ABSTRACT

It is often necessary to produce summary tables showing statistical values such as mean, median, standard deviation (sd), minimum, and maximum, for numerical data such as lab parameters. These values may be ordered by other variables such as treatment group or visit.

This paper describes an easy to use, flexible macro that will produce a variety of these tables.

The SAS product used in this paper is SAS BASE, with no limitation of operating systems.

INTRODUCTION

The SAS macro `dstat.sas` conforms to the author's company standards that require a linesize of 90 and a pagesize of 32, which corresponds to a 6 inch by 9 inch display area for Courier 12 point font.

The arguments of this macro are:

<table>
<thead>
<tr>
<th>ARGUMENT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>dsin</td>
<td>the prepared dataset from which the summary will be produced</td>
</tr>
<tr>
<td>sortby1</td>
<td>the first sortby variable which will become column one</td>
</tr>
<tr>
<td>sortby1f</td>
<td>the format for the sortby1 variable</td>
</tr>
<tr>
<td>noprintv</td>
<td>a variable that provides the correct sort order but does not print on the summary report</td>
</tr>
<tr>
<td>grpvar</td>
<td>the grouping variable</td>
</tr>
<tr>
<td>grpf</td>
<td>the format for the grpvar variable</td>
</tr>
<tr>
<td>sortby2</td>
<td>the second sortby variable</td>
</tr>
<tr>
<td>sortby2f</td>
<td>the format for the sortby2 variable</td>
</tr>
<tr>
<td>value</td>
<td>the variable used for the statistical summary</td>
</tr>
<tr>
<td>chgbase</td>
<td>code &quot;Y&quot; for &quot;change from baseline&quot; title to appear across statistical columns</td>
</tr>
</tbody>
</table>

Required variables are dsin, value, and at least one of the sortby1, sortby2, or grpvar variables. All other variables are optional.

SAS MACRO DESIGN SPECIFICATION

The framework of the program flow is described as follows:

STEP 1

This step prepares the input dataset for the macro and accepts the user's input macro argument information.

There are ten macro arguments, namely, `dsin`, `sortby1`, `sortby1f`, `noprintv`, `grpvar`, `grpf`, `sortby2`, `sortby2f`, `value`, and `chgbase`.

STEP 2

This step checks to see if the grpvar variable is present. If it is, then N is calculated for each group using proc sql.

STEP 3

This step sorts the file by sortby1, grpvar, and sortby2. At least one of these three arguments must be valued or errors will result from the sort. The statistics (n, mean, median, standard deviation, minimum, and maximum) are then calculated using proc summary, and, if present, N is properly formatted.

STEP 4

This step is the proc report statement which produces the table. The column, define, and break statements are customized depending on the presence of the sortby1, sortby2, and grpvar variables. The linesize limitation requires that the width of each column be determined by the number of columns requested.

STEP 5

All temporary files are deleted by the proc datasets procedure at the end of the execution of the SAS macro `dstat.sas`.

SAS CODE FOR DESCRIPTIVE STATISTICAL SUMMARY TABLES

The entire macro code is as follows:

```sas
%macro dstat(dsin=, sortby1=, sortby1f=, noprintv=, grpvar=, grpf=, sortby2=, sortby2f=, value=, chgbase=);
    ******************************************;
    * Get total (N) for each grpvar group
    ******************************************;
    %if &grpvar ^= %then
    %do;
    A Reporting Macro to Create Descriptive Statistical Summary Tables
Ellen Forman, GlaxoSmithKline, Collegeville, PA
```
PROC SQL;
CREATE TABLE TOTGRP AS
   SELECT &GRPVAR, COUNT (DISTINCT PID)
     AS TOTN
   FROM &DSIN;
GROUP BY &GRPVAR;
QUIT;
END;
*******************************************************;
 SORT THE FILE.
*******************************************************;
PROC SORT DATA=&DSIN;
  BY &SORTBY1 &GRPVAR &SORTBY2; RUN;
*******************************************************;
* CALCULATE STATISTICS.
*******************************************************;
PROC SUMMARY DATA=&DSIN;
  BY &SORTBY1 &GRPVAR &SORTBY2;
  VAR &VALUE;
  %IF &NOPRINTV ^= %THEN
    %DO;
    ID &NOPRINTV;
    %END;
  OUTPUT OUT=SUMTOT N=N MEAN=MEAN STD=SD
       MEDIAN=MEDIAN MIN=MIN MAX=MAX;
RUN;
*******************************************************;
* CUSTOMIZE VARIABLES FOR PROC REPORT. IF
  GRPVAR IS NOT VALUED USE PROC DATASETS
  TO COPY SUMTOT TO RPTTOT.
*******************************************************;
%IF &GRPVAR ^= %THEN
  %DO;
  PROC SQL;
    CREATE TABLE RPTTOT AS
      SELECT S.*,
          T.TOTN,
          PUT (TOTN, 4.) AS TOTNC
      FROM SUMTOT S, TOTGRP T
        WHERE S.&GRPVAR = T.&GRPVAR;
    QUIT;
  %END;
%ELSE %DO;
  PROC DATASETS LIBRARY=WORK NOLIST;
    CHANGE SUMTOT=RPTTOT;
  RUN;
  %END;
%END;
*******************************************************;
* SORT FILE FOR REPORT
*******************************************************;
PROC SORT DATA=RPTTOT;
  BY &SORTBY1 &GRPVAR &NOPRINTV &SORTBY2;
RUN;
*******************************************************;
* PRODUCE THE TABLE
*******************************************************;
PROC REPORT DATA=RPTTOT HEADLINE HEADSKIP
   COLUMN (&SORTBY1 &GRPVAR %IF &GRPVAR ^= %THEN TOTNC;
        T.TOTN, 4.) AS TOTNC
        FROM SUMTOT S, TOTGRP T
        WHERE S.&GRPVAR = T.&GRPVAR;
    QUIT;
%END;
%ELSE %DO;
  PROC DATASETS LIBRARY=WORK NOLIST;
    CHANGE SUMTOT=RPTTOT;
  RUN;
%END;
*******************************************************;
* DELETE TEMPORARY DATASETS
*******************************************************;
PROC DATASETS NOLIST;
  DELETE &DSIN TOT TOT1 TOTGRP RPTTOT;
RUN;
%END;
*******************************************************;
SAS MACRO INVOCATION AND APPLICATION
Before the macro invocation, it is important that you use
the DATA STEP to prepare the SAS dataset to be used in
the summary report. A few examples of the macro will
Example 1:
Example 1 demonstrates the simplest use of the macro, using only one sortby variable. This example can be useful in searching for outliers.

Example 2:
Example 2 displays the summary statistics for each Electrocardiogram (ECG) parameter, grouped by treatment group.

Example 3:
Example 3 shows the summary statistics for each ECG parameter, grouped by visit. V iso was created as a nonprintv variable to properly order the Visit Id column since a format was used for the sortby2 argument.

Example 4:
Example 4 displays summary statistics for each ECG parameter, grouped by visit within treatment group. Txvis was created as a nonprintv variable to properly order the Visit Id column since a format was used for the sortby2 argument. The output example is shown in Figure 4.

Example 5:
Example 5 demonstrates the flexibility of the macro by substituting centre for treatment group as the groupby variable. Ctrvis was created as a nonprintv variable to properly order the Visit Id column since a format was used for the sortby2 argument. The output example is shown in Figure 5.
CONCLUSION

This paper describes a SAS macro for generating statistical summary reports that is both flexible and easy to use. Various examples of how to invoke the macro are provided along with corresponding output.

ACKNOWLEDGMENTS

The author would like to thank Shi-Tao Yeh, Biostatistics & Data Management, GlaxoSmithKline, for his help, suggestions, and review.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:
Ellen Forman
GlaxoSmithKline
Work Phone: (610) 917-6180
Email: Ellen.2.Forman@gsk.com

Example 6 displays the summary statistics for change from baseline in hemoglobin by visit. The example uses the chgbase argument to add the "Change from Baseline" header over the column headings for the summary statistics.