SAS® Macros for Grouping Count and Its Application to Enhance Your Reports

Shi-Tao Yeh, EDP Contract Services, Bala Cynwyd, PA

ABSTRACT

This paper provides two SAS® macros, one for one grouping variable, and the second for two grouping variables. Both macros count the subjects in the grouping variables and the number of sub-groups. For some report-writing tasks, the number of sub-groups and subjects in each subgroup may vary and cannot be predicted prior to invocation. The macros that perform the counts include an overall total number of subjects and a summation of subjects in each subgroup. They then save the counts as SAS macro variables for later use. This paper also discusses how to use these macros with provided examples to illustrate how to enhance your reports.

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

INTRODUCTION

How to produce the customized reports effectively is a challenge to many programmers. The SAS procedures of TABULATE and REPORT are popular and powerful tools for report-writing. In the past, topics of report-writing using SAS products had been presented at different SAS User Group Conferences. [2], [3], [4], [5], [6].

Some customized reports involve tabulation of categorical or grouping variables and a presentation of subject counts in the header row of the reports. The count for each sub-group may vary and is unknown prior to report-writing procedure. This paper provides SAS macros to compute the number of subject counts. The SAS macros then save it as macro variables for later macro variable calling. This paper demonstrates techniques for a series of data manipulation steps in conjunction with the FORMAT statement to produce customized reports. Two SAS macros are provided: one for one grouping variables and the other for two grouping variables.

DATA FILE

A data file contains fifty-three patients and five variables; Patient ID, Sex, Age, Reason for withdraw, and Treatment group, from a clinical trial is used for all SAS output lists and graphs in this paper. The variables in the data file are:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>pid</td>
<td>Char</td>
<td>Patient ID</td>
</tr>
<tr>
<td>sex</td>
<td>Char</td>
<td>Sex</td>
</tr>
<tr>
<td>age</td>
<td>Num</td>
<td>Age in Years</td>
</tr>
<tr>
<td>trt</td>
<td>Char</td>
<td>Treatment Group</td>
</tr>
<tr>
<td>wd</td>
<td>Num</td>
<td>Reason for Withdraw</td>
</tr>
</tbody>
</table>

SAS MACRO FOR ONE GROUPING VARIABLE

The following SAS macro is developed for computing one grouping variable and its subject count. The arguments for this macro are:

dsin: name of input dataset

varin: name of grouping variable in the input dataset
dsout: name of output dataset

varout: name of output variable that contains counts for each sub-group.

%macro gpcount(dsin=, varin=, dsout=, varout=);
  data _dsin;
  set &dsin;
  count = 1;
  keep &varin count;
  proc sort data= _dsin; by &varin;
  proc summary data = _dsin;
    class &varin;
    var count;
    output out= &dsout sum = &varout;
  run;

  data &dsout;
    set &dsout;
    if _type_ = 0 then call symput('nall', put( &varout, 4.));
    keep &varin &varout _type_;
  %global nall;

  data &dsout;
    set &dsout;
    if _type_ = 1;
    drop _type_;
  data t1;
    set &dsout;
    keep &varout;
  data &dsout;
    set &dsout nobs = _nobs;
    call symput('_nobs', put(_nobs, 3.));
  %do i = 1 %to &_nobs;
  %tp;
  proc transpose data=t1 out = t2;
  data t2;
    set t2;
    %do i = 1 %to &_nobs;

call symput("n&i", put(col&i,4.));
%end;
%mend tp;
%tp;
%mend gpcount;

After execution of this macro, it produces several SAS macro variables:
1) &nall : total count of subjects,
2) &n1, &n2, …, &nk for count of each sub-group, where k is the number of sub-group.

It also creates a data set and a variable with user's specified names. The output variable contains &n1, &n2, ..., &nk as its observations in the data file.

FORMAT

Formats are a set of instructions to allow the SAS system to read or display the data. The sources of formats are either system-supplied or user defined formats. Formats include both the FORMAT statement and FORMAT procedure. Formats can be efficient report writing tools. If used correctly, they can reduce data manipulation steps. The following PROC FORMAT statement creates temporary formats that are used through-out the SAMPLE CODE examples in the paper.

```sas
proc format;
  value $sexf 'M' = '-   MALE'
           'F' = '-   FEMALE';
  value $trt
        'A' = "Placebo   (N = &n1)"
        'B' = "50 mg     (N = &n2)"
        'C' = "100 mg    (N = &n3)";
  value agec
        0 - 34 = '-   < 35 yrs.'
        35 - 60 = '-   35 - 60 yrs'
        other =  '-   > 60 yrs';
  value $sexfa
        'F' = "Female  (N = &a1)"
        'M' = "Male     (N = &a2)";
  value $trta
        'A' = '-   Placebo'
        'B' = '-   50 mg'
        'C' = '-   100 mg';
  value ageca
        0 - 45 = 'A'
        other =  'B';
  value $agecfa
        'A' = "< 45 Yr.  (N = &b1)"
        'B' = "> 45 Yr.  (N = &b2)";
  value wdf
        1 = 'Adverse Experience'
        2 = 'Insufficient Effect'
        3 = 'Protocol Deviation'
        4 = 'Other';
  picture percpic (round)
run;
```

Tip: Picture formats are numeric formats only. The picture is a sequence of characters in single quotes.

SAMPLE CODE I

The following SAS code illustrates the usage of the SAS macro gpcount, in conjunction with FORMAT statement.

```sas
%gpcount(dsin=f1,varin=trt, dsout=p1, varout= _sum);

proc sort data=f1; by trt;
data final;
merge f1 p1;
by trt;
count = 1;
pctn = ( count / _sum ) * 100;
pctall = ( count / &nall) * 100;
format age agec.
title 'Example Output 1';
proc tabulate data=final;
class sex trt age;
var pctn pctall count;
table sex='Gender' age='Age Group',
       trt='Treatment Group'*sum=' '*
           (count= 'N'*f=5.0 pctn='%'*f=5.1)
       all="Total   (N = &nall)"*sum=' '*
           (count='N'*f=5.0 pctall='%'*f=5.1)
       /rts=15 row=float;
format trt $trt. sex $sexf. age agec.;
```

The output list from this SAMPLE CODE I is shown in Table 1.

<table>
<thead>
<tr>
<th>TREATMENT GROUP</th>
<th>Placebo (N = 12)</th>
<th>50 mg (N = 12)</th>
<th>100 mg (N = 23)</th>
<th>Total (N = 53)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENDER</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Female</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>Male</td>
<td>12</td>
<td>4</td>
<td>19</td>
<td>35</td>
</tr>
<tr>
<td>AGE GROUP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt; 35 yrs</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>35 - 60 yrs</td>
<td>12</td>
<td>4</td>
<td>11</td>
<td>29</td>
</tr>
<tr>
<td>&gt; 60 yrs</td>
<td>4</td>
<td>5</td>
<td>9</td>
<td>18</td>
</tr>
</tbody>
</table>

Table 1. Output from SAMPLE Code I
Tip: You can use format statement for a numeric variable without data manipulation to present it as categorical variable in tabulation procedure

**SAMPLE CODE II**

Suppose the specification of the report is to display the patients with an age greater than 60 years.

Before calling the PROC TABULATE procedure in SAMPLE I code, you can add one more data step to accomplish this task.

```sas
data final final1;
set final;
if age > 60 then output final;
else output final1;
```

**Tip:** This step allows you to eliminate some unneeded rows or columns.

This additional data step stores the patients that meet the criteria. The output list is shown in the following table.

<table>
<thead>
<tr>
<th>GENDER</th>
<th>Placebo (N = 18)</th>
<th>50 mg (N = 12)</th>
<th>100 mg (N = 23)</th>
<th>Total (N = 53)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Female</td>
<td>5.6</td>
<td>25.0</td>
<td>10.0</td>
<td>7.12</td>
</tr>
<tr>
<td>Male</td>
<td>15.7</td>
<td>16.7</td>
<td>26.1</td>
<td>20.8</td>
</tr>
</tbody>
</table>

**Table 2. Output from SAMPLE CODE II with Data Final**

You can input dataset final1 to SAMPLE CODE II and produce the following table for patients with less than 60 years.

<table>
<thead>
<tr>
<th>GENDER</th>
<th>Placebo (N = 18)</th>
<th>50 mg (N = 12)</th>
<th>100 mg (N = 23)</th>
<th>Total (N = 53)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Female</td>
<td>22.2</td>
<td>41.7</td>
<td>35.1</td>
<td>34.0</td>
</tr>
</tbody>
</table>

**Table 3. Output from SAMPLE CODE II with Data Final1**

An alternative method to present the wanted row in the table was suggested by Bernholc[1]. This method treats the categorical variable as an analysis variable with additional binary variable coded as 0 or 1. The proportion or percentage is the mean of 1s in the binary variable. This method can also produce the desired output.

**SAMPLE CODE III**

The following sample code uses the same data set final conjunction with PROC REPORT procedure and FORMAT statement to produce Table 3.

```sas
data final;
set final;
tcount = '1';
if wd ne .;

proc report data = final headline headskip center nowindows split='*';
column ("==" wd ('--- TREATMENT GROUP -- -- TREATMENT GROUP -- --') tcount, ('--- count pctn ) tcount, ('--- count pctall) );
define wd /group 'REASON OF WITHDRAWAL' f=wdf. width=20;
define trt / across order=internal f=$trt. ' ';
define count / analysis sum 'N' width=6;
define pctn / analysis sum f=percpic. width=8 spacing=0 '(%)';
define tcount / group order "Total *(N = 23)"
        define pctall / analysis sum f=percpic. width=8 spacing=0 '(%)';
break after wd / skip;
run;
```

**Table 4. Output from Sample Code III**

**SAS MACRO FOR TWO GROUPING VARIABLES**

The following macro expands the grouping variable to two and accepts the same arguments:

dsin: name of input dataset
dvarin: name of grouping variable in the input dataset
dsout: name of output dataset
dvarout: name of output variable that contains counts for each sub-group.
%macro gpcount2(dsin= , varin= , dsout= , varout= );
data _dsin;
set &dsin;
count = 1;
keep &varin count;
proc sort data= _dsin; by &varin;
proc summary data = _dsin;
class &varin;
var count;
output out= &dsout sum = &varout;
run;
data &dsout;
set &dsout;
if _type_ = 0 then call symput('nall',
put( &varout, 4.));
keep &varin &varout _type_;
%global nall _nobsa _nobsb;

%macro subg(sgin= , sgout= , typen= ,
labl1= , var1= );
data &sgout;
set &sgin;
if _type_ = &typen;
drop _type_;
data a1;
set &sgout;
keep &var1;
data a1;
set a1 nobs = _nobs&labl1;
call symput('_nobs&labl1',
put(_nobs&labl1, 3.));
%mend subg;
%subg(sgin = &dsout, sgout = outa, typen = 1, labl1= a , var1 = &varout);
%subg(sgin = &dsout, sgout = outb, typen = 2, labl1= b , var1 = &varout);
%macro tp(tin=, tout=, nobs = ,labl=);
proc transpose data= &tin out = &tout;
data &tout;
set &tout;
%do i = 1 %to &nobs;
call symput('_labl&i', put(col&i,4.));
%global &labl&i;
%end;
%mend tp;
%tp(tin = a1, tout = aout, nobs = _nobsa, 
labl=a);
%tp(tin = b1, tout = bout, nobs = _nobsb, 
labl=b);
%mend gpcount2;

Again, it produces several SAS macro variables:
1) &nall : total count of subjects,
2) &a1, &a2, ..., &ai for count of each sub-group in grouping variable 1 where i is the number of sub-group of grouping variable 1..
3) &b1, &b2, ..., &bj for count of each sub-group in grouping variable 2 where j is the number of sub-group of grouping variable 2..

**SAMPLE SAS CODE IV**

The following SAS code illustrates the usage of the SAS macro gpcount2 conjunction with FORMAT statement.

data f1;
set f1;
agec = put(age, agec.);
%gpcount2(dsin=f1,varin= agec sex, 
dsout=p1, varout= _sum);
proc sort data=f1; by sex;
proc sort data=outa(keep=sex _sum);by sex;
data final;
merge f1 outa;
by sex;
count = 1;
pcts = ( count / _sum ) * 100;
pctall = ( count / &nall) * 100;
drop _sum;
proc sort;by agec;
proc sort data=outb(keep=agec _sum);by agec;
data final;
merge final outb;
by agec;
pcta = (count / _sum ) * 100;
title ‘ ’;
proc tabulate data=final;
class sex trt agec;
var pcts pcta pctall count;
table trt ='TREATMENT GROUP',
sex='GENDER'*sum=' '
(count= 'N'*f=4.0 pcts='%'*f=5.1)
all="Total (N = &nall)"*sum=' '
(count= 'N'*f=4.0 pctall='%'*f=5.1)
agec='AGE GROUP'*sum=' '
(count= 'N'*f=4.0 pcta='%'*f=5.1)
/rts = 13 row = float;
format trt $trta. sex $sexfa. agec $agecfa.;
run;

The output from SAMPLE CODE IV is shown in Table 5.
Table 5. Output from SAMPLE CODE IV

CONCLUSION

This paper provides the SAS macros and SAS code for application by examples. It includes:

* SAS macros for grouping counts are the efficient way to generate the customized reports.

* A series of data steps to manipulate the data before calling procedures TABULATE or REPORT. This practice makes it easier for producing reports.

* The usage of FORMAT statement can reduce the necessity for data manipulation.

REFERENCES


SAS is a registered trademark of SAS Institute Inc., in the USA and other countries.

GIF is a service mark of CompuServe Incorporated.

® indicates USA registration.

Author

Shi-Tao Yeh, Ph. D.
(610)917-5883(W)
E-mail: shitaoyeh@sinanet.com