What is *Missing* in SAS®?

Imelda C. Go, South Carolina Department of Education, Columbia, SC

ABSTRACT

Dealing with missing values is an essential part of data processing and analysis. There are currently a total of 28 ways to represent a numeric missing value. The paper discusses how missing values affect input, output, and processing. Topics include the sort order of numeric missing values, the DATA step's MISSING statement, the INFILE statement's MISSOVER option, reading delimited missing data correctly, the MISSING= system option, BY-group processing with missing values, how procedures treat missing values, the propagation of missing values, and single imputation.

Dealing with missing data is an inevitable fact of data processing and analysis. A missing value in SAS means there is no data value.

There are 28 different ways to represent a numeric missing value. There are also many ways to represent a character missing value. A character missing value is a blank enclosed in quotes. Two consecutive quotes as well as two or more blanks enclosed in quotes also produce a character missing value.

```
charactermissingvalue=' ';
charactermissingvalue='';
charactermissingvalue='  ';
charactermissingvalue='   ';
```

The three types of numeric missing values are described below.

Representing Missing Values		
Missing Value Type	**Repre-senta-tion**	**Description**
Regular Numeric	.	Single period
Special Numeric	.a .b .c . . . .x .y .z	Special representation: Single period followed by a letter Special numeric missing values are not case-sensitive (.A is equivalent to .a).
Special Numeric	._	Special representation: Single period followed by an underscore
Character	' '	Blank enclosed in quotes (The length of a character missing value is one and not zero. A string of blanks enclosed in quotes is considered a character missing value and has a length of one regardless of how many blanks are involved.)

SORT ORDER OF NUMERIC VALUES

A missing value is less than any numeric value. Suppose x is a numeric variable. The condition of x<10 is true when x is a missing value or is a nonmissing value less than 10. The numeric missing values also have an order as shown in the following table.

Sort Order of Numeric Values
._
.
.A
.B
.C
.
.
.
.X
.Y
.Z
nonmissing values

The condition below is true for all y values that are numeric missing values. (If the z below was not preceded by a period, z would be interpreted as a variable name.)

```
y<=.z
```

Because the missing value is less than any numeric value, the following type of error is common for those who are not aware of this fact. A condition might be written to include missing values when the intent is to test the condition only for nonmissing values. In the example below, x values can take on regular numeric missing and nonmissing values.

```
if x<10 then flag='no';
   else flag='yes';
```

The above IF-THEN/ELSE statement divides the x values into two subsets: x values less than 10 for the IF condition and those that are greater than or equal to 10 for the implied ELSE condition. If the intent is for the above to apply only to nonmissing values of x, then there is a problem. If missing values need to be excluded, the following code does so.

```
if .<x<10 then flag='no';
   else if x>=10 then flag='yes';
```

The first IF condition is .<x<10 and the second IF condition is x>=10. Without x>=10, all values of x that do not satisfy .<x<10 are missing values OR are values greater than or equal to 10. Hence, it is important to be aware of how missing values satisfy the conditions being written. Conditions should reflect the true intent. A safe strategy is to make it clear in the SAS code (via programming statements or comments) if missing values are to be processed or not.

UTILITY OF SPECIAL NUMERIC MISSING VALUES

Using special numeric missing values is a great improvement over using arbitrary out-of-range numeric values as a code for missing values with a special meaning.

```
1     → yes
0     → no
.     → regular missing: no response
888   → special missing: data were illegible
999   → special missing: attrition
```

Using these arbitrary numbers can be hazardous. If the only `x` values are `0` and `1`, then the mean of `x` is the percentage of `yes` values among the nonmissing values.

```
proc means; var x;
```

If there were `888` and `999` values in the data, they would have been included in the computation of the mean because they are valid numeric values. In order to prevent this from happening, such values can remain the same but be excluded from the computation, or be set to missing prior to the computation. Using special missing numeric values eliminates this hassle. The user is able to denote different types of numeric missing values for whatever purpose and is confident that all missing values are indeed treated as missing values.

```
1     → yes
0     → no
.     → regular missing: no response
.i    → special missing: data were illegible
.a    → special missing: attrition
```

DENOTING MISSING VALUES IN THE DATA

The MISSING statement in the DATA step is used to identify special characters (`_, A-Z`) that are to be treated as missing values when reading the numeric input data. In the following example, specifying `A` as a numeric value produces an error message since `x` is numeric and `a` is a nonnumeric value.

```
data sample;
input x;
cards;
a
;
```

By specifying `A` in the MISSING statement, the `A` is interpreted as a special numeric missing value (`.A` or `.a`) for all numeric variables in the DATA step. The special value with the period can also be specified in the data, but the period is not necessary. In referring to the special missing value in expressions or assignment statements, the period must precede `A`. The data set below will have no observations.

```
data sample;
missing a;
input x;
if x=.a then delete;
cards;
a
.a
;
```

Except for list input, a blank can be used to represent a numeric missing value. With list input, a period (.) is used to represent a missing character value.

```
data sample;
input x y$;
cards;
. .
;
```

Missing data values do not always need explicit values in list input. The INFILE statement's MISSOVER option can be used to set to missing the rest of the variables with no explicit values from the current input record. Once SAS reaches the end of an input record and does not see enough values for the variables in the INPUT statement, SAS does not go to the next input data record to continue reading for values.

```
data responses;
infile cards missover;
input item1-item5;
cards;
1 2
;
```

The result is:

```
Obs  item1  item2  item3  item4  item5
 1     1      2      .      .      .
```

READING MISSING VALUES IN DELIMITED DATA

Consider the following DATA step.

```
data error;
infile cards delimiter=',';
input school :$20. students teachers;
cards;
,253,32
Lyon Elementary,432,47
Webber School,566,65
;
```

There will be two instead of three observations in the data set. The log will have the following notes:

```
NOTE: Invalid data for TEACHERS in line 50 1-15.
RULE:----+----1----+----2----+----3----+----4----+----5----+----6
50   Lyon Elementary,432,47
SCHOOL=253 STUDENTS=32 TEACHERS=. _ERROR_=1 _N_=1
NOTE: SAS went to a new line when INPUT statement reached past
      the end of a line.
NOTE: The data set WORK.ERROR has 2 observations and 3
      variables.
```

The result is:

```
Obs  School         Students Teachers
 1   253                32        .
 2   Webber School     566       65
```

Use the INFILE statement's DSD option to correct this situation so that if there is no value between delimiters, it is treated as a missing value. When the DSD option and no DELIMITER= option is used, the delimiter defaults to a comma.

```
data stats;
infile cards dsd;
input school :$20. students teachers;
cards;
,253,32
Lyon Elementary,432,47
Webber School,566,65
;
```

The result is:

```
Obs  School           Students Teachers
 1                      253      32
 2   Lyon Elementary    432      47
 3   Webber School      566      65
```

PRINTING SPECIAL NUMERIC MISSING VALUES

In SAS output, special numeric missing values print in capital letters without the period. Consider the following DATA step.

```
data one;
input x @@;
y=1;
cards;
.i .a .
;

proc sort data=one; by x;
proc print data=one;
```

This data set is used for the remaining examples that refer to data set one.

The result is:

```
Obs   x   y
 1    .   1
 2    A   1
 3    I   1
```

A format can be applied to the variable x.

```
proc format;
value sample 1='yes'
             0='no'
             .='no response'
            .i='illegible'
            .a='attrition';

proc sort data=one; by x;
proc print data=one; format x sample.;
```

The result is:

```
Obs   x             y
 1    no response    1
 2    attrition      1
 3    illegible      1
```

PRINTING REGULAR NUMERIC MISSING VALUES

By default, a numeric missing value is printed as a single period. The MISSING= system option is used to change the display of a regular *numeric* missing value. The special numeric missing values are not affected.

```
options missing='*';
proc print data=one;
```

The result is:

```
Obs   x  y
 1    *  1
 2    A  1
 3    I  1
```

PRINTING CHARACTER MISSING VALUES

Character missing values do not print at all by default. A format can be used to print such values. Two consecutive quotes does not have the same effect as a blank enclosed in quotes.

```
proc format;
value $sample ' '='no response';

data two;
input x$;
cards;
.
;

proc print data=two;
```

The result is:

```
Obs   x             y
 1    no response    1
```

BY-GROUP PROCESSING WITH MISSING VALUES

With BY-group processing, SAS distinguishes between the different types of numeric missing values.

```
proc sort data=one; by x;
options missing='*';
proc means data=one;
  by x;
  var y;
```

The result is:

```
------------------------- x=* ------------------------

                The MEANS Procedure

               Analysis Variable : y

 N        Mean      Std Dev      Minimum      Maximum
 ---------------------------------------------------
 1   1.0000000            *    1.0000000    1.0000000
 ---------------------------------------------------

------------------------ x=A -------------------------

               Analysis Variable : y

 N        Mean      Std Dev      Minimum      Maximum
 ---------------------------------------------------
 1   1.0000000            *    1.0000000    1.0000000
 ---------------------------------------------------

------------------------ x=I -------------------------

               Analysis Variable : y

 N        Mean      Std Dev      Minimum      Maximum
 ---------------------------------------------------
 1   1.0000000            *    1.0000000    1.0000000
 ---------------------------------------------------
```

There is only one type of character missing value. Hence, such values are processed as a group in BY-group processing.

HOW PROCEDURES TREAT MISSING VALUES

In general, the default is missing values are excluded from calculations in procedures. There are various procedure options that can dictate how SAS deals with missing values. Consult the manual for a comprehensive discussion of missing values and a particular procedure. A few examples follow.

- PROC FREQ excludes missing values in the calculations by default. Percentages are based on nonmissing values.

```
proc freq data=one; tables x;
```

The result is:

```
                    The FREQ Procedure

                                    Cumulative    Cumulative
x    Frequency     Percent     Frequency      Percent
-----------------------------------------------------


                 Frequency Missing = 3
```

Use the /MISSING option in the TABLES statement to include missing values. Percentages are based on missing and nonmissing values.

```
proc freq data=one; tables x/missing;
```

The result is:

```
                    The FREQ Procedure

                                    Cumulative    Cumulative
x    Frequency     Percent     Frequency      Percent
-----------------------------------------------------
.           1       33.33             1        33.33
A           1       33.33             2        66.67
I           1       33.33             3       100.00
```

Use the /MISSPRINT option to exclude the missing values in the calculations while printing them in the table.

```
proc freq data=one; tables x/missprint;
```

The result is:

```
                    The FREQ Procedure

                                    Cumulative    Cumulative
x    Frequency     Percent     Frequency      Percent
-----------------------------------------------------
.           1           .             .           .
A           1           .             .           .
I           1           .             .           .

            Frequency Missing = 3
```

- PROC MEANS can only compute values using nonmissing values. It provides by default the number of observations used in the calculation. Specifying the NMISS option will list the number of missing values.

```
proc means data=one n nmiss sum; var x;
```

The result is:

```
                  The MEANS Procedure

                Analysis Variable : x

                       N
              N     Miss              Sum
              -------------------------
              0        3                .
              -------------------------
```

- PROC CORR computes correlations between variables using all pairs of nonmissing data for those variables (pairwise deletion of missing data). If the NOMISS option is used, PROC CORR is applied only to observations with nonmissing values for all the variables specified in the VAR statement (listwise deletion of missing data).

- PROC REG uses listwise deletion of missing data based on whether the variables in the MODEL or VAR statement have missing data.

WHEN VALUES ARE SET TO MISSING

Variable values are set to missing at the beginning of each DATA step iteration except for:

- variables in the RETAIN statement
  `retain x;`
- variables created by a SUM statement
  `x+y;`
- data in _TEMPORARY_ arrays
- variables created with options in FILE/INFILE statements
- variables created by the FGET function
- data elements initialized in an ARRAY statement
- automatic variables

After a variable acquires a missing value, its value can be changed during DATA step processing. However, missing values are propagated when missing values occur due to other missing values. Regardless of whether the numeric missing value(s) involved is/are regular or special, both types propagate as a regular missing value. In the example below, the value of `x` is missing if for a record, any one of the addends (`a`, `b`, or `c`) is missing.

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

In such cases, a SAS log message will indicate how many missing values were generated at a particular line in the program.

```
320  data two;
321  input a b c;
322  x=a+b+c;
323  list;
324  cards;

RULE:      ----+----1----+----2----+----3----+----
325        . . .
326        . 2 3
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).
      2 at 322:4
```

Missing values also result due to illegal operations or whenever there is an attempt to compute an undefined value, such as division by zero and the logarithm of zero. An overflow error (resulting number exceeds the largest number that can be represented by the system) will also result in a missing value.

Character values are converted to numeric ones when a character variable is an arithmetic expression. When the conversion fails (i.e., the character value does not resolve to a strictly numeric value), the result is set to missing.

If the SET, MERGE, or UPDATE statements are involved, the values are set to missing only before the DATA step's first iteration. If the BY statement is also involved, the values are set to missing whenever the BY group changes. The variable values are retained until there are new values.

FUNCTIONS

A number of functions are related to missing data:

- The MISSING function returns 1 if the argument resolves to a missing value and returns 0 otherwise.
```
if missing(x-y) then note='no value';
```

- The N function returns the number of nonmissing values. If the function returns a value that is equal to the number of arguments in the function, then there is no missing value among the record's N function arguments.
```
if n(of x y)=2 then note='none missing';
```

- The NMISS function returns the number of missing values.
```
if nmiss(x,y)=2 then note='both missing';
```

In the example below, the value of `x` is missing if for a record, any one of the addends (`a`, `b`, or `c`) is missing.

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

If the SUM function was used, the value of `x` will be the sum of nonmissing addends (`a`, `b`, or `c`) for the record. Missing values are skipped/ignored.

```
x=sum(of a b c);
x=sum(a, b, c);
```

Like the SUM function, some other functions exclude missing values.

STATISTICAL ANALYSIS AND MISSING DATA

Some approaches for dealing with missing data are enumerated below (Smith, 2002). Each method has its advantages and disadvantages. The literature regarding these methods is rich and continues to grow.

1. Ignoring the missing data (listwise/case deletion): By default SAS excludes observations with variables that have missing values if any of the variables are to be used for the analysis. This leads to reduced sample size, which results in less power for statistical tests.

2. Single imputation (replacing the missing value with a nonmissing one)

    - *Arbitrary methods.* Examples of arbitrary methods include using the sample mean and the last-observation-carried-forward method. The missing values can be substituted with the mean using PROC STANDARD, which can also be used to standardize data set values to a specified mean and standard deviation.

```
proc standard data=original
              out=imputed replace;
var x;
```

    If the MEAN= option is used, the specified mean value will replace missing values. In the example below, the missing values are replaced by 30.

```
proc standard data=original
              out=imputed replace
              mean=30;
 var x;
```

    - *Regression-based imputation*. A regression model is used for data where the dependent variable has some missing values. The estimated regression coefficients are used to predict/impute the missing values.

    - *Hot deck imputation*. Observations are divided into groups with similar characteristics. For example, observations are grouped by grade and age. A 6-yr old 1st grader with nonmissing values is randomly selected from the subgroup. The missing values for a 6-year-old 1st grader is then replaced by the nonmissing values from the randomly selected subject.

3. *Model the missingness*. In certain situations, a missing value can be represented by a binary variable on whether the observation has a value or not.

4. *Multiple imputation*. A data set with missing values is imputed and then analyzed a number of times. The results from the repeated analyses are then pooled. (PROC MI and PROC MIANALYZE provide methods for multiple imputation.)

REFERENCES

SAS Institute Inc., *SAS® Language Reference, Version 8*, Cary, NC: SAS Institute Inc., 1999. 1256 pp.

SAS Institute Inc., *SAS OnLineDoc®, Version 8*, Cary, NC: SAS Institute Inc., 1999.

Smith, M. (2002). *What can I do about missing data?* Retrieved March 6, 2003, from the Health Economics Resource Center Web Site: http://www.herc.research.med.va.gov/faq_19.htm

TRADEMARK NOTICE

CONTACT INFORMATION

Imelda C. Go (icgo@juno.com)
South Carolina Department of Education
Columbia, SC