

How can missing values be replaced with zero in SAS?

Authored by
stats writer

July 1, 2024

RECOMMENDED CITATION

stats writer (2024). *How can missing values be replaced with zero in SAS?*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=164901>

SAS is a powerful analytical tool that allows for the manipulation and analysis of large datasets. In some cases, missing values can occur within these datasets, which can hinder the accuracy and effectiveness of data analysis. To address this issue, SAS provides a simple and efficient method for replacing missing values with zero. This can be done by using the "MISSOVER" and "ZERO" options within the "INPUT" statement. The "MISSOVER" option instructs SAS to read past missing values, while the "ZERO" option replaces any missing values with zeros. By utilizing these options, missing values can be easily and accurately replaced with zeros, ensuring the integrity and accuracy of data analysis in SAS.

Replace Missing Values with Zero in SAS

Often you may want to replace missing values in a SAS dataset with zeros.

Fortunately this is easy to do using a simple if then statement.

The following examples show how to replace missing values with zeros in practice.

Example 1: Replace Missing Values in All Columns

Suppose we have the following dataset in SAS with three columns, each with some missing values:

```
/*create dataset*/  
data my_data;  
input x y z;  
datalines;
```

```
1 . 76
```

```
2 3 .
```

```
2 3 85
```

```
4 5 88
```

```
2 2 .
```

```
1 2 69
```

```
5 . 94
```

```
4 1 .
```

```
. . 88
```

```
4 3 92
```

```
;
```

```
run;
```

```
/*view dataset*/
```

```
proc printdata=my_data;
```

Obs	x	y	z
1	1	.	76
2	2	3	.
3	2	3	85
4	4	5	88
5	2	2	.
6	1	2	69
7	5	.	94
8	4	1	.
9	.	.	88
10	4	3	92

We can use the following code to replace the missing values with zeros in every column of the dataset:

```
/*create new dataset with missing values replaced by zero*/
```

```
data my_data_new;
```

```
set my_data;
```

```
array variablesOfInterest _numeric_;
```

```
do over variablesOfInterest;
```

```
if variablesOfInterest=. then variablesOfInterest=0;
```

```
end;
```

```
run;
```

```
/*view new dataset*/
```

```
proc printdata=my_data_new;
```

Obs	x	y	z
1	1	0	76
2	2	3	0
3	2	3	85
4	4	5	88
5	2	2	0
6	1	2	69
7	5	0	94
8	4	1	0
9	0	0	88
10	4	3	92

Notice that the missing values in each column have been replaced with zeros.

Note: The argument `_numeric_` tells SAS to replace the missing values with zeros in every numeric column in the dataset.

Example 2: Replace Missing Values in Specific Column

Once again suppose we have the following dataset in SAS with three columns, each with some missing values:

```
/*create dataset*/  
data my_data;  
input x y z;  
datalines;  
1 . 76  
2 3 .  
2 3 85  
4 5 88  
2 2 .  
1 2 69  
5 . 94  
4 1 .
```

```
.. 88
```

```
4 3 92
```

```
;
```

```
run;
```

```
/*view dataset*/
```

```
proc printdata=my_data;
```

Obs	x	y	z
1	1	.	76
2	2	3	.
3	2	3	85
4	4	5	88
5	2	2	.
6	1	2	69
7	5	.	94
8	4	1	.
9	.	.	88
10	4	3	92

We can use the following code to replace the missing values with zeros in only the "y" column of the dataset:

```
/*create new dataset with missing values in "y" column  
replaced by zero*/
```

```
data my_data_new;
```

```
set my_data;
```

```
array variablesOfInterest y;  
do over variablesOfInterest;  
if variablesOfInterest=. then variablesOfInterest=0;  
end;  
run;
```

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

Obs	x	y	z
1	1	0	76
2	2	3	.
3	2	3	85
4	4	5	88
5	2	2	.
6	1	2	69
7	5	0	94
8	4	1	.
9	.	0	88
10	4	3	92

Notice that only the missing values in the "y" column have been replaced with zeros.

Additional Resources

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