Advanced Techniques for Creating Format with Variable Values

Erika Liu, Allergan, Irvine, California

ABSTRACT

It is often necessary to have formats in order to generate tables or to print off datasets in certain nice layout. Programmers usually use PROC FORMAT with lots of value statements to do it, which is tedious and time consuming, and also typing in the name of each variable manually is prone to error. There is a more efficient way to create formats without having to write VALUE statements by altering the variable values. This paper first explore a way of specifying options in PROC FORMAT to use variable's values to create its format, and then demonstrates how to generate generic codes to make formats automatically by using MACRO. An example of Adverse Events is presented to demonstrate how to use this simple and helpful tip.

INTRODUCTION

The most common and easy way to create format is to use value statements in the FORMAT procedure. In generating listing or tables processing, we usually print out each item of variables with meaningful sentences instead of codes. For instance, in order to print a listing of adverse event, Body Parts and Reaction Terms are frequently used to sort a dataset for safety analysis. It's very useful to use this technique. First of all, this paper will show you how to build Body Parts and Reaction Terms formats with control data sets to create a list of adverse event efficiently.

OPTIONS IN PROC FORMAT

To construct formats with a input control data set, an option CNTLIN = in the FORMAT procedure can be used to achieve our goal as long as the FMTNAME, START, and LABEL variables are created for the output control data set. An input control data set usually comes from a previous invocation of the FORMAT procedure, or it may be a codebook data from data entry or from a existing database. An output control data set contains specific information about each range (START) and value (LABEL) pair associated with a format (FMTNAME)). It takes two steps: DATA step and FORMAT procedure. In the DATA step, it conforms to the control data set structure. While in the second step, it creates its format with CNTLIN option. Let's use the adverse event as an example to demonstrate how to use this technique.

CREATE CONTROL DATA SET

Step 1: Within Data Step

To output the Adverse Event, we need to count how many patients and percentage for each event according to the orders of body parts and reaction terms. There are two formats “bodypart” and “costar” constructed from Body Parts and Reaction Terms in data step in order to get the output counting and sorting correctly. The codes are described as following to create the control data set (AECNTL):

```
Proc sort data=ae;
        by bodypart costar;
        run;
data aecntl;
        set ae;
        by  aevbdy costar;
        num=_n_;
        rename              bodypart=start
        num=label;
        fmtname=’bodypart’;
        run;
```

Step 2: FORMAT Procedure

Once the control data set created, the “bodypart” can be built by using the FORMAT procedure with CNTLIN options as follows:

```
proc  format cntlin=bodypart;
         run;
```

Thus the format of ‘bodypart” has been created successfully. To create the format of “costar”, we only need to change the word of “bodypart” to “costar” repeatly in both steps.

GENERATE GENERIC CODES USING MACRO

Since we have to repeat the steps to create another format by using variables values, we can write the generic codes as follows:

```
%macro fmt(var);
data &var;
        set ae;
        num=_n_;
        rename &var=start num=label;
        fmtname=”&var”;
        run;
proc format cntlin=&var; run;
%mend fmt;
%fmt(bodypart);
```

EXAMPLE TO USE FORMAT

```
%fmt(costar);
data ae;
        set ae;
        aeid=put(ae, costar.);
        run;
proc freq data=ae;
         tables aeid/out=aeout;
         run;
proc sort data=aeout;
         by aeid;
         run;
```

```
data aeout;
        retain count (0);
        set aeout;
        by aeid;
        count+1;
        if last.aeid then
        call symput(“aecnt”,put(count,2.));
        run;
```

Apparently, we can easily count how many patients and percentage for each adverse event as we described above.

CONCLUSION

It is obvious that using these options to construct formats to manipulate data or to generate tables is more efficient and flexible than the way of using VALUE statements. Once you know how to use the techniques, you will find out it is a powerful and helpful tip in various fields.

REFRENCES

SAS Procedures Guide, ver. 6, third edition.

AUTHOR CONTACT

Erika Liu

2525 Dupont Dr.

Irvine, CA 92623

E-mail: liu_erika@allergan.com