How do you round numbers in SAS?

Authored by

stats writer

July 1, 2024

RECOMMENDED CITATION

stats writer (2024). *How do you round numbers in SAS?*. PSYCHOLOGICAL SCALES. Retrieved from https://scales.arabpsychology.com/?p=164245

In SAS, rounding numbers is a simple process that involves using the ROUND function. This function allows you to specify the number of decimal places to round to, as well as the direction of rounding (up or down). Additionally, SAS also has a variety of other rounding functions, such as CEIL, FLOOR, and TRUNC, which offer different methods of rounding. By using these functions, you can easily and accurately round numbers in SAS for a variety of data analysis and reporting purposes.

Round Numbers in SAS (4 Examples)

You can use the following methods to round numbers in SAS:

Method 1: Round to Nearest Integer

```
data new_data;
set original_data;
new_value = round(value);
run;
```

Method 2: Round to Specific Decimal Places

```
data new_data;
set original_data;
new_value1 = round(value, .1); /*round to 1 decimal place*/
new_value2 = round(value, .01); /*round to 2 decimal places*/
```

```
new_value3 = round(value, .001); /*round to 3 decimal places*/run;
```

Method 3: Round All Values Down (Or Up) to Next Integer

```
data new_data;
set original_data;
new_value1 = floor(value); /*round down to next integer*/
new_value2 = ceil(value); /*round up to next integer*/run;
```

Method 4: Round to Nearest Multiple

```
data new_data;
set original_data;
new_value1 = round(value, 10); /*round to nearest multiple of 10*/
new_value2 = round(value, 100); /*round to nearest multiple of 100*/run;
```

The following examples show how to use each method with the following dataset in SAS:

```
/*create dataset*/
data original_data;
input value;
datalines;
0.33
0.9
1.2593
1.61
2.89
4.3
8.8
14.4286
18.2
51.4
;
run;

/*view dataset*/
proc printdata=original_data;
```

Obs	value
1	0.3300
2	0.9000
3	1.2593
4	1.6100
5	2.8900
6	4.3000
7	8.8000
8	14.4286
9	18.2000
10	51.4000

Example 1: Round to Nearest Integer

The following code shows how to round each value to the nearest integer:

```
/*round to nearest integer*/
data new_data;
set original_data;
new_value = round(value);
run;

/*view new dataset*/
proc printdata=new_data;
```

Obs	value	new_value
1	0.3300	0
2	0.9000	1
3	1.2593	1
4	1.6100	2
5	2.8900	3
6	4.3000	4
7	8.8000	9
8	14.4286	14
9	18.2000	18
10	51.4000	51

Example 2: Round to Specific Decimal Places

```
data new_data;
set original_data;
new_value1 = round(value, .1); /*round to 1 decimal place*/
new_value2 = round(value, .01); /*round to 2 decimal places*/
new_value3 = round(value, .001); /*round to 3 decimal places*/run;

/*view new dataset*/
proc printdata=new_data;
```

Obs	value	new_value1	new_value2	new_value3
1	0.3300	0.3	0.33	0.330
2	0.9000	0.9	0.90	0.900
3	1.2593	1.3	1.26	1.259
4	1.6100	1.6	1.61	1.610
5	2.8900	2.9	2.89	2.890
6	4.3000	4.3	4.30	4.300
7	8.8000	8.8	8.80	8.800
8	14.4286	14.4	14.43	14.429
9	18.2000	18.2	18.20	18.200
10	51.4000	51.4	51.40	51.400

Example 3: Round All Values Down (Or Up) to Next Integer

The following code shows how to round all values down (or up) to the next integer using the floor() and ceil() functions:

```
data new_data;
set original_data;
new_value1 = floor(value); /*round down to next integer*/
new_value2 = ceil(value); /*round up to next integer*/run;

/*view new dataset*/
proc printdata=new_data;
```

Obs	value	floor_value	ceil_value
1	0.3300	0	1
2	0.9000	0	1
3	1.2593	1	2
4	1.6100	1	2
5	2.8900	2	3
6	4.3000	4	5
7	8.8000	8	9
8	14.4286	14	15
9	18.2000	18	19
10	51.4000	51	52

Method 4: Round to Nearest Multiple

The following code shows how to round all values to the nearest multiple of some value:

```
data new_data;
set original_data;
nearest10 = round(value, 10); /*round to nearest multiple of 10*/
nearest100 = round(value, 100); /*round to nearest multiple of 100*/run;

/*view new dataset*/
proc printdata=new_data;
```

Obs	value	nearest10	nearest100
1	0.3300	0	0
2	0.9000	0	0
3	1.2593	0	0
4	1.6100	0	0
5	2.8900	0	0
6	4.3000	0	0
7	8.8000	10	0
8	14.4286	10	0
9	18.2000	20	0
10	51.4000	50	100

Additional Resources

The following tutorials explain how to perform other common tasks in SAS: