an introduction in front that will help you get around.

If you are new to Mata, here is a helpful reading list. Start by reading

[M-1] first	Introduction and first session
[M-1] interactive	Using Mata interactively
[M-1] how	How Mata works

You may find other things in section [M-1] that interest you. For a table of contents, see

[M-1] intro	Introduction and advice
--------------------	-------------------------

Whenever you see a term that you are unfamiliar with, see

[M-6] Glossary	Mata glossary of common terms
-----------------------	-------------------------------

Now that you know the basics, if you are interested, you can look deeper into Mata's programming features:

[M-2] **syntax** Mata language grammar and syntax

[M-2] **syntax** is pretty dense reading, but it summarizes nearly everything. The other entries in [M-2] repeat what is said there but with more explanation; see

[M-2] **intro** Language definition

because other entries in [M-2] will interest you. If you are interested in object-oriented programming, be sure to see [M-2] **class**.

Along the way, you will eventually be guided to sections [M-4] and [M-5]. [M-5] documents Mata's functions; the alphabetical order makes it easy to find a function if you know its name but makes learning what functions there are hopeless. That is the purpose of [M-4]—to present the functions in logical order. See

[M-4] **intro** Index and guide to functions

Mathematical

[M-4] **matrix** Matrix functions
[M-4] **solvers** Matrix solvers and inverters
[M-4] **scalar** Scalar functions
[M-4] **statistical** Statistical functions
[M-4] **mathematical** Other important functions

Utility and manipulation

[M-4] **standard** Functions to create standard matrices
[M-4] **utility** Matrix utility functions
[M-4] **manipulation** Matrix manipulation functions

Stata interface

[M-4] **stata** Stata interface functions

String, I/O, and programming

[M-4] **string** String manipulation functions
[M-4] **io** I/O functions
[M-4] **programming** Programming functions

What's new

This section is intended for previous Stata users. If you are new to Stata, you may as well skip it.

1. **New Stata commands `getmata` and `putmata`** make it easy to transfer your data into Mata, manipulate them, and then transfer them back to Stata. `getmata` and `putmata` are especially designed for interactive use. See [D] [putmata](#).
 2. **New functions** imported from Stata,
 - a. **Tukey's Studentized range**, cumulative and inverse, `tukeyprob()` and `invtukeyprob()`.
 - b. **Dunnett's multiple range**, cumulative and inverse, `dunnettprob()` and `invdunnettprob()`.
 - c. **New date conversion functions** `dofb()` and `bofd()` convert between business dates and standard calendar dates. See [D] [datetime business calendars](#).
- See [D] [functions](#), [M-5] [normal\(\)](#), and [M-5] [date\(\)](#).
3. **Support for hidden and historical saved results.** Existing Mata functions `st_global()`, `st_numscalar()`, and `st_matrix()` now allow an optional third argument specifying the hidden or historical status. Three new functions—`st_global_hcat()`, `st_numscalar_hcat()`, `st_matrix_hcat()`—allow you to determine the saved hidden or historical status. See [M-5] [st_global\(\)](#), [M-5] [st_numscalar\(\)](#), and [M-5] [st_matrix\(\)](#).
 4. **Support for new ml features.** Stata's `ml` now distinguishes the Hessian matrix produced by `technique(nr)` from the other techniques that compute a substitute for the Hessian matrix. This means that `ml` will compute the real Hessian matrix of second derivatives to determine convergence when all other convergence tolerances are satisfied and `technique(bfgs)`, `technique(bhhh)`, or `technique(dfpr)` is in effect.

Mata's commands `optimize()` and `moptimize()` have been similarly changed. See [M-5] [optimize\(\)](#) and [M-5] [moptimize\(\)](#).

Also see

[M-1] [first](#) — Introduction and first session

[M-6] [Glossary](#)