Paper BB-14

PROC_CODEBOOK, Automating the Review and Documentation of SAS® Files

James Terry, Carolina Population Center
Kim Chantala, Dept. of Health Behavior and Health Education
University of North Carolina at Chapel Hill

ABSTRACT

If a SAS file has variable names and formats for categorical variables, a highly recommended practice, the PROC_CODEBOOK macro can be called to produce a comprehensive, well formatted and easy to read codebook. In the heading the codebook will provide codebook title(s), file name, file label, and date created, number of observations and number of variables and file organization. The body of the codebook will provide the variable name, label, type, format and length, the mean, the range of values, frequency category, number and percent. Optional footnotes provide additional documentation. In addition, a codebook warning report is produced for categories that are not used, for variables with all missing data and variables that are constant.

To produce a codebook, the user provides one or two titles for the codebook (title1 and title2), file organization (%let organization = user defined) and calls the macro, it is that easy.

BACKGROUND

In research oriented departments such as the Carolina Population Center and the Dept. of Health Behavior and Health Education at the University of North Carolina at Chapel Hill, codebooks are often used to document survey based statistical files. However, it has never been easy to produce these. They have always been produced with a great deal of programming or manually produced. The authors have for a long time wanted to automate codebook generation, but struggled with how to distinguish a categorical variable from a continuous variable. But it turns out that a simple programming rule, if a variable has a user format, it is categorical, if not, it is continuous, solved this problem and enabled the development of an easy-to-use, general purpose codebook program.

WHAT IS A CODEBOOK?

Traditionally, codebooks are used to document what the codes stand for when used in categorical variables, i.e. 1 for male and 2 for female or 1 for always, 2 for sometimes, 3 for usually and 4 for always. However, you should think of this "PROC_Codebook" as a super duper PROC CONTENTS. Not only does it tell you what variables are in the file, it provides significant information on the contents of the variables themselves. For categorical variables, numeric or character, it will give you the codes, the definition of the codes, and the frequency and percent distribution of the codes. For numeric continuous variables, it provides the mean, range and frequency of non-missing values. Of course it gives you the normal PROC CONTENTS information such as data set name, number of observations, date created, data set label, variable names, variable label and variable type.

CODEBOOK BENEFITS

PROC_CODEBOOK provides both an easy to use and comprehensive way to document and review not only files you create, but also files that you receive, especially if they do not have good documentation. The highlight of PROC_CODEBOOK is its ability to document and review categorical variables. Are the distributions reasonable? Are there any undocumented codes? Is the data all missing for certain categories? For continuous variables, is the data all missing? Should there be any missing data? Is the mean reasonable? Does the range indicate the possible presence of outliers? It provides a quick way to look for unexpected or embarrassing results while producing first-rate documentation.

USING PROC_CODEBOOK

The PROC_CODEBOOK macro and Complete Up-to-Date Documentation is available at:

http://www.cpc.unc.edu/research/tools/data_analysis/proc_codebook

To produce a codebook, the user provides one or two titles for the codebook (title1 and title2), file organization (%let organization = user defined) and calls the macro:

proc_codebook (lib,file1,fmtlib,pdffile);

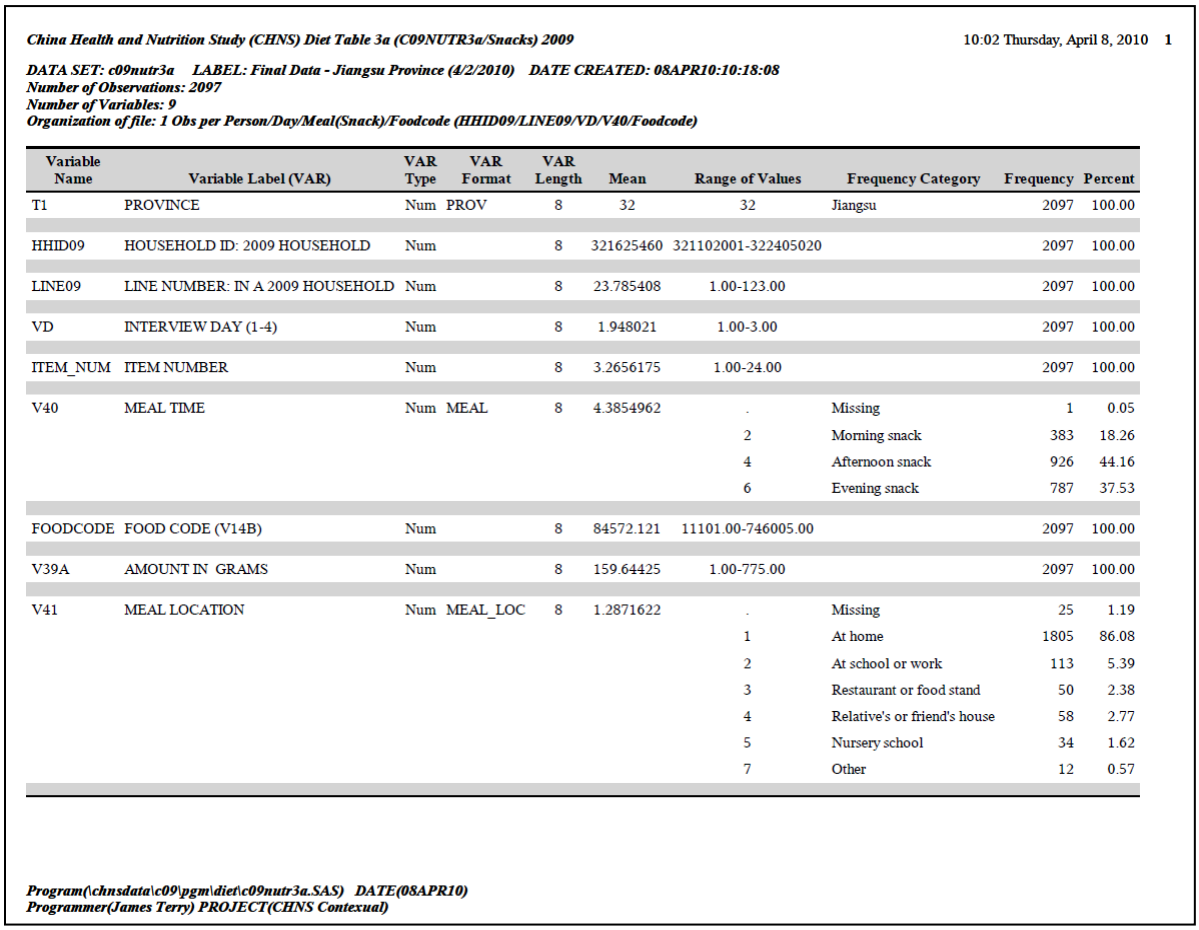
- lib is the libname for the file to be "coded"
- file1 is the name of the file to be "coded"
- Fmtlib is the location of the format library, usually "library" or "work:
- Pdffile is the fully qualified name of the codebook (.pdf) file

Following is a sample program for producing a codebook:

```
*-------------------------------------------------------------------------------;
footnote1 "Program(\MyProject\c09pgm3a.SAS)   DATE(&sysdate) ";
footnote2 "Programmer(John Doe) PROJECT(SESUG)";
run;
libname c09 "d:\SAS_data\C23";
libname library 'f:\masters_07'; run;

title1 China Health and Nutrition Study Diet Table 3a (C09NUTR3a/Snacks) 2009;

%let organization = 1 Obs per Person/Day/Meal(Snack)/Foodcode
(HHID09/LINE09/VD/V40/Foodcode);

data order;
  length name $ 32;
input name $ order;
datalines;
T1       1
HHID09   2
LINE09   3
VD       4
ITEM_NUM 5
V40      6
FOODCODE 7
V39A     8
V41      9
; run;

%proc_codebook (lib=c09,file1=c09nutr3a,fmtlib=library,pdffile=c:\mycodebooks\C23NUTR3a.pdf);
run;
*-------------------------------------------------------------------------------;
```

You could also have a title2 if the codebook needs more description. Organization is a free form way to indicate key variables and file organization. The "order" file (work.order) is optional. It is a two variable file and if it is found it will determine the order that the variables appear in the codebook, otherwise, they appear as they would in a PROC CONTENTS. It is used here so that they appear in the order they appear in the questionnaire. This file can be produced many ways.

Following is the codebook, this program produced:



China Health and Nutrition Study (CHNS) Diet Table 3a (C09NUTR3a/Snacks) 2009 10:02 Thursday, April 8, 2010 1

DATA SET: c09nutr3a LABEL: Final Data - Jiangsu Province (4/2/2010) DATE CREATED: 08APR10:10:18:08
Number of Observations: 2097
Number of Variables: 9
Organization of file: 1 Obs per Person/Day/Meal(Snack)/Foodcode (HHID09/LINE09/VD/V40/Foodcode)

Variable Name	Variable Label (VAR)	VAR Type	VAR Format	VAR Length	Mean	Range of Values	Frequency Category	Frequency	Percent
T1	PROVINCE	Num	PROV	8	32	32	Jiangsu	2097	100.00
HHID09	HOUSEHOLD ID: 2009 HOUSEHOLD	Num		8	321625460	321102001-322405020		2097	100.00
LINE09	LINE NUMBER: IN A 2009 HOUSEHOLD	Num		8	23.785408	1.00-123.00		2097	100.00
VD	INTERVIEW DAY (1-4)	Num		8	1.948021	1.00-3.00		2097	100.00
ITEM_NUM	ITEM NUMBER	Num		8	3.2656175	1.00-24.00		2097	100.00
V40	MEAL TIME	Num	MEAL	8	4.3854962	.	Missing	1	0.05
						2	Morning snack	383	18.26
						4	Afternoon snack	926	44.16
						6	Evening snack	787	37.53
FOODCODE	FOOD CODE (V14B)	Num		8	84572.121	11101.00-746005.00		2097	100.00
V39A	AMOUNT IN GRAMS	Num		8	159.64425	1.00-775.00		2097	100.00
V41	MEAL LOCATION	Num	MEAL_LOC	8	1.2871622	.	Missing	25	1.19
						1	At home	1805	86.08
						2	At school or work	113	5.39
						3	Restaurant or food stand	50	2.38
						4	Relative's or friend's house	58	2.77
						5	Nursery school	34	1.62
						7	Other	12	0.57

Program(\chnsdata\c09\pgm\diet\c09nutr3a.SAS) DATE(08APR10)
Programmer(James Terry) PROJECT(CHNS Contexual)

SUMMARY

The PROC_CODEBOOK macro becomes a super proc as it combines the output of PROC CONTENTS, PROC FORMAT, PROC MEANS, PROC FREQ, PROC REPORT, PROC SORT, and a few data steps all packaged nicely with ODS.

By making codebook generation easy and automated, it becomes practical to incorporate it into your normal routine. For files with many variables, it is not practical to review them with PROC PRINT because of its horizontal nature. The vertical nature of this codebook allows the integration of frequency categories and means and ranges. It is such an easy way to get an overall look at what you have created. It does a great job of highlighting mistakes and unexpected results. By itself, it is not the only way to validate your results, but it is a great way to start. Not to mention, it provides great documentation.

PROC_CODEBOOK provides a great way to "Know Your Data".