Those Missing Values in Questionnaires

John R. Gerlach, Maxim Group, Plymouth Meeting, PA
Cindy Garra, IMS HEALTH; Plymouth Meeting, PA

Abstract

Questionnaires are notorious for containing responses that later become missing values during processing and analyses. Contrary to a non-response that results in a typical, bona fide, missing value, questions might allow several alternative responses, necessarily outside the usual range of appropriate responses. Whether a question represents continuous or categorical data, a good questionnaire offers meaningful alternatives, such as: "Refused to Answer" and, of course, the quintessential "Don't Know." Traditionally, these alternative responses have numeric values such as 97, 998, or 9999 and, therefore, pose problems when trying to distinguish them from normal responses, especially when multiple missing values exist. This paper discusses missing values in SAS and techniques that facilitate the process of dealing with multi-valued, meaningful missing values often found in questionnaires.

Introduction

Questionnaires pose many challenges for the data analyst, especially when processing missing data that have meaning. Initially, these so-called missing values are not true SAS missing values. Rather, such variables contain actual values; however, these values are outside the normal range, by offering the respondent an alternative response. Moreover, even though these quasi-missing values contain important information, they are usually excluded from the primary analysis.

Keep in mind that the problem posed here has nothing to do with imputing missing values from those instances when the respondent did not answer one of the questions. This paper does not discuss, for example, hot deck imputation, which attempts to assign a reasonable response in the event of a non-response. Rather, this paper addresses the problem of dealing with actual values that, at times, must be ignored and, other times, might be included in the analysis.

Consider a categorical (numeric) variable that contains the values 1 through 5 and includes possible missing, alternative, responses, valued as: 97, 98, and 99. Or, consider another variable denoting income that includes two other responses, coded as: 9998 and 9999. Typically, the analyst does not include such values for most of the analysis. Unfortunately, however, these alternative values are not bona fide SAS missing values; consequently, a SAS procedure, expectedly, will include these alternative values, thereby producing bogus results.

Typically, you can re-code the original data, so that the missing values become ordinary missing values, thus allowing SAS to process only appropriate values. Of course, there's a loss of information since the original missing values did not constitute a non-response. Also, such pre-processing might include a myriad of IF THEN / ELSE statements, which can be very tedious and time-consuming to write, and difficult to maintain. Thus, given a questionnaire that has over a hundred variables with varying levels of missing values, the task of re-coding these variables becomes very time consuming at best. Even worse, the permanent re-coding of alternative responses to ordinary missing numeric values in SAS precludes categorical analysis that requires the inclusion of these alternative responses.

Certainly, there are efficient methods to deal with alternative responses, those 'other' missing values, that would facilitate the analysis and ensure its integrity, without losing information.

Missing Values

Missing values are commonplace in the real world of data analysis. Not surprisingly, SAS affords the means to designate missing values for character and numeric data. There are two well-known notations for missing values in SAS: the blank character that represents character missing values and the single decimal point that represents numeric missing values.

The SAS System also recognizes as many as 27 special missing values for numeric variables, comprised of the underscore character and the letters of the alphabet, each preceded by the single decimal point (ie, ._, .A -.Z). You can use either upper or lowercase letters to represent special missing values. Also, when printing a special missing value, the SAS System prints only the underscore or the letter, without the period. However, when specifying a special missing value, you must precede it with the period; otherwise, SAS treats it as a variable name. Ironically, the unknown variable would be assigned an ordinary missing value.

The order of numeric values in SAS, including missing values and non-missing values is well defined and listed below, from the smallest to largest value.

_	Underscore
.	Period
A-Z	A (smallest) through Z (largest)
-n	Negative numbers
0	Zero
+n	Positive numbers

Even though the blank character denotes a missing value for character data, keep in mind that, by definition, missing values for character data are smaller than any printable character value. Thus, in the event that you have non-printable characters, such as carriage control data, it's possible that missing values may not appear first in a sorted data set.

The MISSING Statement

In regards to analyzing questionnaire data, the problem of dealing with missing values begins with the coding scheme used during data entry. The traditional scheme of using values outside the normal range of responses becomes a nagging problem for the analyst when processing the data. Alternatively, if you assign letters denoting special missing values, at the point of data entry, rather than actual values, you can eliminate this problem. Then, during the analysis, you can use a format to discern these values accordingly.

The MISSING statement declares that certain characters in the raw data represent missing values for numeric data. For example, SAS reads data so that coded values R (Refused to Answer) and X (Invalid Response), become special missing values. Consider the following Data step that reads numeric data allowing for two special missing values, along with the ordinary missing value and non-missing values. Notice that you can specify special missing values (as data, not as a constant) with or without the preceding period.

```
data survey1;
   missing r x;
   input id quest1-quest5;
cards;
1001  3  R  4  2  X
1002  1  3  .  R  4
1003  X  X  R  3  1
1004  .  3  4  5  1
   < More data >
;
```

The MISSING= system option differs from the MISSING statement used in the Data Step. This option specifies a character to be printed when numeric variables contain ordinary missing values. Thus, using this option will not alter special missing values to a single specified character. Of course, the default character for ordinary missing values is the period (.).

The Unfortunate Premise

Now, consider the situation when you have a SAS data set containing non-missing numeric values that represent alternative responses outside the normal range of values. And, as with any survey, you have non-responses that later become bona fide (ordinary) missing values in SAS . The Data step below creates a data set that typifies this unfortunate premise of dealing with such data. Also, notice that question #3 (q3) contains both categorical and continuous values.

```
data survey;
   input id q1 q2 q3;
cards;
1001   1   98    10.1
1002   2    .   998
1003  97    3    14.3
1004   .   99      .
1005   5    1   999
1006   2    4    13.8
1007  99   97    12.2
1008   3    5   999
;

proc print noobs;
   title1 'Original Data';
run;
```

```
Original Data

 ID       Q1        Q2         Q3

1001       1        98        10.1
1002       2         .       998.0
1003      97         3        14.3
1004       .        99          .
1005       5         1       999.0
1006       2         4        13.8
1007      99        97        12.2
1008       3         5       999.0
```

The output above clearly illustrates that you must contend at times with alternative values prior to doing the analysis, while other times you might prefer those values as part of the analysis.

A Poor Solution

Using IF statements to convert alternative responses offers only more work for the analyst / programmer. Indeed, this tedious, error-prone technique only compounds the problem of preparing the data for analysis. For instance, consider the following code that reassigns alternative response to special missing values.

```
data survey;
   set raw;
   if q1 eq 97 then q1 = .R;
   if q1 eq 98 then q1 =  X;
run;
```

There's a problem with the second IF statement. The special missing value X is missing (pun intended) a period; therefore, SAS assigns the variable X an ordinary missing value, which disrupts the re-coding process. Even worse, if the variable X exists in the input data set or, perhaps, it so happens to be a loop control variable, the re-coding ruins the integrity of the data, completely. A devastating typographic error, and not a good idea.

The WHERE statement offers a quick and dirty solution that might accommodate an ad hoc analysis, simply by discarding observations having inappropriate values, meaning, those alternative responses that the SAS System does not recognize as missing. Unfortunately, since the WHERE statement selects observations, it's possible that an observation might have an appropriate response in one variable and not in the other. Thus, this technique fails to select all the observations needed for the analysis when selecting on several variables. Hence, this technique would suffice only for ad hoc analysis using single variable, as shown below.

```
proc means min max;
   var q1;
   where q1 le 5;
   title1 'Minimum Maximum Values';
run;
```

User-defined Formats

User-defined formats offer a better way to report data by creating categories, even for continuous data. The formats **sect1f** and **sect2f** might serve sections of a large questionnaire that contain similar responses. Notice that the format **sect1f** lists all possible values: normal and alternative responses, along with non-responses. In contrast, the format **sect2f** uses an existing, SAS-supplied format to accommodate, perhaps, a question denoting a continuous numeric. Thus, in which case, the format uses the existing format (best5.) for normal values and formats the alternative and non-response values accordingly.

```
proc format;
   value sect1f
           1 = "Rarely"
           2 = "Occasionally"
           3 = "Regularly"
           4 = "Often"
           5 = "Very Often"
          97 = "Refused"
          98 = "Don't Know"
          99 = "Not Ascertained"
       other = "Non-Response";
   value sect2f
        1-20 = [best5.]
         998 = "Unavailable"
         999 = "Declined"
       other = "Non-response";
run;

proc print noobs;
   var q1 q3;
   format q1 sect1f. q3 sect2f.;
   title1 'Questions #1 and #3';
   title2 '( With Format )';
run;
```

Given the data considered previously, along with the formats **sect1f** and **sect2f**, the listing below illustrates the advantage of using formats. That is, no Data step was required to re-code the data. Notice how7 the report lists the continuous values found in question #3 (q3) while specifying the appropriate text for other responses.

```
Questions #1 and #3
( With Format )

Q1                Q3

Rarely             10.1
Occasionally      Unavailable
Refused            14.3
Non-Response      Non-response
Very Often        Declined
Occasionally       13.8
Not Ascertained    12.2
Regularly         Declined
```

Analyzing the same data, the FREQ procedure produces two distributions, once again, using the formats to perform the re-coding.

```
proc freq;
   tables q1 q3 / missing nocum;
   format q1 sect1f. q3 sect2f.;
```

```
   title1 'Distribution of Questions #1 and #3';
   title2 '( With Format )';
run;
```

Notice that all possible values are represented including, unfortunately, *every* unique value belonging to question #3, which may not be desirable. By the way, the FREQ procedure orders the distribution of a variable by its internal value, for which the Non-response (missing) value represents the smallest value. You can change this default by using the ORDER= option of the FREQ statement.

```
Distribution of Questions #1 and #3
( With Format )

             Q1   Frequency   Percent
ffffffffffffffffffffffffffffffffff
Non-Response              1      12.5
Rarely                    1      12.5
Occasionally              2      25.0
Regularly                 1      12.5
Very Often                1      12.5
Refused                   1      12.5
Not Ascertained           1      12.5


          Q3   Frequency   Percent
ffffffffffffffffffffffffffffffffff
Non-response           1      12.5
 10.1                  1      12.5
 12.2                  1      12.5
 13.8                  1      12.5
 14.3                  1      12.5
Unavailable            1      12.5
Declined               2      25.0
```

Rather than indicate the myriad responses, found in Question #3, you can modify the **sect2f** format so that these responses indicate a single category, as follows.

```
proc format;
   value sect2f
      1-20 = "Normal Response"
       998 = "Unavailable"
       999 = "Declined"
     other = "Non-response";
run;
```

```
Distribution of Questions #3 (Modified)

             Q3   Frequency   Percent
ffffffffffffffffffffffffffffffffff
Non-response              1      12.5
Normal Response             4      50.0
Unavailable               1      12.5
Declined                  2      25.0
```

Now, consider what happens when you use the MEANS procedure to produce simple descriptive statistics. Again, keep in mind that the issue concerns alternative responses that must be ignored, like missing data, during the analysis. In determining the range of normal values, the results are incorrect because of the extraneous values representing alternative responses.

```
proc means min max maxdec=1;
   var q1 q2 q3;
   format q1 q2 sect1f. q3 sect2f.;
   title1 'Minimum / Maximum Values';
run;
```

```
Minimum / Maximum Values

Variable       Minimum       Maximum
-----------------------------------
Q1                 1.0          99.0
Q2                 1.0          99.0
Q3                10.1         999.0
-----------------------------------
```

Dynamic Formats

Finally, let's consider the task of actually re-coding the data into special missing values so that the SAS System can interpret the alternative responses as missing values or as relevant values, that is, different from ordinary missing values.

SAS 6.07 introduced an enhanced version of the INPUT and PUT functions that use formats, dynamically, at run time. These four functions: INPUTN, INPUTC, PUTN, and PUTC; allow the use of dynamic formats that determine the intended format required for converting an internal value to its formatted value. In this case, the INPUTN function serves the purpose of converting alternative responses to bona fide special missing values, as needed. With a suitable format library, the INPUTN function accomplishes this task easily. That is, this function uses the appropriate format, which is

determined by the variable it's processing using the formats below.

```
proc format;
   invalue nv2f
       97 = .R   98 = .D   99 = .N;
   invalue nv3f
      997 = .A  998 = .B  999 = .M;
   value $nvarsf
      'Q1' = 'nv2f.'
      'Q2' = 'nv2f.'
      'Q3' = 'nv3f.';
run;
```

The following Data step does the re-coding. Notice the VNAME call routine that identifies the variable (e.g., q1) which determines the format used for re-coding. That is, the format **$nvarsf** associates each variable to its appropriate format based on the variable identifier. Then, the INPUTN function uses that specific format at run time, thereby accomplishing the task.

```
data;
   length nvar $8;
   set testds;
   array nvars{*} q1-q3;
   do i = 1 to dim(nvars);
      call vname(nvars{i}, nvar);
      nvfmt = put(upcase(trim(nvar)),$nvarsf.);
      nvars{i} = inputn(
         put(nvars{i},best12.),nvfmt);
      end;
run;
```

The following PRINT procedure lists the effect of the re-coding scheme, albeit showing the internal values only. Observe the difference between the ordinary missing value (.) and the special missing values (.D, .N). Also, notice that the procedure does not show the preceding period for special missing values.

```
proc print;
   title1 'Revised Data (Non-formatted)';
run;
```

Revised Data (Non-formatted)

ID	Q1	Q2	Q3
1001	1	D	10.1
1002	2	.	B
1003	R	3	14.3
1004	.	N	.
1005	5	1	M
1006	2	4	13.8
1007	N	R	12.2
1008	3	5	M

Preferred Output

Now let's add two new formats that will convert those special missing values, as follows.

```
proc format;
   value othr1f
      0-high = [best15.]
          .R = "Refused"
          .D = "Don't Know"
          .N = "Not Ascertained"
       other = "Non-Response";

   value othr2f
      0-high = "Normal"
          .A = "Not Available"
          .B = "Bad Conditions"
          .M = "Not Measurable"
       other = "Non-Response";
run;
```

Using the formats **othr1f** and **othr2f** defined above, the following PRINT procedure produces a much better report discerning even those alternative responses.

```
proc print noobs;
   var id q1 q3;
   format q1 othr1f. q3 othr2f.;
   title1 'Revised Data (Formatted)';
run;
```

Revised Data (With Format)

ID	Q1	Q3
1001	1	Normal
1002	2	Bad Conditions
1003	Refused	Normal
1004	Non-response	Non-Response
1005	5	Not Measurable
1006	2	Normal
1007	Not Ascertained	Normal
1008	3	Not Measurable

The following MEANS procedure lists minimum and maximum values of the several questions. Notice this time that the results are correct, not being influenced by the alternative responses, since these values are missing and, therefore, extraneous for this analysis.

```
proc means min max maxdec=1;
   var q1--q3;
   title 'Min / Max Values';
run;
```

```
Min / Max Values

Variable       Minimum       Maximum
-----------------------------------
Q1                 1.0           5.0
Q2                 1.0           5.0
Q3                10.1          14.3
-----------------------------------
```

Finally, the FREQ procedure shows several distributions illustrating the possibilities of formatting special missing values that have similar importance to so-called normal range values.

```
proc freq;
   tables q1--q3 / missing nocum;
   format q1 q2 othr1f. q3 othr2f.;
   title 'Distribution of Questions 1,2,3';
run;
```

```
Distribution of Numeric Values

             Q1   Frequency   Percent
ffffffffffffffffffffffffffffffffff
Non-response             1      12.5
Not Ascertained          1      12.5
Refused                  1      12.5
           1             1      12.5
           2             2      25.0
           3             1      12.5
           5             1      12.5

             Q2   Frequency   Percent
ffffffffffffffffffffffffffffffffff
Don't Know               1      12.5
Non-response             1      12.5
Not Ascertained          1      12.5
Refused                  1      12.5
           1             1      12.5
           3             1      12.5
           4             1      12.5
           5             1      12.5

             Q3   Frequency   Percent
ffffffffffffffffffffffffffffffffff
Bad Conditions           1      12.5
Non-Response             1      12.5
Not Measurable           2      25.0
Normal                   4      50.0
```

Conclusion

Questionnaires often generate lots of "Other" responses that require special handling depending on the immediate analysis. These alternative responses, at times, must be treated like missing data. But, on the other hand, it's not wise to convert these values into ordinary missing values, straight away, thereby losing information.

Understanding missing values in the SAS System and using dynamic formats greatly facilitate the process of converting alternative responses into special missing values. Such features as special missing values along with the INPUTN function that uses formats during run time greatly facilitates the process of converting the original values into distinguishable, special missing values without too much effort.

Author Information

John R. Gerlach
Maxim Group
Plymouth Meeting, PA

Cindy Garra
IMS HEALTH
Plymouth Meeting, PA