runningsum( ) — Running sum of vector

Description Syntax Remarks and examples Conformability Diagnostics Also see

Description

`runningsum(`*x*`)` returns a vector of the same dimension as *x* containing the running sum of *x*. Missing values are treated as contributing zero to the sum.

`runningsum(`*x*`,` *missing*`)` does the same but lets you specify how missing values are treated. `runningsum(`*x*`, 0)` is the same as `runningsum(`*x*`)`. `runningsum(`*x*`, 1)` specifies that missing values are to turn the sum to missing where they occur.

`quadrunningsum(`*x*`)` and `quadrunningsum(`*x*`,` *missing*`)` do the same but perform the accumulation in quad precision.

`_runningsum(`*y*`,` *x* [`,` *missing*]`)` and `_quadrunningsum(`*y*`,` *x* [`,` *missing*]`)` work the same way, except that rather than returning the running-sum vector, they store the result in *y*. This method is slightly more efficient when *y* is a view.

Syntax

numeric vector	`runningsum(`*numeric vector x* [`,` *missing*]`)`
numeric vector	`quadrunningsum(`*numeric vector x* [`,` *missing*]`)`
void	`_runningsum(`*y*`,` *numeric vector x* [`,` *missing*]`)`
void	`_quadrunningsum(`*y*`,` *numeric vector x* [`,` *missing*]`)`

where optional argument *missing* is a *real scalar* that determines how missing values in *x* are treated:

1. Specifying *missing* as 0 is equivalent to not specifying the argument; missing values in *x* are treated as contributing 0 to the sum.
2. Specifying *missing* as 1 specifies that missing values in *x* are to be treated as missing values and turn the sum to missing.

Remarks and examples

The running sum of (1, 2, 3) is (1, 3, 6).

All functions return the same type as the argument, real if argument is real, complex if complex.

Conformability

`runningsum(`*x*`, `*missing*`)`, `quadrunningsum(`*x*`, `*missing*`)`:

x:	$r \times 1$	or	$1 \times c$	
missing:	1×1			(optional)
result:	$r \times 1$	or	$1 \times c$	

`_runningsum(`*y*`, `*x*`, `*missing*`)`, `_quadrunningsum(`*y*`, `*x*`, `*missing*`)`:

input:				
x:	$r \times 1$	or	$1 \times$ c	
y:	$r \times 1$	or	$1 \times$ c	(contents irrelevant)
missing:	1×1			(optional)
output:				
y:	$r \times 1$	or	$1 \times$ c	

Diagnostics

If *missing* $= 0$, missing values are treated as contributing zero to the sum; they do not turn the sum to missing. Otherwise, missing values turn the sum to missing.

`_runningsum(`*y*`, `*x*`, `*missing*`)` and `_quadrunningsum(`*y*`, `*x*`, `*missing*`)` abort with error if *y* is not p-conformable with *x* and of the same eltype. The contents of *y* are irrelevant.

Also see

[M-5] **sum( )** — Sums

[M-4] **Mathematical** — Important mathematical functions

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