Title

eltype( ) — Element type and organizational type of object

Syntax Description Remarks and examples Conformability
Diagnostics Also see

Syntax

string scalar `eltype(`*X*`)`

string scalar `orgtype(`*X*`)`

Description

`eltype()` returns the current *eltype* of the argument.

`orgtype()` returns the current *orgtype* of the argument.

See [M-6] **Glossary** for a definition of *eltype* and *orgtype*.

Remarks and examples

If X is a matrix (syntax 1), returned is

`eltype(`*X*`)`	`orgtype(`*X*`)`
`real`	`scalar`
`complex`	`rowvector`
`string`	`colvector`
`pointer`	`matrix`
`struct`	
`class`	

The returned value reflects the current contents of X. That is, X might be declared a `transmorphic matrix`, but at any instant, it contains something, and if that something were 5, returned would be `"real"` and `"scalar"`.

For `orgtype()`, returned is `"scalar"` if the object is currently 1×1; `"rowvector"` if it is $1 \times k$, $k \neq 1$; `"colvector"` if it is $k \times 1$, $k \neq 1$; and `"matrix"` otherwise (it is $r \times c$, $r \neq 1$, $c \neq 1$).

X can be a function (syntax 2). Returned is

`eltype(*(&`*func*`()))`	`orgtype(*(&`*func*`()))`
`transmorphic`	`matrix`
`numeric`	`vector`
`real`	`rowvector`
`complex`	`colvector`
`string`	`scalar`
`pointer`	`void`
`struct`	
`structdef`	
`class`	
`classdef`	

These types are obtained from the declaration of the function.

Aside: `struct` and `structdef` have to do with structures; see [M-2] **struct**. `structdef` indicates that the function not only returns a structure but is the routine that defines the structure as well. `class` and `classdef` have to do with Mata classes; see [M-2] **class**. `classdef` indicates the function not only returns a class but is the routine that defines the class as well.

Conformability

`eltype(`*X*`)`, `orgtype(`*X*`)`:

X:	$r \times c$
result:	1×1

Diagnostics

None.

Also see

[M-5] **isreal()** — Storage type of matrix

[M-5] **isview()** — Whether matrix is view

[M-4] **utility** — Matrix utility functions