SESUG Paper 260-2018

Using SAS® to create HTML codebooks and more!

Ethan Ritchie, RTI International

ABSTRACT

With some basic knowledge of SAS® and HTML, you can create custom documents using your data and associated documentation by using a file statement and put statements to write to an HTML file. This paper provides an example of creating an organized and formatted HTML codebook based on a national education survey and the resulting data. The codebook is complete with variable labels, survey question wording, notes, value labels, and summary statistics. The process is dynamic (i.e., variable names do not need to be specified and the code does not need editing when documentation, variables, or values change). This allows the code to be ported between projects with minimal modification.

INTRODUCTION

The process described in this paper includes all of the pieces needed to go from data and metadata to a codebook complete with frequencies and summary statistics. The sample data used in this paper are from the High School Longitudinal Study of 2009 (HSLS:09) conducted by the National Center for Education Statistics (NCES). The data are publicly available online the NCES Education Data Analysis Tool (https://nces.ed.gov/edat/).

CODEBOOK INPUTS

For our example, we will have three variables, an ID variable, a continuous variable, and a discrete variable. Creating a codebook requires three inputs.

1. The data to be described by the codebook
2. Information about each variable (labels, descriptions, variable types, etc.) PROC PRINT output for example variable-level metadata are displayed below in Output 1.
3. Information about the values corresponding to each variable (value labels, value order, etc.) PROC PRINT output for example value-level metadata are displayed below in Output 2.

Output 1. PROC PRINT output for example variable-level metadata

Variable	VariableLabel	Description	AppliesTo	VariableOrder	VariableType
STU_ID	Student ID	Student identifier assigned for all base year eligible students (including respondents, nonrespondents, and questionnaire incapable students). IDs randomly assigned from 10001 to 35206 across all students.		1	X
X4CLGAPPNUM	X4 Number of colleges applied to when first applied	Indicates the number of colleges the respondent applied to when first applying. Those who never applied or registered and never attended a postsecondary institution are coded as 0. Information on the number of applications submitted is not available for small number of sample members who indicated in the HSLS:09 2013 Update or second follow-up interview that they never applied or registered but indicated that they attended. This variable conveys information related to respondents' first round of college applications; respondents' first round of college applications may have occurred while in high school, or some time after high school.	All second follow-up respondents.	2	C
X4PS1LEVEL	X4 First postsecondary institution level	Indicates the level of first postsecondary institution attended after high school. This variable indicates the level of the first postsecondary institution attended after high school. It is based on IPEDS variable ICLEVEL. To identify public 4-year colleges that primarily award subbaccalaureate credentials, see X4PS1SECTOR. See X4PS1 for a definition of the first institution.	Second follow-up respondents who ever enrolled in a postsecondary institution after high school (i.e., X4EVRATNDCLG = 1)	3	D

Output 2. PROC PRINT output for example value-level metadata

Variable	Value	ValueLabel	ValueOrder
X4CLGAPPNUM	-9	Missing	1
X4CLGAPPNUM	-8	Unit non-response	2
X4CLGAPPNUM	1	Continuous	3
X4PS1LEVEL	-9	Missing	1
X4PS1LEVEL	-8	Unit non-response	2
X4PS1LEVEL	-7	Item legitimate skip/NA	3
X4PS1LEVEL	1	4-year	4
X4PS1LEVEL	2	2-year	5
X4PS1LEVEL	3	Less than 2-year	6

FREQUENCIES AND SUMMARY STATISTICS

In addition to the inputs described in the previous section, we will need to calculate frequencies and summary statistics for continuous and discrete variables. The first step in accomplishing that task is to generate two macro variables; one containing a list of continuous variables and the other containing a list of discrete variables. We will then use those lists to dynamically generate appropriate summary statistics for each variable. The code to generate the two macro variables uses the dataset shown in Output 1 and is included below:

```
data _NULL_;
   length ContinuousText $32000 DiscreteText $32000;
   set VariableInfo end=finally;
   retain ContinuousText '' DiscreteText;
   if VariableType = 'C' then
      ContinuousText = catx(' ',ContinuousText,Variable);
   if VariableType = 'D' then
      DiscreteText = catx(' ',DiscreteText,Variable);
   if finally then do;
      call symput('ContinuousList',ContinuousText);
      call symput('DiscreteList',DiscreteText);
   end;
run;
```

Next, we need to temporarily recode continuous variables so that we can create a frequency table of reserve codes (negative values in the HSLS data) with one row for all values greater than or equal to zero. The code uses one of the previously created macro variables and is shown below:

```
data hsls_freqs_prep;
   set hsls;
   array ContinuousVars &ContinuousList;
   do over ContinuousVars;
      if ContinuousVars >= 0 then ContinuousVars = 1;
   end;
 run;
```

Once those preparatory steps are complete, we are ready to compute frequencies and summary statistics. First, we will use the code below to loop over the list of discrete and recoded continuous variables and generate frequencies of each value. PROC PRINT output of the resulting dataset is shown below in Output 3.

```
%macro getFreqs;
data freqs;
   set _NULL_;
run;
%do i=1 %to %sysfunc(countw(&DiscreteList &ContinuousList));
%let var = %scan(&DiscreteList &ContinuousList,&i);
proc freq data=hsls_freqs_prep noprint;
   tables &var / out=temp;
run;
data temp;
   set temp;
   rename &var = Value;
   Variable = "&var";
run;
data freqs;
   length Variable $50;
   set freqs temp;
run;
%end;
%mend;
%getFreqs;
```

Next, we will use the code below to loop over the list of continuous variables (not recoded this time) and generate summary statistics for each variable. PROC PRINT output of the resulting dataset is shown below in Output 4.

```
%macro getMeans;
data means;
   set _NULL_;
run;
%do i=1 %to %sysfunc(countw(&ContinuousList));
%let var = %scan(&ContinuousList,&i);
proc means data=hsls noprint;
   var &var;
   where &var >= 0;
   output out=temp mean=Mean std=StDev min=Min max=Max;
run;
data temp;
   set temp;
   Value = 1;
   Variable = "&var";
   drop _TYPE_ _FREQ_;
run;
data means;
   length Variable $50;
   set means temp;
run;
%end;
%mend;
%getMeans;
```

We then merge everything together using the code below to create the final input file for the codebook. PROC CONTENTS output of the resulting dataset is shown below in Output 5.

```
proc sort data=ValueInfo; by Variable Value;
proc sort data=freqs; by Variable Value;
proc sort data=means; by Variable Value;
data sumStats;
   length Variable $50;
   merge ValueInfo freqs means;
   by Variable Value;
run;


proc sort data=VariableInfo; by Variable;
data codebookInputs;
   length Variable $50;
   merge VariableInfo sumStats;
   by Variable;
   percent = PERCENT/100;
   format COUNT comma9.0 PERCENT percent9.1 Min comma9.0 Max comma9.0
      Mean comma9.2 StDev comma9.2;
run;
```

Output 3. PROC PRINT output for frequencies

Variable	Value	COUNT	PERCENT
X4CLGAPPNUM	-9	1506	6.4077
X4CLGAPPNUM	-8	6168	26.2435
X4CLGAPPNUM	1	15829	67.3488
X4PS1LEVEL	-9	97	0.4127
X4PS1LEVEL	-8	6168	26.2435
X4PS1LEVEL	-7	4281	18.2147
X4PS1LEVEL	1	8601	36.5953
X4PS1LEVEL	2	4008	17.0531
X4PS1LEVEL	3	348	1.4807

Output 4. PROC PRINT output for summary statistics

Variable	Mean	StDev	Min	Max	Value
X4CLGAPPNUM	3	2	0	11	1

Output 5. PROC CONTENTS output for combined information

Variables in Creation Order						
#	Variable	Type	Len	Format	Informat	Label
1	Variable	Char	50	$11.	$11.	Variable
2	VariableLabel	Char	51	$51.	$51.	VariableLabel
3	Description	Char	645	$645.	$645.	Description
4	AppliesTo	Char	120	$120.	$120.	AppliesTo
5	VariableOrder	Num	8	BEST.		VariableOrder
6	VariableType	Char	1	$1.	$1.	VariableType
7	Value	Num	8	BEST.		Value
8	ValueLabel	Char	23	$23.	$23.	ValueLabel
9	ValueOrder	Num	8	BEST.		ValueOrder
10	COUNT	Num	8	COMMA9.		Frequency Count
11	PERCENT	Num	8	PERCENT9.1		Percent of Total Frequency
12	Mean	Num	8	COMMA9.2		Mean
13	StDev	Num	8	COMMA9.2		Standard Deviation
14	Min	Num	8	COMMA9.		Minimum
15	Max	Num	8	COMMA9.		Maximum

GENERATING THE CODEBOOK

Once all the necessary information is compiled, generating a codebook is a matter of using a file statement, by-group processing and put statements to create an HTML file. Note that this paper does not go into detail on writing HTML code. The scope is limited to the use of SAS to create HTML files. First, we sort the codebook input dataset by variable order and value order in preparation for by-group processing. Then, we specify a file path for the HTML output. The code for this part is below:

```
proc sort data=codebookInputs; by VariableOrder ValueOrder;
data _NULL_;
   set codebookInputs;
   by VariableOrder ValueOrder;
   file "somePath/filename.html";
```

The next step is to output the style specifications for the document. This needs to be included once at the beginning of the document, so it is written to the file at the beginning of the DATA step.

```
   if _N_ = 1 then do;
   put '<style type="text/css"> <!-- .tab { margin-left: 40px; } -->
      table {
         border-collapse: collapse;
         border: 1px solid black;
         word-wrap: break-word;
         table-layout: auto;
         width: 80%;
      }
      td {border: 1px solid black;}
      </style>';
   end;
```

We then write variable-level information to the file along with the heading for the table of frequencies for continuous and discrete variables. The code is shown below:

```
if first.VariableOrder then do;
   put "<b>" VariableOrder " - " Variable "</b><br><br>";
   put "<b>Label:</b> " VariableLabel "<br>";
   put "<b>Description:</b> " Description "<br>";
   if not missing(AppliesTo) then
      put "<b>Applies to:</b> " AppliesTo "<br>";
   if VariableType in ('C','D') then
      put '<br><table>
         <tr>
            <td style="background-color: #D0D0D0;
                       text-align:center">
                <strong> Value </strong>
            </td>
            <td style="background-color: #D0D0D0;
                       padding-left:5px">
                <strong> Label </strong>
            </td>
            <td style="background-color: #D0D0D0;
                       text-align:center">
                <strong> Count </strong>
            </td>
            <td style="background-color: #D0D0D0;
                       text-align:center">
                <strong> Percent </strong>
            </td>
         </tr>';
end;
```

Next, we write a row in the frequency table for each variable-value combination. This is done only for continuous and discrete variables. The code is shown below:

```
if VariableType in ('C','D') then put
   '<tr>
      <td style="width:10%; text-align:center">'
         Value
     '</td>
      <td style="width:50%; padding-left:5px">'
         ValueLabel
     '</td>
      <td style="width:10%; text-align:right">'
         COUNT
     '</td>
      <td style="width:10%; text-align:right">'
         PERCENT
     '</td>
    </tr>';
```

On the last record for each variable we add a table of summary statistics if the variable is continuous and add a horizontal line to visually separate each variable in the codebook. The code is shown below:

```
if last.VariableOrder then do;
   if VariableType = 'C' then do;
      put '</table><br>';
      put '<br><table>
         <tr>
            <td style="background-color: #D0D0D0;
                       text-align:center">
               Minimum
            </td>
            <td style="background-color: #D0D0D0;
                       text-align:center">
               Maximum
            </td>
            <td style="background-color: #D0D0D0;
                       text-align:center">
               Mean
            </td>
            <td style="background-color: #D0D0D0;
                       text-align:center">
               Standard Deviation
            </td>
         </tr>';
       put '<tr>
            <td style="wdth:20%; text-align:center">'
               Min
           '</td>
            <td style="width:20%; text-align:center">'
               Max
           '</td>
            <td style="width:30%; text-align:center">'
               Mean
           '</td>
            <td style="width:30%; text-align:center">'
               StDev
           '</td>
         </tr>';
   end;
   if VariableType in ('C','D') then put '</table>';
   put '<br><hr style="width:"100%"; color:#000000"><br>';
end;
```

The resulting codebook is shown below in **Output 6**.

Output 6. FILE statement output - HTML codebook

1 - STU_ID

Label: Student ID
Description: Student identifier assigned for all base year eligible students (including respondents, nonrespondents, and questionnaire incapable students). IDs randomly assigned from 10001 to 35206 across all students.

---

2 - X4CLGAPPNUM

Label: X4 Number of colleges applied to when first applied
Description: Indicates the number of colleges the respondent applied to when first applying. Those who never applied or registered and never attended a postsecondary institution are coded as 0. Information on the number of applications submitted is not available for small number of sample members who indicated in the HSLS:09 2013 Update or second follow-up interview that they never applied or registered but indicated that they attended. This variable conveys information related to respondents' first round of college applications; respondents' first round of college applications may have occurred while in high school, or some time after high school.
Applies to: All second follow-up respondents.

Value	Label	Count	Percent
-9	Missing	1,506	6.4%
-8	Unit non-response	6,168	26.2%
1	Continuous	15,829	67.3%

Minimum	Maximum	Mean	Standard Deviation
0	11	2.55	2.47

---

3 - X4PS1LEVEL

Label: X4 First postsecondary institution level
Description: Indicates the level of first postsecondary institution attended after high school. This variable indicates the level of the first postsecondary institution attended after high school. It is based on IPEDS variable ICLEVEL. To identify public 4-year colleges that primarily award subbaccalaureate credentials, see X4PS1SECTOR. See X4PS1 for a definition of the first institution.
Applies to: Second follow-up respondents who ever enrolled in a postsecondary institution after high school (i.e., X4EVRATNDCLG = 1)

Value	Label	Count	Percent
-9	Missing	97	0.4%
-8	Unit non-response	6,168	26.2%
-7	Item legitimate skip/NA	4,281	18.2%
1	4-year	8,601	36.6%
2	2-year	4,008	17.1%
3	Less than 2-year	348	1.5%

CONCLUSION

Using SAS to write HTML code has many practical applications beyond codebooks. Data collection reports and derived variable progress reports are just a couple of the additional places where this method has been implemented. The process is dynamic (i.e., variable names do not need to be specified and the code does not need editing when documentation, variables, or values change). This allows the code to be ported between projects with minimal modification.

CONTACT INFORMATION

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

Ethan Ritchie
RTI International
(919) 541-7036
eritchie@rti.org