

Martin Barron, Abt Associates Inc., Chicago, IL

By necessity, even beginning SAS users are familiar with the use of missing values in SAS. However, even more advanced users may be unfamiliar with the variety of ways in which special missing values can be assigned and used with SAS. This presentation explores the "mysteries of missings" and provides tips for working with missing values in common SAS tasks. Our overall focus will be on special missing values and their uses in project management. We will also explore the basics of missing values, SAS's handling of missing values in updates, and SAS's handling of missing values in a few example procedures.

Missing data are a fact of life for virtually all data collectors and data users. As such, they are an integral part of the SAS system and must be dealt with in virtually every SAS program. As one would expect, therefore, SAS contains many powerful ways of dealing with missing values. This paper explores the basics of missing values in the SAS system, as well as some of their more advanced features. We pay special attention to the many ways values come to be considered "missing" and the importance of this data from a project management perspective (especially survey management).

Although in each of the examples cited above the data gathered from the respondent is the same (i.e., none or missing), these different types of missing values have significantly different meanings to a data collection manager. A question with an extremely high number of missing values caused by refusals might indicate the question needs to be reworded, while a question with a large number of missing values caused by a particular skip pattern might be perfectly acceptable. Further, while aggregating missing values *may* be of no consequence to a data analyst, interesting analytic data can be lost in the process. For example, when a respondent is asked the question "Has your child been vaccinated for polio?" and answers "I don't know", the substantive interpretation of that response can be just as interesting as yes/no responses to that question, and is certainly interpreted differently than responses that were missing because the respondent had no children.

Missing values are denoted differently, depending on whether you

In example2, two separate statements are needed to find a regular numeric missing (.) and a special missing (A). But suppose you had a variable that used multiple special missing values and you wanted to determine if any type of missing was present. To accomplish this, you can employ the MISSING function.

For example, suppose we had a dataset containing five cases and the numeric variable VAR1. Four of the cases have missing values for VAR1, but three of these cases are set to *different* special missing values.

TABLE 1: INITIAL EXAMPLE3 DATASET

case	VAR1
1	1
2	.
3	X
4	Y
5	Z

You could identify these missing values using four if ... then statement (as long as you knew exactly which special missing values were assigned). Or you could use one missing statement. When the example code below is used with the dataset above, VAR2 resolves to 1 for cases 2-5.

```
Data example3;
  set example3;
  if missing(var1) then var2=1;
Run;
```

Note that this function works for character data as well as numeric and special missing values.

RESOLVING SIMPLE ARITHMETIC AND LOGIC WITH MISSING VALUES

The result of any mathematical operation on a numeric missing value is a missing. This is equally true of numeric missing, special missing, and (of course) character missing values. In the following example, VAR1-VAR3 all resolve to regular numeric missing (the character missing value used to calculate VAR3 is converted to numeric data before SAS attempts to resolve the expression).

```
data example4;
  a = 1;
  b = .;
  c = .a;
  d = " ";
  var1=a + b;
  var2=a - c;
  var3=a*d;
run;
```

One might expect that, just as 1 + missing is undefinable, so to would 1>missing. In fact, this is not the case. SAS defines missing values as "smaller" than regular numbers and treats them thusly in logical expressions. Therefore in the following code VAR3 is set to 1.

```
Data example5;
  var1=1;
  var2=.;
  if var1>var2 then var3=1;
Run;
```

In fact, there is even an order to the "size" of different types of missing values. We discuss this below in our discussion of sorting.

SAS FUNCTIONS AND MISSING VALUES

Unlike simple arithmetic, most SAS functions easily handle missing values. In the following example, VAR1 correctly resolves to 1.5, VAR2 correctly resolves to 2, and VAR3 correctly resolves to 3.

```
Data example6;
  A = 1;
  B = 2;
  C = .;
  D = .a;
  var1= MEAN(a,b,c,d);
  var2= MAX(a,b,c,d);
  var3= SUM(a,b,c,d);
Run;
```

THE MISSING OPTION

By default, SAS displays a period (.) to denote numeric missing values. You can change this by setting the system option MISSING to a new value. You can only use one character to denote numeric missing values, and this character must be confined in single or double quotes in the MISSING option. For example, the code below tells SAS to display all numeric missing values as an asterisk (*).

```
options missing="*";
```

Keep in mind that it is possible to set the default display of numeric missing values to the letters A-Z and underscore (_) even though these characters are used to denote special missing values (as discussed below). This *does not* change the value of the numeric missing value to that of the corresponding special missing value, but they will appear the same in output. I recommend, therefore, that you avoid setting the default missing value to display as characters A-Z or underscore.

THE MISSING STATEMENT

The missing statement is used to define special missing values when reading raw data into SAS. When a numeric variable contains character data, a missing statement can be used to assign those values to a corresponding special missing value, as in the code below.

```
Data example7;
  missing X Y;
  input var1 var2 var3;
  datalines;
  1001 X 55
  1002 33 Y
  ;
Run;
```

In the example above, the raw data contains two character values in two different numeric variable fields. By using the MISSING statement, we instruct SAS to assign the special missing value .X to VAR2 for the first case and .Y for VAR3 for the second case.

USING MISSINGS & SPECIAL MISSINGS

We now turn to a discussion of how missing values and special missing values are used and are treated by several common procedures. This by no means exhausts the idiosyncrasies of missing values in procedures. Instead, the discussion below is meant to demonstrate to the reader that missing values are often treated in non-intuitive ways, and therefore should be handled with care, whatever the procedure being used.

SORTING

When sorting data, missing values are handled separately from non-missing data. In character data, missing are considered "smaller" than any printable character. Therefore, they appear before other observations (when sorted in the default ascending order). However, a special case may occur when a variable

contains ASCII data that is typically unprintable (for example, machine carriage-control characters). Some unprintable characters have values less than character missing values (blanks). Care should be given then, when sorting character data containing unprintable character data.

For numeric data, the situation is slightly more straightforward. Missing data is always considered "smaller" than numeric data. In addition, a set sort order is set for the different types of numeric and special missing values. When numeric data contains different types of missing values, underscore (.) missing values are considered the smallest, followed by default (.) missing values (i.e., traditional missing values), followed by .A - .Z missing values (with .A considered the smallest, .Z the largest). The sort order for numeric variables is summarized in table 2 below.

TABLE 2: 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

FREQUENCY

Frequencies are perhaps the most common, and the most basic, of all SAS procedures. In a default frequency table, missing values are not displayed. Instead, the total number of missing values is listed below the created table or tables, as in the example below.

OUTPUT 1: TYPICAL FREQUENCY OUTPUT

The FREQ Procedure				
Var1	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	1	50.00	1	50.00
2	1	50.00	2	100.00
Frequency Missing = 2				

In this instance, SAS treats character, numeric, and special missing values the same. In the output above, for example, the frequency missing contains one "regular" numeric missing value and one .X missing value.

If you prefer, you can have SAS treat missing values as regular data for the purposes of the frequency table, as in the code below.

```
Proc freq data=example8;
  tables var1 /missing;
Run;
```

This produces output as shown in the example below.

OUTPUT 2: FREQUENCY OUTPUT WITH MISSING OPTION

The FREQ Procedure				
Var1	Frequency	Percent	Cumulative Frequency	Cumulative Percent
.	1	25.00	1	25.00
X	1	25.00	2	50.00
1	1	25.00	3	75.00
2	1	25.00	4	100.00

Notice that different types of missing values are listed in the same order noted above in our discussion of the sort procedure.

It is important to show care when using the /missing option and computing a two-way or larger table. In the case of a two-way table, missing variables will appear in both row and columns. While this may be appropriate for your analysis, it can produce unintended results when calculating some statistics. Chi-square, for example, will treat rows and columns containing missing data as ordinary data, therefore increasing your degree of freedom and changing your computed statistic.

UPDATE

Although not nearly as common as the Freq and Sort procedures, the Update statement of the data step treats missing values in some unusual ways and is, therefore, worth mentioning here.

The update statement allows you revise a master dataset by replacing values with values in a transaction dataset. By default, missing values in the transaction database *do not* overwrite existing data but special missing values *do* overwrite existing data. This is illustrated in the code below.

```
Data master;
  id=1;
  var1=1;
  var2=.;
  var3=3;
Run;

Data transaction;
  id=1;
  var1=.;
  var2=2;
  var3=.N;
Run;

Data example9;
  update master transaction;
  by id;
Run;
```

The master, transaction, and resulting datasets are summarized in table 3 below

TABLE 3: UPDATE DATASETS

	Master dataset	Transaction dataset	Example9 dataset
VAR1	1	.	1
VAR2	.	2	2
VAR3	3	.N	.N

VAR1 resolves to 1 since that is its value in the master database and it cannot be overwritten with the missing value from the transaction dataset. VAR2 resolves to 2 because it was set to missing in the master dataset and was therefore overwritten with the 2 in the transaction dataset. VAR3 resolves to .N because the value in the master dataset (3) is overwritten by the special missing value in the transaction dataset.

You can change the behavior of numeric missing values in the update statement by using the UPDATEMODE option. By setting UPDAGTEMODE=NOMISSINGCHECK, a missing value in the transaction dataset will overwrite other values in the master dataset (and special missing values will continue to overwrite master dataset values). By adding this option to our example from above:

```

Data example10;
  update master transaction;
  updateoption=nomissingcheck;
  by id;
Run;

```

The resulting dataset has the values summarized in table 4 below.

TABLE 4: UPDATE DATASETS (WITH NO MISSING CHECK)

	Master dataset	Transaction dataset	Example9 dataset
VAR1	1	.	.
VAR2	.	2	2
VAR3	3	.N	.N

VAR1 now has a value of missing because the value of 1 in the master dataset is overwritten by the missing in the transaction dataset. The other two variable resolve the same way as previously; VAR2 resolves to 2 because the value in the transaction dataset overwrite the missing in the master dataset while the .N in VAR3 of the transaction dataset overwrites the value of 3 in the master dataset.

CONCLUSION

In this paper we have reviewed SAS missing values. As should be clear from our discussion above, missing values are far more than simply the absence of data. They are an integral, and sometimes messy, part of the SAS system. However, with due care, they can provide a rich managerial and analytic tool.

REFERENCES

1. SAS Institute Inc., *SAS Language: Reference, Version 6, First Edition*, Cary, NC: SAS Institute Inc., 1990.
2. SAS Institute Inc., *SAS OnlineDoc, Version 8*, Cary, NC: SAS Institute Inc., 1999.

CONTACT INFORMATION

Your comments and questions are valued and encouraged.
Contact the author at:

Martin Barron
Abt Associates Inc.
640 N LaSalle Suite 400
Chicago, IL 60610-3781

Work Phone: 312-867-4031
Fax: 312-867-4200
Email: Martin_Barron@abtassoc.com