Yes, No, Maybe So: Tips and Tricks for Using 0/1 Binary Variables

Laurie Hamilton, Healthcare Management Solutions LLC, Columbia MD

ABSTRACT

Many SAS® programmers are familiar with the use of 0/1 binary variables in various statistical procedures. But 0/1 variables are also useful in basic database construction, data profiling and QC techniques.

By definition, a binary variable is a flavor of categorical variable, an outcome or response measure with only two possible values. In this paper, we will use a sample dataset composed of 0/1 numeric binary variables to demonstrate some tricks and tips for quick Data Profiling and Quality Assurance using basic SAS functions and the MEANS and FREQ procedures.

INTRODUCTION

Binary variables are a type of categorical variable, specifically those variables which can have only a Yes or No value. We see these types of variables often in Questionnaire type data, which is the example we will use in this paper.

We will begin with a brief discussion of the options constructing 0/1 variables, including selection of data type and the implications for the coding of missing values. We will then look at some tips and tricks for using the SAS® functions SUM, NMISS and CAT to create summary variables from multiple binary variables across individual observations and also demonstrate one method for constructing a Pattern variable which combines all the information in multiple binary variables into one character string.

The SAS® procedures PROC MEANS and PROC FREQ are ideally suited to quickly profiling data composed of 0/1 numeric binary variables and we will explore some applications of those procedures using our sample data. Finally, we will demonstrate a pattern-recognition application based on the composite Pattern variable constructed from individual binary data elements.

WHY USE 0/1 INSTEAD OF Y/N

You may be asking “What is problematic about coding a Yes/No variable as Yes = ‘Y’ and No = ‘N’?”

The alpha characters ‘Y’ and ‘N’ are certainly short and easy to understand values in this context. And as a character data type, they take up less disk and memory storage than a similar one-digit numeric value. But character variables containing alphabetic values are notoriously susceptible to mistakes in value assignment (remember those ‘y’ and ‘n’ values when the database codebook indicated all ‘Y’s and ‘N’s?).

Limiting the valid values to 0 or 1 in a numeric field sidesteps some of the problems associated with ‘Y’/’N’ coding such as that described above.

VALUES TO REPRESENT “MAYBE SO” (MISSING) RESPONSES

The value or values chosen to represent a response of ‘Missing’ for a binary variable has implications for analysis of the data down the road.

Many database applications pad missing data with ‘9’s to clearly indicate that the variable was considered and did not have a value. Just as with ‘Y’/’N’ values, the use of ‘9’ for designating a ‘Missing’ value is clear and a well-known data convention.

However, padding “Missing” 0/1 binary fields with any value can interfere with some of our tricks for taking advantage of the 0’s and 1’s to have SAS mathematical functions and descriptive procedures quickly profile our data.

So for the purposes of this paper, we will represent “Maybe So” (“Missing”) responses with the SAS Numeric Missing value of ‘.’.

EXAMPLE DATA

THE BINARY VARIABLES

Let's work with a set eight responses to a six-item questionnaire allowing only Yes or No responses. We'll implement our decision to code the responses using numeric 0's and 1's and the SAS missing for items with no response. Notice that in some cases we do, in fact, have missing answers to certain questions.

Table 1. Example Data

test_site	subject	Q1	Q2	Q3	Q4	Q5	Q6
Site A	SubjA	.	1	1	1	1	0
Site A	SubjB	0	0	1	0	1	0
Site B	SubjC	1	1	0	1	0	0
Site A	SubjD	0	0	.	.	0	1
Site A	SubjE	1	1	1	1	1	1
Site B	SubjF	0	0	1	0	1	0
Site B	SubjG	1	1	1	1	1	1
Site A	SubjH	0	0	.	.	0	1

ENHANCING THE DATA WITH RESPONSE-TYPE TOTALS

The first thing we might like to do is profile each subject's set of questionnaire responses. This will give us an overall look at how clean the data are for Quality Assurance purposes, and also allow us to create some initial summary reports.

To accomplish this, we'll add three variables: Total Number of 'Yes' responses, Total Number of 'No' responses and Total Number of 'Missing' responses for each subject.

```
data example2;
  set example1;

  /*Sum the 1's to get a Count of 'Yes' Responses */
  tot_yes = sum(of q1-q6);

  /*Count the Missing Responses */
  tot_miss = nmiss(of q1-q6);

  /* Use the 'Yes' and 'Missing' Counts to back calculate the 'No's */
  tot_no = 6 - sum(of tot_yes,tot_miss);
run;
```

First we use the SAS® SUM function to get a count of 'Yes' responses, it's easy – just sum up the 1's across each record. Next we'd like to know how many responses were missing on each questionnaire. Easy again, invoke the SAS® NMISS function to count them up. And now we can use the counts of 'Yes' and 'Missing' responses to back-calculate the number of 'No's.

Our data now contain information on the total number of Yes/No/Maybe So (missing) responses:

Table 2. Example Data After Adding Response-Type Totals

test_site	subject	Q1	Q2	Q3	Q4	Q5	Q6	tot_yes	tot_miss	tot_no
Site A	SubjA	.	1	1	1	1	0	4	1	1
Site A	SubjB	0	0	1	0	1	0	2	0	4
Site B	SubjC	1	1	0	1	0	0	3	0	3
Site A	SubjD	0	0	.	.	0	1	1	2	3
Site A	SubjE	1	1	1	1	1	1	6	0	0
Site B	SubjF	0	0	1	0	1	0	2	0	4
Site B	SubjG	1	1	1	1	1	1	6	0	0
Site A	SubjH	0	0	.	.	0	1	1	2	3

PROFILING INDIVIDUAL OBSERVATIONS

Now we can use the response-type totals to profile the data. PROC FREQ will give us a quick summary snapshot useful for Quality checking and reporting on data completeness. We could produce frequency tables for each of the response-type total variables separately, but let's combine the information and create a cross-tabulation of all three:

```
proc freq data = example2;
  tables tot_yes * tot_no * tot_miss/list missing;
run;
```

Table 3. Individual Profiles Using Response-Type Totals

tot_yes	tot_no	tot_miss	Frequency	Percent
1	**3**	**2**	2	25
2	**4**	**0**	2	25
3	**3**	**0**	1	12.5
4	**1**	**1**	1	12.5
6	**0**	**0**	2	25

The cross-tabulation above allows us to quickly answer general questions both about the data quality and the type of responses collected. Each row is a particular pattern of number of responses in each of the three categories – 'Yes', 'No' and 'Missing.' From the cross tab we can see, for example, that 3 of the 8 responses contain at least one 'Missing' response and that two subjects answered 'Yes' to at least four of the six items.

If we'd like various flavors of this report, we could use PROC FORMAT with PROC FREQ to further categorize the response-type totals. Here we create a format 'Happy' to group subjects based on the number of 'Yes' responses, and another format 'Incomplete' to create similar groupings for 'Missing' responses by questionnaire:

Table 4. Individual Profiles Using PROC Format Categorization

FORMATS

```
proc format;
         value happy
         0-2  = 'Not Happy'
         3-4  = 'Somewhat Happy'
         5-6  = 'Very Happy'
;
         value incomplete
         0     = 'Complete'
         other = 'Incomplete';
   run;
```

PROC FREQ CODE

```
proc freq data = example2;
     tables tot_yes/list missing;
     tables tot_miss/list missing;
     format tot_yes  happy.
            tot_miss incomplete.
;
run;
```

The results are similar to those obtained from the unformatted frequency distributions, but more customized to the specific questions. Note that the Happy format makes the assumption that 'Yes' on this questionnaire is a positive response.

Table 5. Profiling Subjects Using Formats

How Many Questionnaires Are Missing Data?				
tot_miss	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Complete	5	62.5	5	62.5
Incomplete	3	37.5	8	100
What is the Distribution of Overall "Positive" Subjects?				
tot_yes	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Not Positive	4	50	4	50
Somewhat Positive	2	25	6	75
Very Positive	2	25	8	100

PROFILING WITH A PATTERN VARIABLE

Another way to look at the set of responses from each subject is to consider the complete set of responses for each subject as a pattern. For example, perhaps we have a subset of subjects who answered 'Yes' consistently to one group of questions and not another. Or maybe one of the questions was consistently omitted ('Missing').

CREATING THE PATTERN VARIABLE

We can use the binary 0/1 responses to each question variable to create a summary pattern variable for each subject using SAS® CAT function:

```
data example3 (drop = i);
     set example2
     length pattern $ 6;
     pattern = cat(of q1-q6);
 run;
```

With the code above we created a character variable Pattern of length 6 to hold the string of responses to variables Q1 through Q6. The CAT function reads through the parameter list in order, concatenating the contents of the next variable to the contents of Pattern so far.

The Patterns for each of the eight subject's responses are below in Table 6. Notice that as SAS translates the contents of each individual input variable, 'Missing' values are added with the standard SAS missing notation '.':

Table 6. Example Data After Adding Response-Pattern Variable

test_site	subject	Q1	Q2	Q3	Q4	Q5	Q6	pattern
Site A	SubjA	.	1	1	1	1	0	.00000
Site A	SubjB	0	0	1	0	1	0	001010
Site B	SubjC	1	1	0	1	0	0	110100
Site A	SubjD	0	0	.	.	0	1	00..01
Site A	SubjE	1	1	1	1	1	1	111111
Site B	SubjF	0	0	1	0	1	0	001010
Site B	SubjG	1	1	1	1	1	1	111111
Site A	SubjH	0	0	.	.	0	1	00..01

PATTERN VARIABLE ANALYSIS EXAMPLE

Just as we profiled individual subject response-type totals earlier, we can use unformatted and formatted PROC FREQs to create an initial window into the distribution of response patterns in the data.

The unformatted results in Table 6 below show us that some patterns occur more frequently than others. For example, note that more than one Subject has 'Missing' responses for variables Q3 and Q4 only. This finding may have implications for the design of the study data collection instrument and/or interpretation of study results.

Table 6. Unformatted Frequency Distribution for the Pattern Variable

pattern	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0.1111	1	12.5	1	12.5
00..01	2	25	3	37.5
001010	2	25	5	62.5
110100	1	12.5	6	75
111111	2	25	8	100

We can also create various Formats to apply to the Pattern variable to answer specific questions about the distribution of study responses, using the results of the unformatted distribution as a guide.

We can see, for example, that more than one subject has 'Missing' responses for variables Q3 and Q4 only. Since this finding may have implications for the design of the study data collection instrument and/or interpretation of study results, we'll build it specifically into our format. We'll also use the Order = Freq option on PROC FREQ so we can quickly see which patterns are occurring most frequently.

Table 7. Formatted Frequency Distribution for the Pattern Variable

FORMAT and PROC FREQ CODE

```
proc format;
   value $ pattern
   '111111' = 'All Yes'
   '000000' = 'All No'
   '001010' = 'Q3 and Q5 Yes'
   other  = 'Other Mixed Responses'
;
run;

proc freq data = example3 order = freq;
     tables pattern/list missing;
     format pattern $pattern.;
run;
```

RESULTS

pattern	Frequency	Percent
Other Mixed Responses	4	50
All Yes	2	25
Q3 and Q5 Yes	2	25

This particular quickly highlights that most response sets in the data are mixed, but that the Q3/Q5 only pattern occurs as frequently as the All Yes pattern and might warrant further exploratory analysis.

PROFILING VARIABLES ACROSS THE DATASET

Our example data is made up of six responses from eight unique subjects and is relatively easy to profile and understand using the binary and pattern profiles for the individual observations. But on many occasions SAS programmers work with much larger datasets and need methods to summarize individual variables across all observations.

PROC MEANS WITH PRINT

Because we've chosen the numeric data type for our binary variables in this example, much of the work of profiling individual variables can be handled by PROC MEANS.

Below is the code to invoke PROC MEANS for all our binary and response totals variables. Notice that we've asked for a subset of the analysis variables PROC MEANS can produce: N, NMISS, SUM, MIN and MAX:

```
proc means data = example3 n nmiss min max sum;
    var q1-q6
        tot_yes
        tot_no
        tot_miss;
run;
```

The output produced by this code contains a summary snapshot for both our binary variables and the response-total summaries for each observation.

Table 8. PROC MEANS Output

	PROC MEANS Statistic				
Data Field	**N**	**N Miss**	**Minimum**	**Maximum**	**Sum**
Q1	7	1	0	1	3
Q2	8	0	0	1	4
Q3	6	2	0	1	5
Q4	6	2	0	1	4
Q5	8	0	0	1	5
Q6	8	0	0	1	4
tot_yes	8	0	1	6	25
tot_no	8	0	0	4	18
tot_miss	8	0	0	2	5

The overall Minimum and Maximum statistics for fields Q1-Q6 give us quick Quality Assurance confirmation that the only values in those fields are the required 0's and 1's. Using the Minimum and Maximum statistics for the

tot_yes field tells us that across the data, the maximum number of Yes responses for a subject was 6 and the minimum was 1.

The SUM field contains the result of adding up all the values in each field across the data. For fields Q1-Q6 this means adding up the 1's and SUM can be interpreted as the total number of 'Yes' responses for each of those questions. For the tot_ variables, SUM provides database totals overall for the three valid response types. For example, the SUM statistic for tot_yes is 25, indicating that of the 48 responses in the data, 25 were Yes.

And finally the N (Number of Non-Missing responses) and NMISS (Number of Missing responses) statistics show us a pattern of almost complete data. Only fields Q1, Q3 and Q4 are sometimes missing and in all cases the number of observations affected is small. N and NMISS

CONCLUSIONS

Using 0/1 binary variables to code 'Yes'/'No'/'Missing' data elements can be a powerful tool to quick and efficient analysis of the resulting dataset. The choice of the numeric versions of the codes allows the binary variables to be manipulated by SAS® arithmetic functions such as SUM and NMISS and PROC MEANS summary statistics. Data-specific formats can be created using PROC FORMAT to add specificity to the descriptive statistics of these binary variables for both reporting and Quality Assurance purposes.

ACKNOWLEDGMENTS

CONTACT INFORMATION

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

Laurie Hamilton
Healthcare Management Solutions, LLC
10420 Little Patuxent Parkway, Suite 101
Columbia, MD 21044
Email: lhamilton@hcmsllc.com