



CHAPTER

11**Missing Values**

<i>Definition</i>	123
<i>Special Missing Values</i>	124
<i>Definition</i>	124
<i>Tips</i>	124
<i>Example</i>	124
<i>Order of Missing Values</i>	125
<i>Numeric Variables</i>	125
<i>Character Variables</i>	126
<i>How to Represent Missing Values in Raw Data</i>	126
<i>How to Set Variable Values to Missing in a DATA Step</i>	126
<i>How to Check for Missing Values in a DATA Step</i>	126
<i>When Variable Values Are Automatically Set to Missing by SAS</i>	127
<i>When Reading Raw Data</i>	127
<i>When Reading a SAS Data Set</i>	128
<i>When Missing Values Are Generated by SAS</i>	128
<i>Propagation of Missing Values in Calculations</i>	128
<i>Illegal Operations</i>	128
<i>Illegal Character-to-Numeric Conversions</i>	129
<i>Special Missing Values</i>	129
<i>Preventing Propagation of Missing Values</i>	129

Definition*missing value*

is a value that indicates that no data value is stored for the variable in the current observation. There are three kinds of missing values:

- numeric
- character
- special numeric.

By default, SAS prints a missing numeric value as a single period (.) and a missing character value as a blank space. See “Special Missing Values” on page 124 for more information about special numeric missing values.

Special Missing Values

Definition

special missing value

is a type of numeric missing value that enables you to represent different categories of missing data by using the letters A-Z or an underscore.

Tips

- Previous releases of SAS software required that you use the uppercase form of a letter to represent special missing values; SAS now accepts either uppercase or lowercase letters. Values are still displayed and printed as uppercase.
- If you do not begin a special numeric missing value with a period, SAS identifies it as a variable name. Therefore, to use a special numeric missing value in a SAS expression or assignment statement, you must begin the value with a period, followed by the letter or underscore, as in the following example:

```
x=.d;
```

- When SAS prints a special missing value, it prints only the letter or underscore.
 - When data values contain characters in numeric fields that you want SAS to interpret as special missing values, use the MISSING statement to specify those characters. For further information on the MISSING statement, see the “Statements” chapter of *SAS Language Reference: Dictionary*.
-

Example

The following example uses data from a marketing research company. Five testers were hired to test five different products for ease of use and effectiveness. If a tester was absent, there is no rating to report, and the value is recorded with an X for “absent.” If the tester was unable to test the product adequately, there is no rating, and the value is recorded with an I for “incomplete test.” The following program reads the data and displays the resulting SAS data set. Note the special missing values in the first and third data lines:

```
data period_a;
  missing X I;
  input Id $4. Foodpr1 Foodpr2 Foodpr3 Coffeem1 Coffeem2;
  datalines;
1001 115 45 65 I 78
1002 86 27 55 72 86
1004 93 52 X 76 88
1015 73 35 43 112 108
1027 101 127 39 76 79
;

proc print data=period_a;
  title 'Results of Test Period A';
  footnote1 'X indicates TESTER ABSENT';
  footnote2 'I indicates TEST WAS INCOMPLETE';
```

```
run;
```

The following output is produced:

Output 11.1 Output with Multiple Missing Values

Results of Test Period A						
Obs	Id	Foodpr1	Foodpr2	Foodpr3	Coffeem1	Coffeem2
1	1001	115	45	65	I	78
2	1002	86	27	55	72	86
3	1004	93	52	X	76	88
4	1015	73	35	43	112	108
5	1027	101	127	39	76	79

X indicates TESTER ABSENT
I indicates TEST WAS INCOMPLETE

Order of Missing Values

Numeric Variables

Within SAS, a missing value for a numeric variable is smaller than all numbers; if you sort your data set by a numeric variable, observations with missing values for that variable appear first in the sorted data set. For numeric variables, you can compare special missing values with numbers and with each other. Table 11.1 on page 125 shows the sorting order of numeric values.

Table 11.1 Numeric Value Sort Order

Sort Order	Symbol	Description
smallest	._	underscore
	.	period
	.A-.Z	special missing values A (smallest) through Z (largest)
	-n	negative numbers
	0	zero
largest	+n	positive numbers

For example, the numeric missing value (.) is sorted before the special numeric missing value .A, and both are sorted before the special missing value .Z. SAS does not distinguish between lowercase and uppercase letters when sorting special numeric missing values.

Note: The numeric missing value sort order is the same regardless of whether your system uses the ASCII or EBCDIC collating sequence. Δ

Character Variables

Missing values of character variables are smaller than any printable character value. Therefore, when you sort a data set by a character variable, observations with missing (blank) values of the BY variable always appear before observations in which values of the BY variable contain only printable characters. However, some usually unprintable characters (for example, machine carriage-control characters and real or binary numeric data that have been read in error as character data) have values less than the blank. Therefore, when your data includes unprintable characters, missing values may not appear first in a sorted data set.

How to Represent Missing Values in Raw Data

Table 11.2 on page 126 shows how to represent each type of missing value in raw data so that SAS will read and store the value appropriately.

Table 11.2 Representing Missing Values

These missing values ...	Are represented by ...	Explanation
Numeric	.	a single decimal point
Character	''	a blank enclosed in quotes
Special	. <i>letter</i>	a decimal point followed by a letter, for example, .B
Special	._	a decimal point followed by an underscore

How to Set Variable Values to Missing in a DATA Step

You can set values to missing within your DATA step by using program statements such as this one:

```
if age<0 then age=.
```

This statement sets the stored value of AGE to a numeric missing value if AGE has a value less than 0.

Note: You can display a missing numeric value with a character other than a period by using the DATA step's MISSING statement or the MISSING= system option. Δ

The following example sets the stored value of NAME to a missing character value if NAME has a value of "none":

```
if name="none" then name=' ';
```

How to Check for Missing Values in a DATA Step

When you check for ordinary missing numeric values, you can use code that is similar to the following:

```
if numvar=. then do;
```

If your data contains special missing values, you can check for either an ordinary or special missing value with a statement that is similar to the following:

```
if numvar<=.z then do;
```

To check for a missing character value, you can use a statement that is similar to the following:

```
if charvar=' ' then do;
```

The MISSING function enables you to check for either a character or numeric missing value, as in:

```
if missing(var) then do;
```

In each case, SAS checks whether the value of the variable in the current observation satisfies the condition specified. If it does, SAS executes the DO group.

Note: Missing values have a value of `false` when you use them with logical operators such as AND or OR. △

When Variable Values Are Automatically Set to Missing by SAS

When Reading Raw Data

At the beginning of each iteration of the DATA step, SAS sets the value of each variable you create in the DATA step to missing, with the following exceptions:

- variables named in a RETAIN statement
- variables created in a SUM statement
- data elements in a _TEMPORARY_ array
- variables created with options in the FILE or INFILE statements
- variables created by the FGET function
- data elements which are initialized in an ARRAY statement
- automatic variables.

SAS replaces the missing values as it encounters values that you assign to the variables. Thus, if you use program statements to create new variables, their values in each observation are missing until you assign the values in an assignment statement, as shown in the following DATA step:

```
data new;
  input x;
  if x=1 then y=2;
  datalines;
4
1
3
1
;
```

This DATA step produces a SAS data set with the following variable values:

```
OBS    X    Y
1      4    .
```

2	1	2
3	3	.
4	1	2

When X equals 1, the value of Y is set to 2. Since no other statements set Y's value when X is not equal to 1, Y remains missing (.) for those observations.

When Reading a SAS Data Set

When variables are read with a SET, MERGE, or UPDATE statement, SAS sets the values to missing only before the first iteration of the DATA step. (If you use a BY statement, the variable values are also set to missing when the BY group changes.) The variables retain their values until new values become available; for example, through an assignment statement or through the next execution of the SET, MERGE, or UPDATE statement. Variables created with options in the SET, MERGE, and UPDATE statements also retain their values from one iteration to the next.

When all of the rows in a data set in a match-merge operation (with a BY statement) have been processed, the variables in the output data set retain their values as described earlier. That is, as long as there is no change in the BY value in effect when all of the rows in the data set have been processed, the variables in the output data set retain their values from the final observation. *FIRST.variable* and *LAST.variable*, the automatic variables that are generated by the BY statement, both retain their values. Their initial value is 1.

When the BY value changes, the variables are set to missing and remain missing because the data set contains no additional observations to provide replacement values. When all of the rows in a data set in a one-to-one merge operation (without a BY statement) have been processed, the variables in the output data set are set to missing and remain missing.

When Missing Values Are Generated by SAS

Propagation of Missing Values in Calculations

SAS assigns missing values to prevent problems from arising. If you use a missing value in an arithmetic calculation, SAS sets the result of that calculation to missing. Then, if you use that result in another calculation, the next result is also missing. This action is called *propagation of missing values*. SAS prints notes in the log to notify you which arithmetic expressions have missing values and when they were created; however, processing continues.

Illegal Operations

SAS prints a note in the log and assigns a missing value to the result if you try to perform an illegal operation, such as the following:

- dividing by zero
- taking the logarithm of zero
- using an expression to produce a number too large to be represented as a floating-point number (known as overflow).

Illegal Character-to-Numeric Conversions

SAS automatically converts character values to numeric values if a character variable is used in an arithmetic expression. If a character value contains nonnumerical information and SAS tries to convert it to a numeric value, a note is printed in the log, the result of the conversion is set to missing, and the `_ERROR_` automatic variable is set to 1.

Special Missing Values

The result of any numeric missing value in a SAS expression is a period. Thus, both special missing values and ordinary numeric missing values propagate as a period.

```
data a;
    x=.d;
    y=x+1;
    put y=;
run;
```

This DATA step results in the following log:

Output 11.2 SAS Log Results for a Missing Value

```
1 data a;
2 x= .d;
3 y=x+1;
4 put y=;
5 run;

y=.
NOTE: Missing values were generated as a result of performing an
operation on missing values.
Each place is given by:
(Number of times) at (Line):(Column).
1 at 3:6
NOTE: The data set WORK.A has 1 observations and 2 variables.
NOTE: DATA statement used:
real time 0.58 seconds
cpu time 0.05 seconds
```

Preventing Propagation of Missing Values

If you do not want missing values to propagate in your arithmetic expressions, you can omit missing values from computations by using the sample statistic functions. For a list of these functions, see the “Sample Statistics” column in Table 6.3 on page 51. For a complete description and examples, see *SAS Language Reference: Dictionary*. For example, consider the following DATA step:

```
data test;
    x=.;
    y=5;
    a=x+y;
    b=sum(x,y);
    c=5;
```

```
c+x;  
put a= b= c=;  
run;
```

Output 11.3 SAS Log Results for a Missing Value in a Statistic Function

```
1 data test;  
2 x=.;  
3 y=5;  
4 a=x+y;  
5 b=sum(x,y);  
6 c=5;  
7 c+x;  
8 put a= b= c=;  
9 run;  
  
a=. b=5 c=5  
NOTE: Missing values were generated as a result of performing  
an operation on missing values.  
Each place is given by:  
(Number of times) at (Line):(Column).  
1 at 4:6  
NOTE: The data set WORK.TEST has 1 observations and 5 variables.  
NOTE: DATA statement used:  
real time 0.11 seconds  
cpu time 0.03 seconds
```

Adding X and Y together in an expression produces a missing result because the value of X is missing. The value of A, therefore, is missing. However, since the SUM function ignores missing values, adding X to Y produces the value 5, not a missing value.

Note: The SUM statement also ignores missing values, so the value of C is also 5. Δ

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS Language Reference: Concepts*, Cary, NC: SAS Institute Inc., 1999. 554 pages.

SAS Language Reference: Concepts

Copyright © 1999 SAS Institute Inc., Cary, NC, USA.

ISBN 1-58025-441-1

All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, by any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute, Inc.

U.S. Government Restricted Rights Notice. Use, duplication, or disclosure of the software by the government is subject to restrictions as set forth in FAR 52.227-19 Commercial Computer Software-Restricted Rights (June 1987).

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, November 1999

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries.® indicates USA registration.

IBM, ACF/VTAM, AIX, APPN, MVS/ESA, OS/2, OS/390, VM/ESA, and VTAM are registered trademarks or trademarks of International Business Machines Corporation. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

The Institute is a private company devoted to the support and further development of its software and related services.