

Using Survey Data—Tips to Pick Up Speed on the Road to Analysis

Cynthia L. Williamson, Technology Assessment Group, San Francisco, CA
Suzanne D. Kreutzer, Social Policy Research Associates, Menlo Park, CA

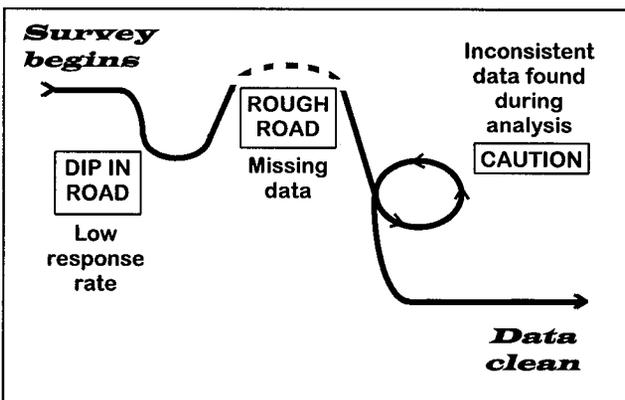
ABSTRACT

In this paper, we offer tips to help SAS® programmers get to know their survey data, thereby ensuring that the road to analysis is smooth. We will address techniques to prepare for a survey, track questionnaire returns, ensure accurate data input, check for data quality, and create a data analysis file.

INTRODUCTION

Surveys are a standard research tool used to gather data from many different types of respondents: patients, employees, consumers, students, and other diverse populations. Despite the variety in the research questions to be addressed through the use of surveys, the need for clean and accurate data for analysis is paramount. Regardless of the population to be surveyed or the types of questions to be answered, programmers face similar issues; similar strategies can be used to identify and deal with problems of missing, inconsistent, or otherwise "dirty" data.

Before you begin analyzing your survey data, you will be involved in at least one of the following tasks: preparing the questionnaire, pretesting and fielding the survey and getting results, entering the survey data, or preparing the data for analysis. Although these tasks are usually considered to be a linear process, they are much more likely to be completed iteratively. As you become more familiar with your data, you may find it necessary to step back, reexamine your assumptions about the data, and rerun some programs given what you learned about the data. The figure below displays a path that might be taken on the road to analysis.



Example of Path Leading to the Road to Analysis

In the example above, several problems are encountered before the data are clean. Because there will always be respondents who do not complete the survey as you expect they would, your data will be imperfect. The best time to find these problems is before you get to the data analysis phase.

The rest of this paper is devoted to techniques that we found helpful in smoothing out the inevitable bumps in the road as we readied survey data for analysis.

PREPARING FOR THE SURVEY

The ideal time for the SAS programmer to be involved in a survey is before the questionnaire is printed. No matter how many times a questionnaire has been reviewed by researchers, it is always useful to have a programmer look it over. SAS programmers are the most likely to identify problems such as faulty skip patterns, inconsistent use of the same scale for different items, and other problems that may make the analysis more difficult or may even make it impossible to answer the research question.

If the population to survey is large, you will most likely be involved in selecting a random sample for your survey. The following code (from *SAS Language and Procedures: Usage 2, Version 6, First Edition*, p. 235) makes it easy for you to select a simple random sample of exact size:

```
*****  
* SELECT A SIMPLE RANDOM SAMPLE OF EXACT SIZE *  
*****;  
data wuss.sample (drop=k n);  
  *** K IS THE NUMBER THE NUMBER TO SAMPLE AND ***;  
  *** N IS THE POPULATION ***;  
  retain k 1200 n;  
  if _n_ = 1 then n = total;  
  set wuss.populatn nobs=total;  
  if ranuni(998877665) <= k/n then do;  
    output;  
    k = k - 1;  
  end;  
  n = n - 1;  
  if k = 0 then stop;  
run;
```

FIELDING THE SURVEY

Mailing out questionnaires or conducting telephone interviews are not SAS-related tasks so most of the tasks related to fielding the survey are not likely to be performed by you. However, you may be asked to prepare labels or data files for vendors who will be conducting the survey; you may also be asked to track returns and calculate response rates.

Preparing Survey Materials

For a mail survey, you can use PROC FORMS to prepare address labels or ID labels for questionnaires. To create labels:

```
*****  
* PREPARE LABELS: 1 INCH x 2-5/8, 3 ACROSS, 10 DOWN *  
*****;  
proc forms data=wuss.sample  
  width = 33 lines = 6 down = 0  
  nskip = 1 nacross = 3 nbetween = 6  
  indent = 0 ncopies = 1;  
  line 2 id / indent=20;  
  line 3 name / lastname;  
  line 4 addr1 / remove;  
  line 5 city state zip / pack;  
run;
```

We created the above example using the SAS System, version 6.10, for Windows. From the print setup menu, we changed all margins to 0.1" and changed the font to 10-point Letter Gothic (W1). If you use different sized labels or a different font, you will have to experiment with the

options in PROC FORMS and in the print setup menu to print the labels properly.

If your survey is being conducted by phone and you are providing the sample to a vendor, you may be responsible for data management issues such as writing data in a machine-readable format or printing the data for the vendor's use.

Tracking Returns

Keeping close track of the returned surveys is an important part of survey coordination. Data kept in a spreadsheet or database can be used to monitor response rates as the survey is being conducted, to identify individuals who require follow-up contact, and to conduct analysis of response bias. A low overall response rate is an indication that additional steps should be taken to increase the response rate. If response rates are lower for important subsets of targeted respondents (for example, if men respond at a lower rate than women), steps can be taken before surveying is complete to ensure an overall representative sample. In addition, data on response rates are necessary for any follow-up contact with possible respondents. Identifiers with codes indicating no response can be merged with the primary address data set to prepare labels for follow-up mailings. Finally, return dates can be used to evaluate levels of response bias, as can demographic or other data merged from available sources.

In tracking returns, you will need to capture the following information, at minimum:

- All identification variables (i.e., those variables that uniquely identify each member in the sample data set).
- The return code (see below for more information).
- The date the questionnaire was received.

You can use a DATA step with the WINDOW statement to enter tracking information directly into a SAS data set. In our example which follows, the tracking data set is initially set up to include identifiers of all potential respondents, with default return codes indicating no response. The user is prompted for the identification number of each returned survey and the status of the response. Return dates are automatically tracked by assigning them the date of data entry. Data entry is checked by comparing the listing of each day's entries with identifiers from the questionnaire.

```
*****
* 1. BACK UP STATUS DATA SET FIRST *
*****
data wuss.statbak;
  set wuss.status;
run;

*****
* 2. START WITH AN INTRO SCREEN *
*****
data _null_;
  window start color=white
  #4 @19 'STATUS OF WUSS QUESTIONNAIRES'
  #6 @7 'SUGGESTIONS -'
  #8 @8 '- If you make a mistake on the'
  #9 @8 'RETURN CODE, re-enter; your last'
  #10 @8 'entry for each ID will be saved.'
  #12 @8 '- If you make a mistake on the ID number,'
  #13 @8 'remember to correct the RETURN CODE'
  #14 @8 'for the incorrectly-entered ID,'
  #15 @8 'i.e., NOT YET RETURNED'
  #16 @8 'after checking Status Printout.'
  #19 @35 'PRESS ENTER TO CONTINUE'
  display start;
  stop;
run;
```

```
*****
* 3. UPDATE THE STATUS OF RETURNED SURVEYS *
*****
data ret_code;
  window ret_code color=yellow
  #5 @10 'ENTER ID NUMBERS OF QUESTIONNAIRES BELOW'
  #7 @15 'RETURN CODES: 1=RETURNED'
  #8 @15 '2=UNDELIVERABLE'
  #9 @15 '0=NOT YET RETURNED'
  #10 @15 '(CORRECTIONS)'
  #12 @5 'ID NUMBER' @19 id best3.
  #14 @5 'RETURN CODE' @19 code 1.
  #20 @10 'After entering the last value, enter STOP'
  #21 @10 'on the Command line (<home>/STOP) instead'
  #22 @10 'of pressing ENTER';
  do while (upcase(_cmd_) ne 'STOP');
    display ret_code bTank;
    ret_date = today();
    output;
    id = .;
    code = .;
    ret_date = .;
  end;
  stop;
run;

*****
* 4. DELETE ANY BLANKS/DUPLICATES *
*****
proc sort data=ret_code;
  by id;
run;

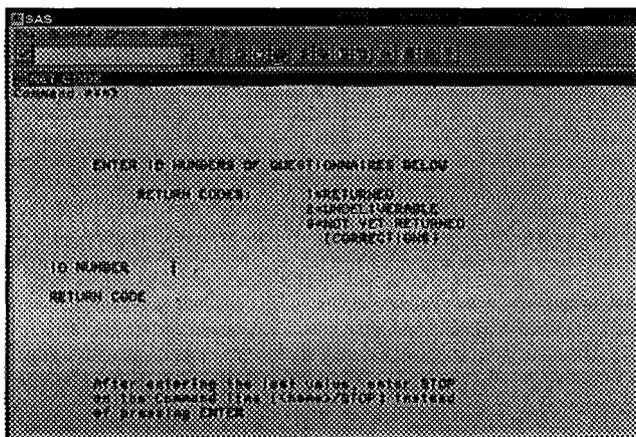
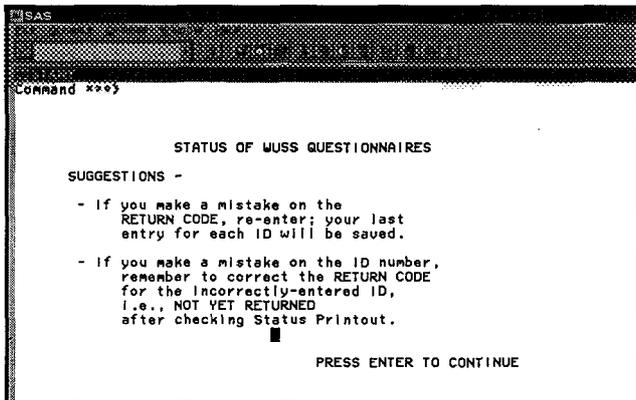
data ret_code;
  set ret_code (where=(id ne .));
  by id;
  if last.id;
run;

*****
* 5. LIST OUT RETURN CODE ENTRY (NEW) *
*****
data _null_;
  set ret_code;
  tday = date();
  tm = time();
  file print header=x;
  put @15 id z3. @25 code @35 ret_date date7.;
  return;
x: put @10 "Questionnaires Entered"
  @35 tm hmmm5. tday weekdate32. //
  @15 'ID' @24 'Code' @32 'Return Date' /
  @15 '---' @24 '----' @32 '-----';
return;
run;

*****
* 6. LIST OUT RETURN CODE ENTRY (PREVIOUS) *
*****
data _null_;
  set wuss.status;
  tday = date();
  tm = time();
  file print header=x;
  put @15 id z3. @25 code @35 ret_date date7.;
  return;
x: put @10 "Questionnaires Entered BEFORE"
  @42 tm hmmm5. tday weekdate32. //
  @15 'ID' @24 'Code' @32 'Return Date' /
  @15 '---' @24 '----' @32 '-----';
return;
run;

*****
* 7. UPDATE STATUS DATA SET *
*****
data wuss.status;
  merge wuss.status
  ret_code;
  by id;
run;
```

The following screens are displayed by the WINDOW statements:



The default return code every respondent starts with is not returned/not completed. Other common return codes are: (1) returned/completed, (2) undeliverable by post office (mail survey), (3) incorrect phone number (phone survey), (4) respondent refused to participate, and (5) respondent unable to respond due to death, illness, or other circumstances. You will probably need to add other codes specific to your study. For example, you may be surveying people about their participation in a program and they may never have participated or respondents may not be able to respond because they do not speak the language used for the survey.

Additional ways to track returns include:

- Create a database using SAS/FSP®.
- Create a database in an application such as Microsoft Access® and import the data into the SAS System.
- Type the information into a spreadsheet, word processing document, or an ASCII file and read that data into a SAS data set.
- Manually track using pencil and paper. (This is the least flexible option for tracking; we do not recommend it.)

ENTERING SURVEY DATA

Unless you are lucky enough to have someone hand you a disk with an easily readable data file, you will need to have the data entered, either in-house or out.

Editing Data Prior to Data Entry

Your data will be more accurate if the data entry person keys in what is seen on the page without trying to

remember special input rules. Therefore, it may be beneficial to have some type of editing occur before the data are entered. For example, what if your questionnaire asks for a number (such as usual hours worked per week) and the respondent enters a range (such as "40-50")? A data editing rule would specify whether the response should be edited to reflect the lowest value in the range, the highest value, the midpoint, or set to missing.

The first step in developing data editing rules is to review a few dozen questionnaires and then to develop rules based on how the respondents answered the questions. The person responsible for data editing should spot-check a few questionnaires on an ongoing basis because different types of problems will continue to pop up.

Entering Data In-House

If the data are entered in-house, you may be responsible for setting up a data entry system. If so, you can consider choices such as using a WINDOW statement within a DATA step (as described in *Tracking Returns* above), SAS/FSP, or database packages (e.g., Microsoft Access). To ensure the best entry possible, you should keep these points in mind:

- Lay out the data entry screen so that it looks as much like the questionnaire as possible.
- Put formats on data entry fields, such as dates or Social Security numbers, so that the data entry operator does not have to spend time entering unnecessary characters.
- Place range specifications on variables so that out-of-range responses cannot be entered.

You will want at least some data entered twice to verify the accuracy of the original data entry. At minimum, you want all ID variables to be entered again in another file. You should also have a sample (or, ideally, all) of the questionnaires reentered into a data set with the same variables as the original. Once the data have been entered a second time, you can use PROC COMPARE to identify any mismatches between the data sets. To compare the original entry with the double data entry:

```
*****
* 1. SORT DATA BY ID VARIABLES (IF NOT ALREADY SORTED)*
*****;
proc sort data=wuss.survey;
  by id1 id2;
run;

proc sort data=wuss.doubssurv;
  by id1 id2;
run;

*****
* 2. COMPARE ALL VARIABLES FROM THE SURVEY *
*****;
proc compare
  base=wuss.survey
  compare=wuss.doubssurv
  listall transpose;
  id id1 id2;
  var q1-q5;
run;
```

The output from PROC COMPARE will list out any mismatches between the original entry (base file) and the double data entry (comparison file). Once the correct response is identified and the appropriate data set (original or double data entry) is updated, you can rerun the same PROC COMPARE program to verify that the corrections were entered accurately and everything matches.

Out-Sourcing Data Entry

If you have a vendor enter the data, you will need to agree on the format in which you want the data delivered (i.e., fixed column, delimited by a designated character, or another format). If the data will be delivered in a fixed-column format, you will need to provide a copy of the questionnaire with the exact positions for variable entry indicated (sometimes called “card and column” locations). We usually hand-write the card and column locations in the margin of a copy of the questionnaire. When possible, the data entry positions are typed in as a part of the final questionnaire sent to the printer.

If you are responsible for preparing a request for a bid from data entry vendors, you should provide the vendors with the following information:

- The date by which you want to receive the bid (which should include sales tax, if appropriate).
- Size of questionnaire—either the total number of cards and columns to be entered or the number of fields/variables to be entered.
- Total number of questionnaires to be entered.
- The date the first questionnaire will be received.
- The date the last questionnaire will be received.
- The date you want to have the data returned. This may be multiple dates if you request test data, any interim data, and the final data.
- Any special instructions regarding data entry and how you want them to handle problems (e.g., “flag” the problem, enter the largest (or smallest or neither) number if respondent indicated more responses than can be entered).
- Your name and phone number as the contact person.

Special Issue: Circle-All-That-Apply Items

There are several ways to enter questionnaire items that ask respondents to circle all responses that apply. An example of a circle-all-that-apply item is:

Which of the following SAS procedures have you used? (Please circle all that apply.)	
PROC MEANS.....	1
PROC GCHART	2
PROC REG.....	3
PROC FABRICATE	4

You can have circle-all-that-apply items entered using the methods below:

- Enter each category in its own column. For example, if responses were entered in columns 74-77 and if responses 1 and 4 were circled, a “1” would be entered in column 74 and a “4” would be entered in column 77.
- Allow enough space for entry should a respondent indicate all responses and have any responses entered in the space left justified. Using the example above, “14” would be entered beginning in column 74. (See *Creating Your Analysis File* below for suggested code for reading in data in this format.) This method minimizes the number of key strokes.

- Enter a “1” in the specified column if the respondent circled the response or enter a “0” if the response was not circled. Using the example above, “1001” would be entered beginning in column 74. This method is probably the most prone to data entry error; therefore, we do not recommend it.

Load Sample Data and Check

We think it is wise to enter a small batch of test data (approximately 10-25 questionnaires). After you read in the test data, carefully review frequencies and means; in addition, print all the test data and check every variable for a subset of questionnaires. Once you have verified that your load program reads the data properly and that data entry staff have understood any special instructions you have given to them, you have significantly reduced the risk that some of your questionnaires will not be entered or read in properly.

PREPARING DATA FOR ANALYSIS

The key to preparing your data for analysis is getting to know your data. This means that you need to read the questionnaire, think about how respondents might misunderstand the instructions and questions, make sure you understand any skip patterns, and pore over any frequencies, means, or univariates you have. You will recoup the time that you spend getting to know your data when you are in the analysis phase.

Cleaning Your Data

Before you begin creating your analysis file, you should look for these undesirable characteristics in your data: duplicate observations, blank questionnaires, out-of-range responses, lack of internal consistency, and outliers:

- **Duplicate observations.** Sometimes a respondent sends in more than one questionnaire or the same questionnaire is entered into the database more than once. To identify any problem cases:

```
*****
* 1. SORT DATA BY ID VARIABLES (IF NOT ALREADY SORTED)*
*****
proc sort data=wuss.survey;
  by id1 id2;
run;
*****
* 2. OUTPUT OBS THAT DO NOT HAVE A UNIQUE ID          *
*****
data dups;
  set wuss.survey;
  by id1 id2;
  if not(first.id2 and last.id2);
run;
*****
* 3. PRINT OUT ALL VARIABLES FOR DUPLICATE OBS        *
*****
proc print data=dups;
  id id1 id2;
  title 'Identify Duplicate Observations';
run;
```

If all variables for all observations are exactly alike, you can safely delete any duplicates. However, if any variables differ, you need to verify that the ID variables for the questionnaires were entered accurately and, if they were, you need a rule (e.g., questionnaire received first) as to which questionnaire to keep.

- **Blank questionnaires.** Sometimes a questionnaire is sent to data entry even though it is blank. For example, the respondent refused to complete the questionnaire but there is ID information on the cover page that is entered. To identify any problem cases:

```
*****
* 1. CREATE A VAR TO COUNT NONMISSING DATA VALUES *
*****
data anyblank;
  set wuss.survey;
  cnt_data = 0;
*****
* 2. CREATE ARRAYS OF ALL CHARACTER & NUMERIC VARS *
*****
array charvars {*} _char_;
array numvars {*} _num_;
*****
* 3. INCREMENT COUNTER VAR FOR NONMISSING CHAR VARS *
*****
do i=1 to dim(charvars);
  if charvars{i} ne '' then cnt_data = cnt_data + 1;
end;
*****
* 4. INCREMENT COUNTER VAR FOR NONMISSING NUM VARS *
*****
do i=1 to dim(numvars);
  if numvars{i} ne . then cnt_data = cnt_data + 1;
end;
label
  cnt_data = 'Count number of non-missing variables';
run;
*****
* 5. GET A FREQUENCY ON THE COUNTER VARIABLE *
*****
proc freq data=anyblank;
  tables cnt_data;
run;
*****
* 6. PRINT OUT IDS FOR BLANK QUESTIONNAIRES *
*****
proc print data=anyblank noobs label double;
  var id1 id2 cnt_data;
  where cnt_data <= 2; ** 2 ID VARIABLES ONLY;
  title 'Identify Blank Questionnaires';
run;
```

Just to be safe, you should verify that the questionnaires are actually blank before deleting the observations. You may also want to update the return code in your tracking data set, if applicable.

- **Out-of-range values.** You should run PROC MEANS on all numeric variables and PROC FREQ on all character variables. Then check the minimums and maximums to verify that there are no out-of-range values. However, for some surveys, you may have to load and check interim data several times until all the data have been entered. Because it is tedious to review similar output multiple times and you will have to write code to print out problems anyway, you may want to program out-of-range checks similar to the following:

```
*****
* 1. MACROS FOR RANGES AND TO SET FLAG FOR OUT-OF- *
* RANGE RESPONSES *
*****
%let r01to02 = .,1,2;
%let r01to05 = .,1,2,3,4,5;
%macro probflag (varname,range);
  if &varname not in(&&range) then p&var = 1;
%mend probflag;
```

```
*****
* 2. SET OUT-OF-RANGE FLAGS FOR YOUR DATA *
*****
data outrange;
  length
    probname $100
    tempprob $7;
  set wuss.survey;
  array allvars {*} q1-q5; ** ARRAY ALL VARS;
  array probvars {*} pq1-pq5; ** CREATE PROBLEM FLAGS;
*****
* 3. SET THE PROBLEM FLAGS TO ZERO *
*****
do i = 1 to dim(probvars);
  probvars{i} = 0;
end;
*****
* 4. IF VAR NOT IN RANGE, SET PROBLEM FLAG TO ONE *
*****
%probflag (q1, r01to02);
%probflag (q2, r01to05);
%probflag (q3, r01to05);
%probflag (q4, r01to02);
%probflag (q5, r01to05);
*****
* 5. CREATE VARIABLE THAT IDENTIFIES PROBLEM VARS *
*****
do i = 1 to dim(probvars);
  tempprob = '';
  if probvars{i} = 1 then do;
    call vname(allvars(i),tempprob);
    probname = trim(left(probname)) || ' ' ||
      trim(left(tempprob));
  end;
end;
run;
*****
* 6. CHECK THAT PROBLEM FLAGS WERE CREATED PROPERLY *
*****
proc freq data=outrange;
  tables pq1*q1 pq2*q2 pq3*q3 pq4*q4 pq5*q5
    probname
  / list missing;
  title 'Check Creation of Problem Flags';
run;
*****
* 7. SUBSET DATA SO THAT YOU HAVE ONLY PROBLEM CASES *
*****
data outrange;
  set outrange;
  if sum(of pq1-pq5) > 0;
run;
*****
* 8. LIST OUT PROBLEMS *
*****
proc sort data=outrange;
  by id1 id2 date initials;
run;
proc print data=outrange noobs uniform;
  var probname q1-q5;
  by id1 id2 date initials;
  title 'Print Out Data with Out-of-Range Values';
run;
```

Someone could take the listing of problem cases, compare the variable values printed with the questionnaire, and mark the correct responses on the printout. After the corrections are made to the SAS data set, you should rerun the program to verify the corrections.

Once you have verified that your program is working accurately, checking for out-of-range responses after more data have been added to the file is a very speedy process.

- **Lack of internal consistency.** Data integrity can be improved if you make sure the data "make sense." Internal consistency checks (or multivariate checks), where responses for individual observations are evaluated based on responses to other items, can be of help. For example, if your survey asks for pregnancy information, respondents who are males should not have data for that section of the questionnaire. Rules regarding changes to inconsistent data (e.g., set pregnancy information to missing for males) should be developed and applied to all records programmatically. A special set of internal consistency checks you should write involve skip patterns. Respondents will sometimes answer questions that they are directed to skip. Decision rules should be developed as to whether data in the questions that should have been skipped should be set to missing or whether the response to the question before the skip should be reset for consistency.
- **Outliers.** Run PROC UNIVARIATE with the PLOT option for continuous variables to identify extreme values. Use the ID statement to display the ID variable with the five lowest and highest values. You need to determine if any outliers are valid values and, if so, whether they should be included in analysis.

Creating Your Analysis File

Once you have checked your data and pronounced it clean, you will create an analysis file. Unlike cleaning your data, which is best done in a series of separate programs, you may prefer to create your analysis file in one program. The following modules may be a part of your analysis file creation:

- **Special missing values.** There are many reasons why a respondent does not answer a question. Oftentimes, these "nonresponses" are purposely given out-of-range values. When you conduct your analyses, you will not want these out-of-range values included in means or in frequencies. This is when the SAS special missing values are invaluable. Common missing value codes we have used are:
 - .S for skipped.
 - .D for don't know.
 - .R for refused.
 - .N for not applicable.
- **Circle-all-that-apply items.** The following macro can be used to create dichotomous variables from circle-all-that-apply items which are entered left justified:

```
*****
* 1. CREATE YOUR MACRO. PARAMETERS ARE: *
* ROOT = STEM OF THE VARIABLE NAME (E.G., Q6 IF *
* YOU WILL BE CREATING VARS Q6 1-Q6 5). *
* N = FINAL NUMBER OF CIRCLE-ALL VARIABLES. *
* O = NUMBER OF CIRCLE-ALL VARS ENTERED. MAY *
* BE LESS THAN N IF ALL POSSIBLE RESPONSES *
* WOULD NOT BE CIRCLED AND ENTERED. *
*****;
%macro circle (root, n, o);
  *** MAKE FLAG TO KEEP TRACK OF ANSWERS ***;
  flagcirc = 0;
  *** ARRAY THE FINAL CIRCLE-ALL VARIABLES ***;
  array &root {*} &root._1-&root.&n;
  *** ARRAY THE CIRCLE-ALL VARIABLES ENTERED ***;
  array t&root {*} &root.t1-&root.t&o;
  *** INITIALIZE FINAL CIRCLE-ALL VARIABLES TO NO ***;
  do i = 1 to dim(&root);
    &root{i} = 0;
  end;
end;
```

```
*** SET ANY YES VALUES FOR CIRCLE-ALL VARIABLES ***;
do i = 1 to dim(&root);
  do j = 1 to dim(t&root);
    if t&root{j} = i then do;
      &root{i} = 1;
      flagcirc = 1;
    end; ** DO J = 1 TO ...;
  end; ** DO I = 1 TO ...;
end;
*** SET FINAL CIRCLE-ALL TO MISSING ***;
*** IF EVERYTHING IS NO ***;
if flagcirc = 0 then do i = 1 to dim(&root);
  &root{i} = .;
end;
%mend circle;
```

- **Labels and formats.** Even though it can be a boring task, labeling variables and applying formats is important. Labels and formats will allow end users to more easily and accurately assimilate the data. In addition, if you have to go back to the data after having set it aside for some time, you too will benefit from having more descriptive output.
- **Variable creation and recoding.** It is always important to verify that the creation of variables from survey responses was done correctly. You can look at cross-tabulations of variables using PROC FREQ with the LIST option to display the input variables by the newly created variable. SAS code that performs calculations or manipulates one or more variables to create another can be double-checked by printing out variables used in the calculation and the created variable for a number of observations and then performing hand calculations. This may seem tedious, but is a good way to catch inadvertent errors.
- **Get the basics.** Whenever you create a permanent SAS data set, run PROC CONTENTS, PROC MEANS (for numeric variables, including ID variables), and PROC FREQ (for character variables).

CONCLUSION

Conducting surveys and preparing data for analysis can be a bumpy road. Programmers can prepare for the twists and turns by offering an important perspective to the research team as surveys are designed and conducted. Although programmers should expect detours along the way, they help lay the foundation for quality research by tracking the flow of survey traffic, paving the potholes of missing and inconsistent data, and preparing a smooth road to analysis.

REFERENCES

SAS Institute Inc. (1991), *SAS Language and Procedures: Usage 2, Version 6, First Edition*, Cary, NC: SAS Institute Inc.

ACKNOWLEDGMENTS

Kathryn A. Valdés, SRI International, developed the circle macro presented in this paper.

We also thank Cyndi's colleagues at Technology Assessment Group for their ideas.

We especially thank Alex Stagner for his support.

SAS and SAS/FSP software are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

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