F_ROUNDTOEVEN

function from adhoc

Entrypoint roundtoeven compatible with UTF-8

Inputs/Outputs

Input	DOUBLE	floatingpoint to be round
	INTEGER	number of digits to be round
Output	DOUBLE	floatingpoint rounded (round to even method)

Syntax

Round to even or bankers' rounding. Described in IEEE-754-Standard for calculating with binary floating-point numbers in computers.
From Wikipedia http://en.wikipedia.org/wiki/Rounding:
This method is also known as statistician's rounding or as bankers' rounding. It is identical to the common method of rounding except when the digit(s) following the rounding digit start with a five and have no non-zero digits after it. The new algorithm is:
* Decide which is the last digit to keep.
* Increase it by 1 if the next digit is 6 or more, or a 5 followed by one or more non-zero digits.
* Leave it the same if the next digit is 4 or less
* Otherwise, all that follows the last digit is a 5 and possibly trailing zeroes; then change the last digit to the nearest even digit. That is, increase the rounded digit if it is currently odd; leave it if it is already even.
With all rounding schemes, there are two possible outcomes: increasing the rounding digit by one or leaving it alone. With traditional rounding, if the number has a value less than the half-way mark between the possible outcomes, it is rounded down; if the number has a value exactly half-way or greater
than half-way between the possible outcomes, it is rounded up. The round-to-even method is the same except that numbers exactly half-way between the possible outcomes are sometimes rounded up—sometimes down.
Although it is customary to round the number 4.5 up to 5, in fact 4.5 is no nearer to 5 than it is to 4 (it is 0.5 away from either). When dealing with large sets of scientific or statistical data, where trends are important, traditional rounding on average biases the data upwards slightly. Over a large
set of data, or when many subsequent rounding operations are performed as

in digital signal processing, the round-to-even rule tends to reduce the total rounding error, with (on average) an equal portion of numbers rounding up as rounding down. This generally reduces the upwards skewing of the result.

   Round-to-even is used rather than round-to-odd as the latter rule would prevent rounding to a result of zero.

   Examples:

      * 3.016 rounded to hundredths is 3.02 (because the next digit (6) is 6 or more)

      * 3.013 rounded to hundredths is 3.01 (because the next digit (3) is 4 or less)

      * 3.015 rounded to hundredths is 3.02 (because the next digit is 5, and the hundredths digit (1) is odd)

      * 3.045 rounded to hundredths is 3.04 (because the next digit is 5, and the hundredths digit (4) is even)

      * 3.04501 rounded to hundredths is 3.05 (because the next digit is 5, but it is followed by non-zero digits)

   Negative numbers are rounded like their absolute values:

      * −2,35 rounded to tenths is −2,4

      * −2,45 (exact) rounded to tenths is −2,4

      * −2,4500001 rounded to tenths is −2,5

      * −2,46 rounded to tenths is −2,5

      * −2,449 rounded to tenths is −2,4

   TestSQL

   SELECT 21.14 AS ISCORRECT, F_ROUNDTOEVEN(21.145, 2) FROM RDB$DATABASE

   SELECT 215.14 AS ISCORRECT, F_ROUNDTOEVEN(215.145, 2) FROM RDB$DATABASE

   SELECT 215.16 AS ISCORRECT, F_ROUNDTOEVEN(215.155, 2) FROM RDB$DATABASE

   SELECT 3.02 AS ISCORRECT, F_ROUNDTOEVEN(3.016, 2) FROM RDB$DATABASE;

   SELECT 3.01 AS ISCORRECT, F_ROUNDTOEVEN(3.013, 2) FROM RDB$DATABASE;

   SELECT 3.02 AS ISCORRECT, F_ROUNDTOEVEN(3.015, 2) FROM RDB$DATABASE;

   SELECT 3.04 AS ISCORRECT, F_ROUNDTOEVEN(3.045, 2) FROM RDB$DATABASE;

   SELECT 3.05 AS ISCORRECT, F_ROUNDTOEVEN(3.04501, 2) FROM RDB$DATABASE;

   SELECT -2.4 AS ISCORRECT, F_ROUNDTOEVEN(-2.35, 1) FROM RDB$DATABASE;

   SELECT -2.4 AS ISCORRECT, F_ROUNDTOEVEN(-2.45, 1) FROM RDB$DATABASE;

   SELECT -2.5 AS ISCORRECT, F_ROUNDTOEVEN(-2.4500001, 1) FROM RDB$DATABASE;

   SELECT -2.4 AS ISCORRECT, F_ROUNDTOEVEN(-2.449, 1) FROM RDB$DATABASE;

   SELECT NULL AS ISCORRECT, F_ROUNDTOEVEN(NULL, NULL) FROM RDB$DATABASE;