

trunc() — Round to integer

[Description](#)
[Diagnostics](#)

[Syntax](#)
[Also see](#)

[Remarks and examples](#)

[Conformability](#)

Description

These functions convert noninteger values to integers by moving toward 0, moving down, moving up, or rounding. These functions are typically used with scalar arguments, and they return a scalar in that case. When used with vectors or matrices, the operation is performed element by element.

`trunc(R)` returns the integer part of R .

`floor(R)` returns the largest integer i such that $i \leq R$.

`ceil(R)` returns the smallest integer i such that $i \geq R$.

`round(R)` returns the integer closest to R .

`round(R , U)` returns the values of R rounded in units of U and is equivalent to `round((R :/ U))* U` . For instance, `round(R , 2)` returns R rounded to the closest even number. `round(R , .5)` returns R rounded to the closest multiple of one half. `round(R , 1)` returns R rounded to the closest integer and so is equivalent to `round(R)`.

Syntax

real matrix `trunc(real matrix R)`

real matrix `floor(real matrix R)`

real matrix `ceil(real matrix R)`

real matrix `round(real matrix R)`

real matrix `round(real matrix R , real matrix U)`

Remarks and examples

stata.com

Remarks are presented under the following headings:

[Relationship to Stata's functions](#)
[Examples of rounding](#)

Relationship to Stata's functions

`trunc()` is equivalent to Stata's `int()` function.

`floor()`, `ceil()`, and `round()` are equivalent to Stata's functions of the same name.

Examples of rounding

x	<code>trunc(x)</code>	<code>floor(x)</code>	<code>ceil(x)</code>	<code>round(x)</code>
1	1	1	1	1
1.3	1	1	2	1
1.6	1	1	2	2
-1	-1	-1	-1	-1
-1.3	-1	-2	-1	-1
-1.6	-1	-2	-1	-2

Conformability

`trunc(R)`, `floor(R)`, `ceil(R)`:

R : $r \times c$
result: $r \times c$

`round(R)`:

R : $r \times c$
result: $r \times c$

`round(R, U)`:

R : $r_1 \times c_1$
 U : $r_2 \times c_2$, R and U r-conformable
result: $\max(r_1, r_2) \times \max(c_1, c_2)$

Diagnostics

Most Stata and Mata functions return missing when arguments contain missing, and in particular, return `.` whether the argument is `.`, `.a`, `.b`, `...`, `.z`. The logic is that performing the operation on a missing value always results in the same missing-value result. For example, `sqrt(.a)==.`

These functions, however, when passed a missing value, return the particular missing value. Thus `trunc(.a)==.a`, `floor(.b)==.b`, `ceil(.c)==.c`, and `round(.d)==.d`.

For `round()` with two arguments, this applies to the first argument and only when the second argument is not missing. If the second argument is missing (whether `.`, `.a`, `...`, or `.z`), then `.` is returned.

For `round()` with one or two arguments, values of R exactly at midpoints (where it may not be clear whether to round up or down) are always rounded up to the larger value. For example, `round(4.5)` is 5 and `round(-4.5)` is -4 . Note that rounding a number is based on the floating-point number representation of the number instead of the number itself. So `round()` is sensitive to representation errors and precision limits. For example, 0.15 has no exact floating-point number representation. Therefore, `round(0.15, 0.1)` is 0.1 instead of 0.2.

Also see

[M-4] [Scalar](#) — Scalar mathematical functions