Title

op_colon — Colon operators

Description

Colon operators perform element-by-element operations.

Syntax

a	`:+`	*b*	addition
a	`:-`	*b*	subtraction
a	`:*`	*b*	multiplication
a	`:/`	*b*	division
a	`:^`	*b*	power
a	`:==`	*b*	equality
a	`:!=`	*b*	inequality
a	`:>`	*b*	greater than
a	`:>=`	*b*	greater than or equal to
a	`:<`	*b*	less than
a	`:<=`	*b*	less than or equal to
a	`:&`	*b*	and
a	`:|`	*b*	or

Remarks and examples

stata.com

Remarks are presented under the following headings:

C-conformability: element by element

The colon operators perform the indicated operation on each pair of elements of *a* and *b*. For instance,

$$\begin{bmatrix} c & d \\ f & g \\ h & i \end{bmatrix} \texttt{:*} \begin{bmatrix} j & k \\ l & m \\ n & o \end{bmatrix} = \begin{bmatrix} c*j & d*k \\ f*l & g*m \\ h*n & i*o \end{bmatrix}$$

Also colon operators have a relaxed definition of conformability:

$$\begin{bmatrix} c \\ f \\ g \end{bmatrix} :* \begin{bmatrix} j & k \\ l & m \\ n & o \end{bmatrix} = \begin{bmatrix} c*j & c*k \\ f*l & f*m \\ g*n & g*o \end{bmatrix}$$

$$\begin{bmatrix} c & d \\ f & g \\ h & i \end{bmatrix} :* \begin{bmatrix} j \\ l \\ n \end{bmatrix} = \begin{bmatrix} c*j & d*j \\ f*l & g*l \\ h*n & i*n \end{bmatrix}$$

$$\begin{bmatrix} c & d \end{bmatrix} :* \begin{bmatrix} j & k \\ l & m \\ n & o \end{bmatrix} = \begin{bmatrix} c*j & d*k \\ c*l & d*m \\ c*n & d*o \end{bmatrix}$$

$$\begin{bmatrix} c & d \\ f & g \\ h & i \end{bmatrix} :* \begin{bmatrix} l & m \end{bmatrix} = \begin{bmatrix} c*l & d*m \\ f*l & g*m \\ h*l & i*m \end{bmatrix}$$

$$c :* \begin{bmatrix} j & k \\ l & m \\ n & o \end{bmatrix} = \begin{bmatrix} c*j & c*k \\ c*l & c*m \\ c*n & c*o \end{bmatrix}$$

$$\begin{bmatrix} c & d \\ f & g \\ h & i \end{bmatrix} :* j = \begin{bmatrix} c*j & d*j \\ f*j & g*j \\ h*j & i*j \end{bmatrix}$$

The matrices above are said to be c-conformable; the c stands for colon. The matrices have the same number of rows and columns, or one or the other is a vector with the same number of rows or columns as the matrix, or one or the other is a scalar.

C-conformability is relaxed, but not everything is allowed. The following is an error:

$$(c\ d\ e) :* \begin{bmatrix} f \\ g \\ h \end{bmatrix}$$

Usefulness of colon logical operators

It is worth paying particular attention to the colon logical operators because they can produce pattern vectors and matrices. Consider the matrix

```
: x = (5, 0 \ 0, 2 \ 3, 8)
: x
        1   2
    +---------+
  1 |   5   0 |
  2 |   0   2 |
  3 |   3   8 |
    +---------+
```

Which elements of x contain 0?

```
: x:==0
        1   2
    +---------+
  1 |   0   1 |
  2 |   1   0 |
  3 |   0   0 |
    +---------+
```

How many zeros are there in x?

```
: sum(x:==0)
  2
```

Use parentheses

Because of their relaxed conformability requirements, colon operators are not associative even when the underlying operator is. For instance, you expect $(a+b)+c$ == $a+(b+c)$, at least ignoring numerical roundoff error. Nevertheless, $(a$:+$b)$:+c == a:+$(b$:+$c)$ does not necessarily hold. Consider what happens when

a:	1×4
b:	5×1
c:	5×4

Then $(a$:+$b)$:+c is an error because a:+b is not c-conformable.

Nevertheless, a:+$(b$:+$c)$ is not an error and in fact produces a 5×4 matrix because b:+c is 5×4, which is c-conformable with a.

For nonassociative operations, parentheses are useful when using colon operators for even the most basic computations. For example, consider the column vectors

```
: x = (4 \ 5 \ 6)
: y = (1 \ 2 \ 3)
```

Below, we attempt to compute 4-x-y with two different statements. The actual computations that are performed are listed as comments:

```
: 4 :- x :- y    /* (4-x)-y */
         1
    +------+
  1 |   -1 |
  2 |   -3 |
  3 |   -5 |
    +------+

: 4 :- x - y     /* 4-(x-y) */
        1
    +-----+
  1 |   1 |
  2 |   1 |
  3 |   1 |
    +-----+
```

As stated in [M-2] **Syntax**, an operator preceded by a colon (that is, a colon operator) has lower precedence than the operator itself. This is why Mata first subtracts y from x in the second statement above. But, if you plan to use a combination of operators and colon operators, you can still set the precedence with parentheses:

```
: (4 :- x) - y  /* (4-x)-y */
        1
    ┌──────┐
  1 │  -1  │
  2 │  -3  │
  3 │  -5  │
    └──────┘
```

This produces the desired result and the same output as the first statement above.

Conformability

a :*op* *b*:

a:	$r_1 \times c_1$
b:	$r_2 \times c_2$, *a* and *b* c-conformable
result:	$\texttt{max}(r_1,r_2) \times \texttt{max}(c_1,c_2)$

Diagnostics

The colon operators return missing and abort with error under the same conditions that the underlying operator returns missing and aborts with error.

Also see

[M-2] **exp** — Expressions

[M-2] **Intro** — Language definition