Paper 232-2011

SAS® Data Management: How Raw Variables with Complex Skip Patterns and Missing Values are Interpreted with Derived Variables

Isabel H. Perry, California State University, East Bay
Samantha E. Dilworth, Elise D. Riley, University of California, San Francisco

ABSTRACT

Shelter, Health, and Drug Outcome in Women (SHADOW) is a research study with the Department of Medicine at the University of California, San Francisco that has been gathering quantitative information through biannual interviews with impoverished women in San Francisco. The project seeks to understand how competing needs influence the health and HIV risk behaviors of homeless women in order to inform healthcare delivery and social services. The focus of this paper is to employ data management techniques in SAS to create common variables of interest that may be requested by study investigators. A question of particular importance is whether a participant partakes in serodiscordant sex (i.e. sex with a man of the opposite HIV status). Roughly half of the 301 women in the study are HIV positive, and multiple conditions must be met for a person of either serostatus to be eligible for the outcome. Ensuring the raw data are transformed into accurately created derived variables and easily understood reports is a necessity of which the use of SAS has many advantages.

WARNING: This presentation is based on data from a research study regarding HIV risk. It contains language regarding sex that some individuals may find offensive. For this reason words in brackets denote substitutions for the original names.

INTRODUCTION

Handling realistic datasets and deriving variables are important preliminary steps for most analyses, yet they are topics not often discussed in introductory statistical courses. Instead new learners of statistics are frequently taught the basic principles of data cleaning through their own applications. The detailed process of data preparation is beneficial to understanding how analytic results evolve from a raw dataset. Here I will demonstrate initial steps of data preparation through creating variables that require multiple assumptions. Data used herein come from "Shelter, Health and Drug Outcomes among Women" (SHADOW), an ongoing cohort study that seeks a better understanding of infectious disease risk among impoverished women.

QUESTIONNAIRE

The SHADOW questionnaire is separated by sections on various topics. I will focus on the section with the most complex skip pattern regarding sexual activities. For these sensitive questions the interviewer has the aid of computer software, ACASI (Audio Computer-Assisted Self-Interview), which enables private input of answers. Each response generates unique pathways on the questionnaire; the question is skipped if it does not apply to the individual. For example, a respondent who reports having no sex in her lifetime will skip out of the entire sexual activities section. Skipped questions are assigned a value as "9", "99", or "999" depending on the format of the answer. This coding indicates the question does not apply to this participant.

Respondents may also refuse to answer any question at any time. These answers are given a value of "8", "98", or "998" depending on the format of the answer. If a respondent chooses "refuse to answer" then she is skipped out of any related subsequent questions. For example, if she refuses to answer the first question "How many sex partners have you had in the past 6 months" then the respondent is skipped from the entire sexual activities section.

A raw variable's frequency including nines that indicate skipped is often meaningless without doing the necessary calculations to determine which previous variable is driving the skip. Thus the DATA step can be used to show why these participants were skipped out. This not only provides useful frequencies but it also assesses whether the ACASI program is performing the defined skip patterns correctly. In addition, reasons for missing observations must be examined to ensure the skip logic covers all possibilities.

SKIP PATTERNS AND MISSING DATA

Due to the complex nature of the skip patterns in the sexual activities section, final variables are derived by information collected at multiple points. In the following examples, the first handful of variables utilizes the skip pattern to illustrate the creation of derived variables.

Table 1. Original survey questions from sexual activities section used to create derived variables.

Raw Variable	Question
SEX1	How many sex partners have you had in the past 6 months?
SEX2	How many of those (Response to SEX1) sex partners were men?
SEX3	How many of those (Response to SEX2) partners did you have [type A] sex with?
SEX4	How many of those (Response to SEX2) partners did you have [type B] sex with?
SEX2B	Was this sex partner a man?
SEX3B	Did you have [type A] sex with this partner?
SEX4B	Did you have [type B] sex with this partner?
SEX7B	Thinking about [type A] and [type B] sex only, were there times when you did not use a condom during the past 6 months?
SEX7	During the past 6 months, how many partners did you have [type A] or [type B] sex with and not use a condom?

From Table 1, variables ending with a “B” denote binary responses where “1” is “yes” and “0” is “no”. All other variables are count variables. For example, if a respondent answers one sex partner (SEX1=1), they move on to answer if this partner was male or not (SEX2B). If a woman answers more than one sex partner (SEX1 > 1), she is next asked how many of those partners were male (SEX2), etc. In order to be asked all the questions in the sexual activities section, the respondent must provide affirmative information on [type A] or [type B] sex with at least one man.

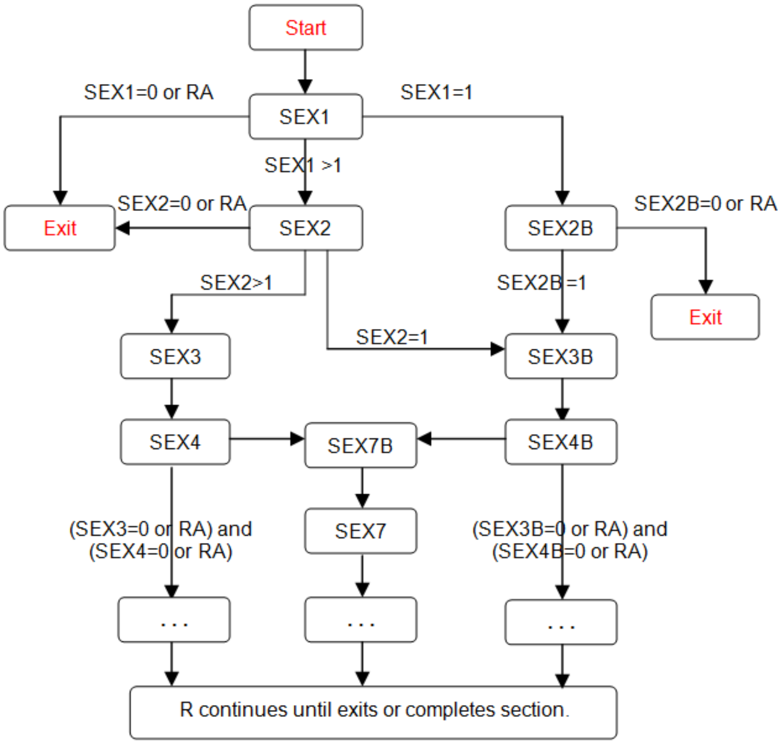


Figure 1. Skip pattern visually explained for sexual activities section of questionnaire (RA=Refuse to answer).

PRELIMINARY ACTIONS

DETERMINING THE CONTENT

Before coding the skip patterns in the DATA step, it is good practice to do a few quick procedures that will save time later in the program. Beginning with PROC CONTENTS is useful because it lists all the variables in the dataset along with their type, length, format, and label.

```
PROC CONTENTS DATA=SEXRISK VARNUM;
RUN;
```

Often it is helpful to include the VARNUM option which lists variables in the order they were created. This is particularly important because the SHADOW dataset contains over 2,000 variables, many of which were not necessarily created in a sequential order of SEX1, SEX2, SEX3, etc. If the order is not known then we are forced to code ARRAY statements such as these:

```
ARRAY AVOID(*) SEX85A SEX85B SEX85C SEX85D SEX85E;
```

It is easily seen how more than a handful of variables will be overly tedious to code when the variables end in an alpha character. Variables ending in sequential numeric characters can use a single dash between the variables. The VARNUM option provides enough information on variable order to avoid lengthy ARRAY statements requiring a detailed list of variables such as above. Instead, a double dash can be used between the appropriate variables, as demonstrated by the following array, USE. The double dash calls all variables that were created between SEX85A and SEX85E:

```
ARRAY USE(*) SEX85A--SEX85E;
```

If, perhaps, the PROC CONTENTS output shows the variables were created backwards, the ARRAY statement is easily changed to the following:

```
ARRAY USE(*) SEX85E--SEX85A;
```

RENAMING RAW VARIABLES

Another common practice is to rename the raw variables. It is not necessary to do so but gives novice end users the ability to compare renamed variables with their corresponding raw variable to check for coding accuracy while also preserving the raw variable values. The array, NEW, demonstrates the single dash and also shows that “refused to answer” responses may be coded before the skip logic. Writing code with dual purposes is an efficient method to clean the data.

```
ARRAY ORIG(*) SEX85A--SEX85E;
ARRAY NEW(*) NEW85 1-NEW85_5;
DO I=1 TO DIM(ORIG);
       IF ORIG(I)=8 THEN NEW(I)=.R; *.R=Refuse to answer;
       ELSE NEW(I)=ORIG(I);
END;
```

RUNNING PRELIMINARY FREQUENCY TABLES TO SEE ALL RAW VALUES

In addition to PROC CONTENTS, it is incredibly useful to create frequencies to show all the possible values in a raw variable with PROC FREQ. One pertinent option in the coding, MISSING, enables a complete list of the various types of missing values.

```
PROC FREQ DATA=SEXRISK;
       TABLES SEX2B / MISSING;
RUN;
```

Table 2. Frequency of binary variable SEX2B (Is this sex partner a man?).

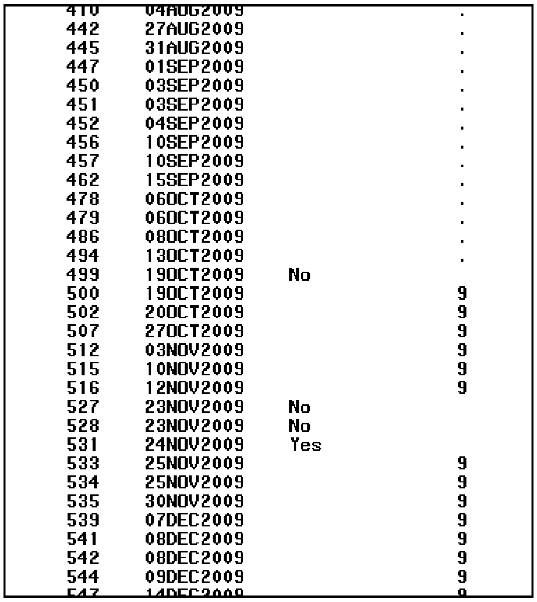
Is sex partner a man?				
SEX2B	Frequency	Percent	Cumulative Frequency	Cumulative Percent
.	1	0.33	1	0.33
No	9	2.99	10	3.32
Yes	121	40.20	131	43.52
Refuse to Answer	1	0.33	132	43.85
9	169	56.15	301	100.00

From Table 2 we can tell that SEX2B is a binary variable where respondents answered yes or no if their only sex partner was a male. This also shows that 169 individuals were skipped out of this question for reasons of having no sex partner, refusing to answer SEX1, or they reported having more than one sex partner. These reasons for being skipped are revealed by coding for the skip logic in the DATA step.

SORTING THE DATA BY A COMMON VARIABLE

It is also pertinent to do a PROC SORT with a certain variable to uncover when survey corrections or changes were made. In this project sorting by the date variable, TODAY, was a necessary tool. Doing so will easily reveal when there was a change in the survey programming.

```
PROC SORT DATA=SEXRISK;
       BY TODAY;
RUN;
PROC PRINT DATA=SEXRISK;
       VAR TODAY SEX15B;
RUN;
```



```
     410    04AUG2009                       .
     442    27AUG2009                       .
     445    31AUG2009                       .
     447    01SEP2009                       .
     450    03SEP2009                       .
     451    03SEP2009                       .
     452    04SEP2009                       .
     456    10SEP2009                       .
     457    10SEP2009                       .
     462    15SEP2009                       .
     478    06OCT2009                       .
     479    06OCT2009                       .
     486    08OCT2009                       .
     494    13OCT2009                       .
     499    19OCT2009    No
     500    19OCT2009                       9
     502    20OCT2009                       9
     507    27OCT2009                       9
     512    03NOV2009                       9
     515    10NOV2009                       9
     516    12NOV2009                       9
     527    23NOV2009    No
     528    23NOV2009    No
     531    24NOV2009    Yes
     533    25NOV2009                       9
     534    25NOV2009                       9
     535    30NOV2009                       9
     539    07DEC2009                       9
     541    08DEC2009                       9
     542    08DEC2009                       9
     544    09DEC2009                       9
     547    14DEC2009                       9
```

Figure 2. PROC PRINT output of raw variable SEX15B when sorted by TODAY.

Raw variable SEX15B has all missing values until 19Oct2009. This question was not added until this date, which should be denoted with another customized code in the final frequency rather than having many unexplained missing values. This ensures the principal investigator is reminded of the change as well as others outside the study looking at the data.

```
IF TODAY LT "19OCT2009"D THEN SEX15B=.X; *.X=Survey change: Question added;
ELSE DO; *Continue with coding;
```

DETAILING OUTPUT WITH PROC FORMAT

All of these customized codes are formatted in a PROC FORMAT statement containing a brief description of the skip or in the above case, survey change. This clearly explains the customized codes in any output.

```
PROC FORMAT;
       VALUE RISKBIN 0="0:No"
                     1="1:Yes"
                     .A=".A:No sex partners"
                     .B=".B:No male sex partners"
                     .C=".C:No vaginal/anal male sex partners"
                     .R=".R:Refuse to answer"
                     .X=".X:Survey change: Question added";
```

CREATING DERIVED VARIABLES IN THE DATA STEP

IF-THEN-ELSE SKIP LOGIC CODING

IF-THEN-ELSE statements are heavily used in the DATA step to code for the skip logic pattern. For instance the first question in sexual activities asks how many sex partners the respondent has had in their past (SEX1). If R replies zero, all remaining questions are automatically scored “9”, “99”, or “999” indicating these questions were skipped. If all the participants are skipped correctly the coding below would be unnecessary, but there are two main issues why this coding is required. First, the nines do not distinguish which skip it refers to. There is no specific information for the PI why these individuals did not answer the question. Second, survey programming errors do occur and the DATA step coding ensures the proper skip patterns are being used. If there are any aberrations then the coding for every possible scenario with IF-THEN-ELSE statements will easily detect them. The following statements were coded to inform the PI individuals are being skipped because they had no previous sex history or no male sex partners.

```
IF SEX1=0 THEN DO; *0 sex partners;
       PARTNERS=0;
       MALEP=.A; *.A=No sex partners;
       FEMALEP=.A;
END;
ELSE IF SEX1=998 THEN DO;
       PARTNERS=.R; *.R=Refuse to answer;
       MALEP=.R;
       FEMALEP=.R;
END;
ELSE IF SEX1=1 THEN DO; *One sex partner;
       PARTNERS=1;
       IF SEX2B=0 THEN DO;
              MALEP=0;
              FEMALEP=1;
       END;
       ELSE IF SEX2B=1 THEN DO; *Yes, one male partner;
              MALEP=1;
              FEMALEP=0;
       END;
END;
ELSE IF SEX1 GT 1 THEN DO; *More than one sex partner;
       PARTNERS=SEX1;
       MALEP=SEX2;
       FEMALEP=SEX2-MALEP;
END;
```

Listed above are three derived variables, the total number of sex partners (PARTNERS), number of male sex partners (MALEP), and number of female sex partners (FEMALEP). The ".A" will be displayed whenever SEX1=0 for MALEP and FEMALEP which will be given a format (using PROC FORMAT) to be more meaningful in the output. If a respondent has more than one male sex partner, then finding female sex partners it is a simple subtraction of the number of male partners from the total number of sex partners.

Another variable likely to be of interest to a principal investigator is whether or not the respondent has had unprotected sex (USEX). This derived variable requires affirmative information on [type A] or [type B] sex with at least one man and involves all variables in Table 1. For instance from Figure 1, if SEX1 is "1" (i.e. only one sex partner) and SEX2B is "1" (i.e. yes, the one sex partner from SEX1 is a man) then they go on to answer SEX3B and SEX4B. The following coding in the DATA step is shown for USEX.

```
IF SEX1=0 THEN USEX=.A; *.A=No sex partners;
ELSE IF SEX1=998 THEN USEX=.R;
ELSE IF SEX1=1 THEN DO; *One sex partner;
       IF SEX2B=0 THEN USEX=.B; *.B=No male sex partners;
       ELSE IF SEX2B=8 THEN USEX=.R;
       ELSE IF SEX2B=1 THEN DO; *Yes, one male partner;
              IF SEX3B=0 AND SEX4B=0 THEN USEX=.C; *.C=No [type A]/[type B] sex
              partners;
              ELSE IF SEX3B=8 AND SEX4B=8 THEN USEX=.R;
              ELSE IF SEX3B NE 1 AND SEX4B NE 1 THEN USEX=.C;
              ELSE IF SEX3B=1 OR SEX4B=1 THEN DO;
                     IF SEX7B=8 THEN USEX=.R;
                     ELSE USEX=SEX7B;
              END;
       END;
END;
ELSE IF SEX1 GT 1 THEN DO; *More than one sex partner;
       IF SEX2=0 THEN USEX=.B; *.B=No male sex partners;
       ELSE IF SEX2=998 THEN USEX=.R;
       ELSE IF SEX2=1 THEN DO; *One male sex partner;
              IF SEX3B=0 AND SEX4B=0 THEN USEX=.C; *.C=No [type A]/[type B] sex
              partners;
              ELSE IF SEX3B=8 AND SEX4B=8 THEN USEX=.R;
              ELSE IF SEX3B NE 1 AND SEX4B NE 1 THEN USEX=.C;
              ELSE IF SEX3B=1 OR SEX4B=1 THEN DO;
                     IF SEX7B=8 THEN USEX=.R;
                     ELSE USEX=SEX7B;
              END;
       END;
       ELSE IF SEX2 GT 1 THEN DO; *More than one male sex partner;
              IF SEX3=0 AND SEX4=0 THEN USEX=.C;
              ELSE IF SEX3=998 AND SEX4=998 THEN USEX=.R;
              ELSE IF SEX3 GE 1 OR SEX4 GE 1 THEN DO;
                     IF SEX7B=8 THEN USEX=.R;
                     ELSE USEX=SEX7B;
              END;
       END;
END;
```

USEX is a binary outcome; either the respondent has unprotected sex or does not (or is not applicable due to being skipped out of the question). Another derived variable could count how many unprotected male sex partners the respondent by replacing SEX7B with the count variable SEX7 to the code above. The code can easily be manipulated to produce either a binary or count variable. Whichever is desired is determined by the research question. The USEX variable is one of the most intricate derived variables with its complex skip pattern but it is an important variable likely to be requested by the principal investigator when researching HIV prevention. In this project the respondents undergo biannual interviews at the research site which includes an HIV blood test at each visit to determine if they are recently positive. Combining the knowledge of a respondent's risky sex behavior such as unprotected sex and their HIV status is significant in identifying who is at risk for transmitting HIV or being infected.

ARRAY SKIP LOGIC CODING

Often in the survey, many variables are coded the same missing code. This can usually be done with an IF-THEN-ELSE statement but when handling more than a handful of variables this procedure becomes immediately time consuming and tiresome. Variables SEX9A to SEX9W asks reasons why the respondent did not use a condom with their primary partners.

```
IF SEX1=1 THEN DO;
       IF SEX2B=0 THEN DO;
              SEX9A=.B;
              SEX9B=.B;
              ...
              SEX9W=.B;
       END;
       ELSE IF SEX2B=1 THEN DO;
              IF SEX3B NE 1 AND SEX4B NE 1 THEN DO;
                     SEX9A=.C;
                     SEX9B=.C;
                     ...
                     SEX9W=.C;
              END;
       END;
END;
```

A more efficient way is to use ARRAY statements which call a vector of variables rather than each individual variable. This tool is heavily utilized in the sexual activities section due to its complex skip nature and the numerous raw variables that need to be coded.

```
ARRAY SKIPS(*) SEX9A--SEX9W;
DO I=1 TO DIM(SKIPS);
       IF SEX1=0 THEN SKIPS(I)=.A; *.A=No sex partners;
       ELSE IF SEX1=998 THEN SKIPS(I)=.R;
       ELSE IF SEX1=1 THEN DO;
              IF SEX2B=0 THEN SKIPS(I)=.B; *.B=No male sex partners;
              ELSE IF SEX2B=8 THEN SKIPS(I)=.R;
              ELSE IF SEX2B=1 THEN DO;
                     IF SEX3B NE 1 AND SEX4B NE 1 THEN SKIPS(I)=.C; *.C=No [type
                     A]/[type B] male sex partners;
                     ELSE IF SEX3B=1 AND SEX4B=1 THEN DO;
                            IF USEX=0 THEN SKIPS(I)=.D; *.D=Used condom 100% of the
                            time;
                     END;
              END;
       END;
       ELSE IF SEX1 GT 1 THEN DO;
              IF SEX2=0 THEN SKIPS(I)=.B;
              ELSE IF SEX2=998 THEN SKIPS(I)=.R;
              ELSE IF SEX2=1 THEN DO;
                     IF SEX3B NE 1 AND SEX4B NE 1 THEN SKIPS(I)=.C; *.C=No [type
                     A]/[type B] male sex partners;
                     ELSE IF SEX3B=1 AND SEX4B=1 THEN DO;
                            IF USEX=0 THEN SKIPS(I)=.D; *.D=Used condom 100% of the
                            time;
                     END;
              END;
              ELSE IF SEX2 GT 1 THEN DO;
                     IF SEX3 IN (0,998) AND SEX4 IN (0,998) THEN SKIPS(I)=.C;
                     ELSE IF SEX3 GE 1 AND SEX4 GE 1 THEN DO;
                            IF USEX=0 THEN SKIPS(I)=.D;
                     END;
              END;
       END;
END;
```

The code above illustrates how ARRAY statements can be powerful when dealing with complex skip patterns. If we were to use IF-THEN-ELSE statements then the code above would be expanded with having to individually code the 23 variables in SEX9A to SEX9W. A general rule of thumb is to keep programs as concise as possible to avoid mistakes. Not incorporating ARRAY statements leads to lengthy programs which do not easily nor quickly convey the skip logic to another programmer. Another good practice is to include as many commented statements (in green above) as you feel necessary so anyone, including yourself, can effortlessly follow the program.

CHECKING THE CODING AGREES WITH SURVEY SKIP LOGIC

If we were to incorrectly apply the skip logic to a derived variable it would easily be spotted with a simple PROC PRINT of the raw variables and the resulting derived variable. For example, USEX (unprotected sex) is used to determine if subjects are at risk for transmitting HIV. In the following PROC PRINT the variables preceding USEX are those involved in the skip pattern.

```
PROC PRINT DATA=SEXRISK (OBS=6);
       VAR SUBJECT SEX1 SEX2 SEX3 SEX4 SEX2B SEX3B SEX4B SEX7B SEX7 USEX;
RUN;
```

Table 3. PROC PRINT output of first six subjects demonstrates how to check for coding accuracy for the derived variable, USEX.

SUBJECT	SEX1	SEX2	SEX3	SEX4	SEX2B	SEX3B	SEX4B	SEX7B	SEX7	USEX
1	RA	RA	RA	RA	RA	RA	RA	RA	RA	.R:Refuse to answer
2	2	0	999	999	9	9	9	9	999	.B:No male sex partners
3	2	2	2	0	9	9	9	No	999	0:No
4	1	999	999	999	Yes	No	No	9	999	.C:No vaginal/anal male sex partners
5	0	999	999	999	9	9	9	9	999	.A:No sex partners
6	2	2	2	2	9	9	9	Yes	2	1:Yes

As mentioned earlier, the practice of renaming variables with ARRAY statements makes the original variables accessible to verify that the DATA step coding agrees with the survey skip logic. If we were to code using raw variables rather than renaming them, Table 3 would show customized formats instead of nines. This would make it difficult to determine if any mistakes were made in the derived variable creation.

To verify if the coding agrees with the skip logic, analyze subjects with every possible scenario to see if it makes sense. From Table 3, subjects 1, 2, 4, and 5 were not applicable for the derived variable, unprotected sex practice (USEX). Subject 1 refused to answer each question, subject 2 reported no male sex partners (SEX2=0), subject 4 had one male sex partner but did not report having sex with that partner, and subject 5 had no sex partners. Subjects 3 and 6 did have information for USEX. Subject 3 said she had [type A] and [type B] sex with each male partner (SEX2=2, SEX3=2) and that they did use a condom (SEX7B=0) thus USEX was determined to be “0” (i.e. no, they did not have unprotected sex). Subject 6 was determined to be USEX=1 (i.e. yes, she did have unprotected sex) because she reported having [type A] and [type B] sex with each male partner (SEX3=2, SEX4=2) and did not use condoms with either partner (SEX7=2). These are all examples of the DATA step coding agreeing with the skip logic. However, if any of these subjects had USEX disagree with their responses then we would need to debug the programming to see what went awry.

CONCLUSION

By using the DATA step in this way, a comprehensive PROC FREQ can be created. For example, using information from the raw variables in Table 1 we can derive whether the respondent had unprotected sex as the variable USEX where USEX=1 indicates “yes, respondent had unprotected sex” and USEX=0 indicates “no unprotected sex”. This is vital information for an investigator to know when studying HIV-risk behaviors. Using the previously mentioned variables, a final frequency likely to be presented crosses the respondent’s HIV status to the number of participants who were determined to have unprotected sex. This expresses a general idea of HIV risk.

```
PROC FREQ DATA=SEXRISK;
      TABLES USEX*HIVSTAT / MISSING NOPERCENT NOROW NOCOL;
RUN;
```

Table 4. Frequency of those who had unprotected sex crossed with respondent HIV status.

Table of usex by HIVSTAT			
usex(Any unprotected vaginal/anal sex)	**HIVSTAT(HIV status)**		
Frequency	**Negative**	**Positive**	**Total**
.A:No sex partners	36	51	87
.B:No male sex partners	9	3	12
.C:No vaginal/anal male sex partners	9	6	15
.R:Refuse to answer	0	1	1
0:No	27	32	59
1:Yes	69	58	127
Total	150	151	301

Using customized codes such as ".A", ".B", or ".C" to distinguish different levels of skips is useful when presenting the data to a principal investigator. This enables the PI to approach the analyses from different perspectives. They may choose to re-code ".A:No sex partners" as "0" for USEX. This is essentially saying that if the respondent reported no sex partners at all, then it is equivalent to reporting no unprotected sex. An alternate analysis may look at only those who had male partners. They would leave the 87 individuals coded as ".A" and the 12 as ".B" but may want to re-code the 15 women who reported no [type A] or [type B] sex with their male partners as "0" for USEX. Then the investigator will approach the research question as only looking at women with male partners, how many of those had unprotected sex. This frequency table also exhibits the data in such a way that is readily accessible to researchers outside the SHADOW study. The customized missing formats clearly outline how participants navigated through the survey.

Data cleaning lays the necessary groundwork to ensure what you are presenting is true and without survey errors. Although it is not a subject shared too often in either my previous coursework or other traditional statistics classes, it is a topic that holds a great weight to the results of any analysis. Using these crucial steps and other processes like them, principal investigators may confidently conduct research assessing the health of these 301 women in San Francisco.

REFERENCES

Cody, Ron. 2007. *Learning SAS® by Example: A Programmer's Guide*. Cary, NC: SAS Institute Inc.

Cody, Ron. 2008. *Cody's Data Cleaning Techniques Using SAS®, Second Edition*. Cary, NC: SAS Institute Inc.

SAS Institute Inc. SAS OnlineDoc® 9.2
http://support.sas.com/cdlsearch?ct=80000

ACKNOWLEDGEMENTS

The authors would like to thank the entire SHADOW team for supporting their participation in SAS Global Forum.

This study was supported by NIH R01 DA15605.

CONTACT INFORMATION

The authors value your comments and questions. You can contact Isabel via e-mail at perryih@gmail.com or phone at (415) 328-5774.

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

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