

Paper 218-28

Creating Tables or Listings with a Zero-Record SAS® Data Set -- Basic Program Structure and Three Simple Techniques

Rubin Nan, Victor Loher
PRA International, Inc., Lenexa, Kansas

ABSTRACT

Oftentimes, particular tables or listings in a clinical trial deal with zero-record SAS® datasets. This occurs primarily in tables/listings that deal with death or serious adverse events. To avoid the hard coding and to re-use the SAS® programs created for the interim analyses, the logical routes in SAS® programming should program for both conditions: one with zero-records and another with non-zero records. This paper will provide basic program structure to program for this type of table or listing and also provide three Techniques: EOF (end of file) option, SAS® Help library, and PROC SQL procedure-created macro variable.

INTRODUCTION

To generate tables or listings in clinical trials for reporting to regulatory agencies, SAS® programmers often need to create tables or listings using a zero-record dataset. This occurs primarily in tables/listings that deal with death or serious adverse events. In order to show headers and titles of a table or listing by using the SAS® PROC REPORT procedure, programmers can generate a new dataset and put new data into the dataset to satisfy the tabling purpose, but this may violate a corporate programming convention against hard coding. Moreover, in a clinical study that has an interim analysis, zero-record datasets at the interim analysis may contain records later with the addition of new subjects. This will require the creation of new coding for tables or listings, thus losing job efficiency. This paper will create a SAE listing as an example using three simple

ways to generate logic routes to manage a situation that has records in the dataset and one that has zero-records in the dataset.

HOW TO DO

The key to this is to set up two logical routes: one for the dataset that has zero records and another for the dataset that has non-zero records. Three techniques are presented in this paper to create logical conditions and sequentially direct implementation of the two logical routes. Here, I will use a SAE-related listing as an example. In this example, I assume that an adverse event (AE) dataset exists in a Master library, and the AE dataset contains no serious event (that is, the values of the serious variable in this dataset will be all blank; otherwise, the value will be '1' if a record shows a serious event).

The basic structure for creating this type of listing starts by using one of three techniques to create a macro variable, *obs*, with a possible value of zero or non-zero, by which, to discriminate the zero or non-zero conditions.

The three techniques are shown as follows:

Technique 1: Use Proc SQL procedure-created macro variable

```
Proc sql noprint;
  select count(*) into: obs
  from master.ae
  where serious = '1';
quit;
```

Technique 2: Use EOF Option

```

data _null_;
  set master.ae (keep =
    serious) end=eof;

  retain S_flag 0;
  if serious ne ' ' then
    S_flag= 1;

  if eof then do;

  ** If zero (no) SAE record;

  if not S_flag then
    call symput('obs','0');

  ** If SAE records are
    present;

  else
    call symput('obs','1');
  end;
run;

```

Technique 3: Use SAS Help Library

```

data _null_;
  set sashelp.vtable;

  if libname eq 'WORK' and
  memname eq 'AE' then do;

  if nobs eq 0 then
    call symput('obs','0');

  else
    call symput('obs','1');
  end;
run;

```

Once the macro variable is established, a macro (e. g. AE here) is used to execute the two logical routes.

```

%macro AE;

**if no SAE happened, then;

```

```

%if &obs = 0 %then %do;

```

```

*****

```

You can create a zero-record dataset using the following method. Of course, you can also use whatever method you can to create a dataset;

```

*****;

```

```

data AE;
  array var1, var2,...;
run;

```

```

proc report data=AE
nowindows;
  col var1 var2 ...;
  define var1;
  define var2;
  ...

```

```

break after /skip;

```

```

compute after;

```

```

line ' ';

```

```

...

```

```

line @65 'No serious
adverse events were reported';
endcomp;

```

```

run;

```

```

%end;

```

```

** if SAE happened, then show
regular listings or tables;

```

```

%else %do;

```

```

proc report data = AE
nowindows;
  col var1 var2 ...;
  define var1;
  define var2;
  ...

```

```

break after /skip;

```

```

%end;

```

```

%mend;

```

```

%AE;

```

CONCLUSION

This paper presents the basic structure and three simple programming techniques to deal with zero-record SAS® datasets in the process of creating tables or listings. With this basic structure and one of the three techniques, SAS® programmers can program tables or listings without losing efficiency in projects having an interim analysis while avoiding the use of hard coding.

ACKNOWLEDGMENTS

The author and co-author of this paper sincerely thank David Mullins and Doug Lassman for their taking precious time to review the draft.

CONTACT INFORMATION

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

Rubin Nan
PRA International, Inc.
16400 College Boulevard
Lenexa, Kansas 66219
Work phone: (913)-577-2882
Fax: (913) 599-0344
Email: nanrubin@praintl.com

Victor Loher
PRA International, Inc.
16400 College Boulevard
Lenexa, Kansas 66219
Work phone: (913)-577-2930
Fax: (913) 599-0344
Email: lohervictor@praintl.com

SAS and all other SAS Institute Inc. product or service names are registered trademark 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.