ASPP: AUTOMATED SAS PROGRAM PRODUCER

David A. Pyne and Theresa J. Hart, The Upjohn Co.

INTRODUCTION

The construction of a SAS database can be a tedious and time consuming exercise, especially when a large number of variables are involved, and when the output is required to have useful variable labels and formatting. The problems of keeping track of the items associated with a given variable, the label, the location of the values, the input format, the format, and perhaps edit information, have been solved to a large extent by use of data dictionaries in the past few years. (1, 2, 3, 4, 5, 6)

In the case of clinical trials, though, much of the data is also longitudinal in nature; that is, the same items of information are gathered at many time points in the study. Therefore, the same form may be used several times in a given study. An additional consideration is that the same form may also be used in many different studies. We feel that a description of the form of interest and a description of the longitudinal use of the form should be all that is needed, and that the computer should be instructed to perform the tedious, repetitious work of producing the programs necessary to construct the data dictionary and the SAS dataset, and programs to perform basic edits and analysis of the data.

AUTOMATED SAS PROGRAM PRODUCER

The Automated SAS Program Producer (ASPP) is simply a first try at an automated method of handling forms and their longitudinal uses by way of a data dictionary. We have found the system to be a fast, efficient way to create a data dictionary, a SAS dataset for the data, and edits on the data. We also foresee expanding the system to include: 1) specification of additional complicated edit checks; 2) construction of new variables from combinations of existing variables; 3) specification of analyses; 4) reuse of as much information as possible within studies and between studies; 5) the ability to specify different datasets for different groups of variables; and, 6) the ability to specify type of dataset (one observation per patient or many observations according to time periods).

OVERVIEW OF THE SYSTEM

A clinical study begins with an investigator filling out a set of forms at specified time intervals for each patient in his study. The completed forms are then sent to The Upjohn Company, where the data is entered into the computer, one record per form according to the column specifications on the form. Any given study ranges from 10 to 170 forms per patient with 10 to 100 data items per form and approximately 20 to several thousand patients per study.

Our Automated SAS Program Producer (ASPP) starts with two basic components: 1) a set of descriptions of the various forms used in our studies and, 2) a description of the particular study being accessed. Each form description is a separate partitioned dataset member describing one particular form, and includes variable names, their locations in the raw data, formats and labels. The study description specifies which forms are used in a given study and at which time period or sequence each is used.

The basic flow of processing, as shown in Figure 1, proceeds as follows. Starting with a set of form descriptions and a particular study description, we produce a dictionary for that study. Using the dictionary a SAS format library and the raw data, we produce a SAS dataset, one observation per patient. That dataset and the dictionary are then used to output edits on the data. Everything is accomplished by executing three SAS programs, each of which first writes the necessary code for that study onto an external file and then, using %INCLUDE, executes that code.

AN EXAMPLE, STEP 1:

A sample study description is shown in Figure 2. The first piece of information in each row is the name of one of the members in our partitioned dataset of form descriptions. Next is the number (record number) used to identify that form in a study. For instance, in this study, "PTHST" is record 10. Finally, the time periods or sequences at which the form is filled out is listed. In our example, form "PTHIST" is only filled out at sequence 1, and "LBEVAL" is filled out at sequences 2, 3, and 4. A sample of the form description, "PTHIST" is shown in Figure 3. For each variable, the following information is currently specified:

1) The type of edit to be produced (Column 1);
2) The starting column on a form for a variable (Columns 2 and 3);
3) The variable name (Columns 5 to 12);
4) The input format for the variable (Columns 14 to 18);
5) The format or table number for the variable (Columns 19 to 27); and,
6) The label for the variable (Columns 28-67).

When the program to produce the dictionary is executed, the following substitutions will be made in the appropriate places to make the
form unique to this study and time period:

1) Sequence number will be substituted for "%";
2) Form (record) number will be substituted for "111111";
3) Sequence label will be substituted for "#######"; and,
4) A study identification number will be substituted for "XXXX".

For each form in each time period, the program produces a set of code to make all the necessary substitutions. Referring again to our study description in Figure 2, one section of code would be generated for "PTPHYS" and four sections for "LBEVAL". The resulting code is then executed producing the dictionary, with unique variable names for each variable in the study and appropriate numeric indicators within the variable name to indicate the time period at which that variable appeared. A section of the dictionary corresponding to our sample study description is shown in Figure 4.

AN EXAMPLE, STEP 2:

The second SAS program executed produces the SAS data set for the study. When executed, it first builds a label macro and a format macro. Then it scans through the dictionary again and builds input statements, using the variable name, the starting column and the input format for all variables in each form (record) and time period (sequence) combination. "OUTPUT" statements direct the resulting data to separate SAS datasets, one for each record and sequence combination. Code for sorting and printing each dataset is then added, and, finally, code for a "MERGE" of all the datasets is produced. The resulting code is executed, using the formal library and the raw data as input, and produces a labelled, formatted SAS dataset for that study with one observation per patient.

AN EXAMPLE, STEP 3:

The final SAS program generates a set of edits of the data. Referring again to the form description in Figure 3, recall that the first column indicates the edit to be performed on that variable. Currently, we classify most of our variables into four types: discrete (D), continuous (C), list (L), and special (S). The program scans the dictionary and creates temporary datasets of the variables of each edit type. Code is generated and executed, producing the following output for each edit type:

1) Frequency tables are produced for all discrete (D) variables,
2) for continuous (C) variables the output from "PROC UNIVARIATE", with PLOT and NORMAL options, is generated, using a variable created from the combination of investigator and patient number as an ID in order to find outliers and identify their source.
3) List (L) variables are those whose values are formatted in an external table. Those tables are referred to by number in the format section of the dictionary entry. They are accessed by the edit program and lists of the formatted values by patient are produced.
4) "PROC PRINT" is generated for the special (S) variables in alphabetical order.

DISCOVERIES

During the creation of our Automated SAS Program Producer, we found a number of items which helped produce SAS datasets in faster or more efficient ways. As mentioned before, "%INCLUDE" made it possible to generate the SAS code and execute it all in the same step. Next, we discovered that 90% of the disk space and half the execution time could be saved by using "OPTIONS GEN=NO". An additional saving of about half the storage area could be gained by using "LENGTH DEFAULT=4". Further savings could be generated by use of several LENGTH statements and including each variable in the appropriate one.

CONCLUSION

We are grateful for all the input from our colleagues during the development of our "first shot" at this system, and plan to continue to enhance what we have. Our Automated SAS Program Producer has already created savings in time and effort, and will probably continue to do so as more improvements are implemented.

REFERENCES


Figure 1 - Overview of the System
Figure 2 - Study Description

PTHST 10 001
PTPHYS 11 001
LBEVAL 12 001 002 003 004
SSS1 18 001 002 003 004
SSS2 19 001 002 003 004
USLP 14 001
PSSQ 15 001 002 003 004
PGIVS 16 001 002 003 004
TRMCD 17 001 002 003 004

Figure 3 - Form Description

22 INV 4. INVXXXF INVESTIGATOR
26 PAT 4. PATIENT
S30 MCX** 1. MCXXXF 'MEDICATION CODE (REC **, SEQ %)'
34 DAT%** 6. 'DATE OF PATIENT PHYSICAL (########)'
D40 AB%_1 1. YN '#### HEAD AND NECK
L41 ABD%_1 4. 7 '#### HEAD AND NECK PROBLEM
D45 AB%_2 1. YN '#### CHEST
L46 ABD%_2 4. 7 '#### CHEST PROBLEM
D50 AB%_3 1. YN '#### ABDOMEN
L51 ABD%_3 4. 7 '#### ABDOMEN PROBLEM
D55 AB%_4 1. YN '#### EXTREMITIES
L56 ABD%_4 4. 7 '#### EXTREMITIES PROBLEM

Figure 4 - Data Dictionary Entries Produced
by Input Lines of Figure 3

11 1 22 INV 4. INV2304F INVESTIGATOR
11 1 26 PAT 4. PATIENT
11 1 S30 MC1_11 1. MC2304F 'MEDICATION CODE (REC 11, SEQ 1)'
11 1 34 DAT1_11 6. 'DATE OF PATIENT PHYSICAL (SCREEN)'
11 1 40 ABI_1 1. YN 'SCREEN HEAD AND NECK
11 1 L41 ABD1_1 4. 7 'SCREEN HEAD AND NECK PROBLEMS
11 1 45 ABI_2 1. YN 'SCREEN CHEST
11 1 L46 ABD1_2 4. 7 'SCREEN CHEST PROBLEM
11 1 50 ABI_3 1. YN 'SCREEN ABDOMEN
11 1 L51 ABD1_3 4. 7 'SCREEN ABDOMEN PROBLEM
11 1 55 ABI_4 1. YN 'SCREEN EXTREMITIES