Facilitate Statistical Analysis with Automatic Collapsing of Small Size Strata

Sunil Gupta, Linfeng Xu, Quintiles, Inc., Thousand Oaks, CA

ABSTRACT

Often in clinical studies, even after great efforts are made to ensure protocol specified sample sizes in each stratum, there are still some strata with small sample sizes in final analysis stage. One major reason for smaller sample sizes in strata could be the difficulty to recruiting patients. When conducting statistical analyses by using the STRATA statement, such as LIFETEST test, there is a need to collapse strata with small sample sizes in order to meet the requirement of statistical procedure assumptions or statistical power considerations.

Thus, there is often need for collapsing strata with small sample sizes (usually less than 2). Specially, the method proposed in this paper is applicable when the strata variable is used in SAS® STRATA statement of selected statistical procedures or when the strata and treatment variables are used as model independent variables. The strata variable could be strata in a stratified sample design, such as disease stage. To collapse small size strata is to assign a new value to the strata variable of the observation when the cross frequency counts of this stratum by either placebo or treatment group is less than &pooln (or 2). The assigned new value could be the value as “pooled” or the different value of the strata variable.

This paper provides three methods for automatically collapsing strata by using the author developed SAS macro. The macro generally uses the strata variable with character ordinal value. By applying the macro, one of these three following approaches can be used: collapsing to the next higher stratum, collapsing to the next lower stratum, or collapsing all small sample size strata into one newly defined stratum. Then statistical procedures, such as LIFETEST, can be performed to get valid statistical results. In addition, vital sensitivity analysis can be preformed. This paper will discuss the strategies of collapsing data, the conceptual design of the macro, and the detailed SAS code for collapsing data strata to be used in statistical procedures.

BACKGROUND

One of the most common reasons for small sample sizes in strata is the difficulty to recruit protocol eligible patients. Another reason is that patients may withdraw early from the study. Once the data is locked, you have to conduct valid statistical analysis even if there are strata with small sample sizes. Some software have options to handle small sample sizes by automatically collapsing them. Since SAS procedures, however, still do not have this option, you will need to manually collapse them first.

STRATEGIES FOR COLLAPSING DATA

There are several methods for collapsing strata with small sample sizes.

First approach is to combine the small stratum to its similar group based upon clinical knowledge. Since the strata value definition usually has some meaning, collapsing the small stratum to its adjacent stratum is often a common approach. Some statistical software have this kind of options in some statistical procedures.

In addition, collapsing all the small size strata into one group (new) is also an alternative, if the data suggest that those small strata have common characteristics.

Sensitivity analyses usually are performed to explore the impacts of different collapsing approaches although you have to adhere to the collapsing method that you defined in statistical analysis plan (SAP). Otherwise, the method can be picked up from the sensitivity analysis.

CONCEPTUAL DESIGN OF %COMBSTRATA MACRO

The macro to combine strata ideally should have the following function features.

1) User assigns the input data set, the analysis variable of an input data set, the treatment group variable, and the stratum variable.

2) User assigns a collapsing number, which is the smallest number that user wants to collapse or poll data into next stratum. The default value is 2 (This is recommend default number in the paper by Barbara Lepidus Carlson).
3) Provide three ways to collapse data - collapsing to the next higher stratum, collapsing to the next lower stratum, or collapsing all small sample size strata into one newly created stratum. The &TYPE parameter can be used to fulfill this task.

4) User has the ability to exclude the cases with missing value of treatment variable, and to exclude the cases with missing value of strata variable.

5) User has the ability to apply data set extraction condition by providing &COND macro parameter.

This paper discussed %combstrat macro needs to have the following requirements based on team programming standards:

1) Assigned treatment group variable to &TXGROUP must be numeric variable with value of 0 for placebo and 1 for treatment.
2) Assigned strata variable to &STRTA must be character variable.
3) Excluded the cases with missing value for analysis variable.

PROGRAM PROCESS FLOW

The following is the program process flow.

1) User can define the name of input data set as macro parameter. The macro will eliminate the cases with missing value for analysis variable. Then, the macro will base upon the parameter value of &EXCLUDE_MISTRT and &EXCLUDE_MISSTRATA to decide whether or not to exclude the cases with missing value of treatment variable and missing value of strata variable.

2) Before conducting collapsing, a procedure of cross frequency is performed in order to check the frequency counts for each original value of strata variable by treatment group in order to compare with final pooled results to check if collapsing was done correctly.

3) The next step is to decide the type of collapsing that user wants to conduct. The macro parameter &TYPE has to be assigned. If &TYPE=ALL, then, macro will collapse all strata with less than parameter &POOLN (default is 2) into “pooled” group. If TYPE=NEXT_HIGH, then macro will combine the stratum with less than parameter &POOLN into next stratum with higher stratum value visa versa, when &TYPE=NEXT_LOW, the macro will collapse small sample size stratum to next stratum with lower stratum value.

4) Under &TYPE=ALL loop, the general idea is to identify those strata with either placebo or treatment group with a “stratum by treatment” counts less than &POOLN, then assign them to a new stratum value as “pooled” and leave other strata values as they were.

The detailed steps in processes are as follows.

- First, sort the input data by strata variable in order to use it as merging variable later.
- Then, run a cross frequency procedure to get “strata by treatment group” counts and save the output as a data set.
- Next, sort this data set by strata variable in order to prepare the transposing.
- Then, transpose this data set to let each stratum has two variables with the total counts of treatment and placebo group in each record.
- Based upon transposed data, if a stratum has the counts in either treatment or placebo group less than &POOLN number, you will set collapsing flag=1 to this stratum.
- Then, you will assign the stratum variable under collapsing flag=1 with “pooled” value to a new strata variable and assign original strata values to new strata variable too if the stratum is not flagged (flag=0).
- In next step, you will merge back the new strata variable to the input data set and drop the original strata variable and rename new strata variable back to original strata variable. The collapsing process is completed.
- Finally, you run a cross frequency to check if collapsing was done correctly. The end data set is saved as user defined &OUTNAME and all the intermediate data sets were deleted.

5) Under either &TYPE=NEXT_HIGH or &TYPE=NEXT_LOW situation, the similar ideas were used.

- You have to identify strata with total counts less than &POOLN. Then, created flag=1 for those strata.
- Based upon flag=1 and you will pick up the previous stratum value of the same treatment group to assign it to a new strata variable and assign original strata value to new strata variable if flag=0.
- Based upon this new strata variable and original strata variable, you will build a format. The start and end value of the format is the value of original strata variable and the label value of the format is the value from new strata variable.
- Once you built the format, you will apply this format back to original strata variable in original input data set in order to create pooled new strata variable with updated strata value.
- You will use this new strata variable in final output data set.
A little bit complexity is added when there is the situation with two or more than two consecutive strata with total counts less than &POOLN. In this case, you may conduct the similar collapsing process more than once. You will build a %do %while loop and use the macro variable &MY_TOT which is gotten from the minimum stratum counts after concurrent collapsing. The same process needs to be done more than once when &MY_TOT is less than 2.

In addition, if the total counts of the stratum with the edge value are less than &POOLN and you can not collapse it to next extreme value since there is no such value, then, you will pool it to next stratum value of opposite direction. For example, you select TYPE=NEXT_HIGH, you have the strata variable with values as 1, 2, 3, and 4. If you find that the counts in strata variable with value 4 is less than &POOLN (2), then you are supposed to collapse to value 5. However, there is no value 5. Therefore, by using the rule of similarity, you have to collapse this stratum to stratum with value 3. Another macro variable &LASTOT (to save the edge value) is created to handle this situation. When &MY_TOT is less then &POOLN and &MY_TOT=&LASTOT, then you will design the reverse collapsing process. Otherwise, you will go through another similar collapsing process until the minimum stratum counts are greater or equal to &POOLN.

**DETAIL CODE**

```sas
%macro combstrata(
  type=  ,  /*required: 
    all: to pull all strata with sizes smaller than pooln to one group. 
    next_high: to pull the stratum with size smaller than pooln to next 
    high stratum value group. 
    next_low: to pull the stratum with size smaller than pooln to next 
    low stratum value group*/

  indata=,  /*required: indata as input data set name*/

  outname=, /*required: the final output data set name*/

  cond=,    /*optional: data extraction condition, such as %str(if 
    evalitt="Y")*/

  caseid=,  /*required: unique caseid*/

  exclude_mistrt=Y,   /*required: excluding cases with missing values for treatment 
    variable*/

  exclude_misstrata=Y,/*required: excluding cases with missing values for strata 
    variable*/

  anavar=,  /*required: anavar as analysis variable name*/

  txgroup=, /*required: txgroup as treatment group variable-must be 
    with value 0 for placebo and 1 for treatment*/

  pooln=2,  /*required: pooln as the size of a pool that you want to pool 
    to next strata*/

  strata=   /*required: strata as character strata variable*/
);

option mprint;

* Delete missing value;

data my&indata;
  set &indata;
  &cond ;
  *if &anavar^=.;
  %if exclude_mistrt=Y %then %do;
  *if &txgroup^=.;
  %end;
  %if exclude_misstrata=Y %then %do;
  *if &strata^=.;
  %end;
run;

proc freq data=my&indata;
  table &txgroup*&strata/list missing;
```
title "Strata distribution by treatment group before collapsing";
run;

* if type as all*;
%if %upcase(&type)=ALL %then %do;
  proc sort data=my&indata out=all0_strata;
    by &strata;
  run;
  proc freq data=my&indata noprint ;
    table &txgroup*&strata/list missing out=all1_strata;
  run;
  proc sort data=all1_strata out=all2_strata;
    by &strata;
  run;
  proc transpose data=all2_strata out=all3_strata prefix=n;
    var count;
    id &txgroup;
    by &strata;
  run;
  * if one of treatment group with less than &pooln subjects then set pool flag to 1;
  data all4_strata;
    set all3_strata;
    if n0>=%eval(&pooln) and n1>=%eval(&pooln) then flag=0;
    else flag=1;
  run;
  data all5_strata;
    length n_strata $10;
    set all4_strata;
    if flag=1 then n_strata='pooled';
    else n_strata=&strata;
  run;
  proc sort data=all5_strata;
    by &strata;
  run;
  data &outname(rename=(n_strata=&strata));
    merge all0_strata all5_strata;
    by &strata;
    drop _name_ n0 n1 &strata   flag;
  run;
  proc datasets lib=work;
    delete  my&indata all0_strata all1_strata all2_strata all3_strata all4_strata all5_strata;
  run;
  proc sort data=&outname;
    by &caseid;
  run;
  proc freq data=&outname;
    table &txgroup*&strata/list missing;
    title "Strata distribution by treatment group after collapsing";
  run;
%end; /*end of type=all*/
* if type as next_high or next_low;

%let my_tot=0;
%if %upcase(&type)=NEXT_HIGH or %upcase(&type)=NEXT_LOW %then %do;
  %do %while (%eval(&my_tot)<%eval(&pooln));
* get strata by txgroup cross frequency count;
  proc freq data=my&indata noprint ;
    table &txgroup*&strata/list missing out=all1_strata;
  run;

  proc sort data=all1_strata out=all2_strata;
    by &strata;
  run;

  proc transpose data=all2_strata out=all3_strata prefix=n;
    var count;
    id &txgroup;
    by &strata;
  run;

* if one of treatment group with lt 2 counts then flag equal 1;
  data all4_strata;
    set all3_strata;
    if n0>=%eval(&pooln) and n1>=%eval(&pooln) then flag=0;
    else flag=1;
  run;

  proc sort data=all4_strata nodupkey out=all5_strata;
    by &strata;
  run;

* merge flag back;
  data checkpooln;
    merge all2_strata all5_strata;
    by &strata;
    drop _name_ n0 n1 ;
  run;

  proc sort data=checkpooln;
    %if %upcase(&type)=NEXT_HIGH %then %do;
      by &txgroup descending &strata ;
    %end;

    %if %upcase(&type)=NEXT_LOW %then %do;
      by &txgroup &strata ;
    %end;
  run;

  data checkpooln;
    set checkpooln;
    %if %upcase(&type)=NEXT_HIGH %then %do;
      by &txgroup descending &strata;
    %end;

    %if %upcase(&type)=NEXT_LOW %then %do;
      by &txgroup &strata ;
    %end;
    run;

  data checkpooln;
    set checkpooln;
    %if %upcase(&type)=NEXT_HIGH %then %do;
      by &txgroup descending &strata;
    %end;

    %if %upcase(&type)=NEXT_LOW %then %do;
      by &txgroup &strata ;
    %end;
    pre_&strata=lag(&strata);
    pre_&txgroup=lag(&txgroup);
    run;
if first.&txgroup then do;
  pre_&strata=&strata;
  pre_&txgroup=&txgroup;
end;

* if this stratum needs to be pooled then assign the next stratum value to it
  depending on collapsing type;

if flag=1 and pre_&txgroup=&txgroup then fmt_&strata=pre_&strata;
else fmt_&strata=&strata;
run;

* create two formats to replace original strata value with new strata value;

* placebo group format;

data placebofmt;
  set checkpooln;
  if &txgroup=0;
    start=&strata;
    end=&strata;
    label=fmt_&strata;
    fmtname='pchg';
    type='C';
    run;
proc format library=work cntlin=placebofmt fmtlib maxselen=200
  maxlable=200;
run;

* treatment group format;

data trtfmt;
  set checkpooln;
  if &txgroup=1;
    start=&strata;
    end=&strata;
    label=fmt_&strata;
    fmtname='tchg';
    type='C';
    run;
proc format library=work cntlin=trtfmt fmtlib maxselen=200 maxlablen=200;
run;

* Apply format to replace strata values with pooled strata value;

data my&indata ;
  set my&indata (rename=(&strata=org_&strata));
  if &txgroup=0 then &strata=put(org_&strata,$pchg.);
  if &txgroup=1 then &strata=put(org_&strata,$tchg.);
  drop org_&strata;
run;

proc freq data=my&indata noprint;
  table &txgroup*&strata/list missing out=smallcount;
  title "After collapsing in final &indata";
run;

proc sql;
  select min(count) into :mytot from smallcount;
  
quit;

* get the smallest counts of last stratum and use this
  to compare with macro variable mytot to see if smallest exist in the very end;
%if %upcase(&type)=NEXT_HIGH %then %do;
  proc sort data=smallcount;
    by descending &strata &txgroup;
  run;
  data lasttot;
    set smallcount;
    if _n_=1;
    call symput("lastot",put(count,10.));
  run;
%end;
%if %upcase(&type)=NEXT_LOW %then %do;
  proc sort data=smallcount;
    by &strata &txgroup;
  run;
  data lasttot;
    set smallcount;
    if _n_=1;
    call symput("lastot",put(count,10.));
  run;
%end;
%put "all_small_count=" &mytot;
%put "last_stata_small_count=" &lastot;
%if &mytot>=2 %then %goto myend;
%if &mytot<2 and &mytot^=&lastot %then %do;
  * clean intermediate data sets;
  proc datasets lib=work;
    delete  all1_strata all2_strata all3_strata all4_strata all5_strata
      checkpooln trtfmt placebofmt ;
  run;
%let my_tot=&mytot;
%end;
%if &mytot<2 and &mytot=&lastot %then %do;
  * clean intermediate data sets;
  proc datasets lib=work;
    delete  all1_strata all2_strata all3_strata all4_strata all5_strata
      checkpooln trtfmt placebofmt ;
  run;
  * if the last stratum with smallest counts- you have to;
  *  combine it to next stratum in opposite direction;
  proc freq data=my&indata noprint ;
    table &txgroup*&strata/list missing out=all1_strata;
  run;
  proc sort data=all1_strata out=all2_strata;
    by &strata;
  run;
  proc transpose data=all2_strata out=all3_strata prefix=n;
    var count;
    id &txgroup;
    by &strata;
  run;
  data all4_strata;
    set all3_strata;
if n0>=%eval(&pooln) and n1>=%eval(&pooln) then flag=0;
else flag=1;
run;

proc sort data=all4_strata nodupkey out=all5_strata;
by &strata;
run;

data checkpooln;
merge all2_strata all5_strata;
by &strata;
drop _name_ n0 n1 ;
run;

proc sort data=checkpooln;
* notice the change of sorting order corresponding to type and it also;
* use the opposite ascending or descending sorting comparing to similar;
* procedures in first part;
%if %upcase(&type)=NEXT_HIGH %then %do;
by &txgroup &strata ;
%end;
%if %upcase(&type)=NEXT_LOW %then %do;
by &txgroup descending &strata ;
%end;
run;

data checkpooln;
set checkpooln;
%if %upcase(&type)=NEXT_HIGH %then %do;
by &txgroup &strata;
%end;
%if %upcase(&type)=NEXT_LOW %then %do;
by &txgroup descending &strata;
%end;
run;

data checkpooln;
set checkpooln;
%if %upcase(&type)=NEXT_HIGH %then %do;
by &txgroup &strata;
%end;
%if %upcase(&type)=NEXT_LOW %then %do;
by &txgroup descending &strata;
%end;
preg _strata=lag(&strata);
preg _txgroup=lag(&txgroup);
if first.&txgroup then do;
preg _strata=&strata;
preg _txgroup=&txgroup;
end;
if flag=1 and preg _txgroup=&txgroup then fmt _strata=preg _strata;
else fmt _strata=&strata;
run;

data placebofmt;
set checkpooln;
if &txgroup=0;
start=&strata;
end=&strata;
label=fmt _strata;
fmtname='pchg';
type='C';
run;

proc format library=work cntlin=placebofmt fmtlib MAXSELEN=200
MAXLABELN=200;
run;

data trtfmt;
set checkpooln;
if &txgroup=1;
start=&strata;
The following test data set is used to illustrate macro calls as examples. The data set has patient ID variable (pt), numeric treatment group variable (trtcd-0 for placebo and 1 for treatment), character stratum variable (stratacd- with a,b,c,d values), and event variable (event-1 for with event and 0 for without event). There is one observation in "trtcd=1 and stratacd='b'" group. In addition, there is also one observation in "trtcd=0 and stratacd='d'". Those groups with observations less than two are the target groups that will be collapsed by calling the macro. You can make three different macro calls.
data one;
  input pt trtcd stratacd $ event;
cards;
1 1 a 1
2 1 a 1
3 1 a 0
4 0 a 1
5 0 a 1
6 0 a 1
7 1 b 1
10 0 b 1
11 0 b 1
12 0 b 1
10 0 b 1
11 0 b 1
12 0 b 1
13 1 c 1
14 1 c 1
15 1 c 1
16 1 c 1
17 1 c 1
19 1 c 1
20 0 c 1
21 0 c 1
22 0 c 1
23 0 c 1
24 0 c 1
25 1 d 1
26 1 d 1
27 0 d 1
);
run;

Before collapsing, you can see that “trtcd=0 and stratacd=d” and “trtcd=1 and stratacd=b” have total counts less than 2.

<table>
<thead>
<tr>
<th>trtcd</th>
<th>stratacd</th>
<th>Frequency</th>
<th>Percent</th>
<th>Cumulative Frequency</th>
<th>Cumulative Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>a</td>
<td>3</td>
<td>11.11</td>
<td>3</td>
<td>11.11</td>
</tr>
<tr>
<td>0</td>
<td>b</td>
<td>6</td>
<td>22.22</td>
<td>9</td>
<td>33.33</td>
</tr>
<tr>
<td>0</td>
<td>c</td>
<td>5</td>
<td>18.52</td>
<td>14</td>
<td>51.85</td>
</tr>
<tr>
<td>0</td>
<td>d</td>
<td>1</td>
<td>3.70</td>
<td>15</td>
<td>55.56</td>
</tr>
<tr>
<td>1</td>
<td>a</td>
<td>3</td>
<td>11.11</td>
<td>18</td>
<td>66.67</td>
</tr>
<tr>
<td>1</td>
<td>b</td>
<td>1</td>
<td>3.70</td>
<td>19</td>
<td>70.37</td>
</tr>
<tr>
<td>1</td>
<td>c</td>
<td>6</td>
<td>22.22</td>
<td>25</td>
<td>92.59</td>
</tr>
<tr>
<td>1</td>
<td>d</td>
<td>2</td>
<td>7.41</td>
<td>27</td>
<td>100.00</td>
</tr>
</tbody>
</table>

You can collapse these to “pooled” stratum if given &TYPE=ALL by using the following macro call.

```
%combstrata(type=all, indata=one, outname=test1, caseid=pt, exclude_mistrt=Y, exclude_misstrata=Y, anavar=event, txgroup=trtcd, pooln=2, strata=stratacd );
```

After the macro call, you will see all the stratacd='b' and stratacd='d' were pooled into stratacd='pooled' under both trtcd=0 and trtcd=1. Thus, 6+1=7 and 1+2=3.
### All Strata Grouping Option

<table>
<thead>
<tr>
<th>trtcd</th>
<th>stratacd</th>
<th>Frequency</th>
<th>Percent</th>
<th>Cumulative Frequency</th>
<th>Cumulative Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 a</td>
<td>3</td>
<td>11.11</td>
<td>3</td>
<td>11.11</td>
<td></td>
</tr>
<tr>
<td>0 c</td>
<td>5</td>
<td>18.52</td>
<td>8</td>
<td>29.63</td>
<td></td>
</tr>
<tr>
<td>0 pooled</td>
<td>7</td>
<td>25.93</td>
<td>15</td>
<td>55.56</td>
<td></td>
</tr>
<tr>
<td>1 a</td>
<td>3</td>
<td>11.11</td>
<td>18</td>
<td>66.67</td>
<td></td>
</tr>
<tr>
<td>1 c</td>
<td>6</td>
<td>22.22</td>
<td>24</td>
<td>88.89</td>
<td></td>
</tr>
<tr>
<td>1 pooled</td>
<td>3</td>
<td>11.11</td>
<td>27</td>
<td>100.00</td>
<td></td>
</tr>
</tbody>
</table>

Next, you can call the macro by using &type=next_high.

```plaintext
%combstrata(type=next_high,indata=one,outname=test2,caseid=pt,
exclude_mistrt=Y,exclude_misstrata=Y,anavar=event,txgroup=trtcd,pooln=2,
strata=stratacd); 
```

The above macro call will pool the stratacd='b' and 'd' to stratacd='c'. Collapsing 'd' to 'c' is because there is no stratum with value higher than 'd' and collapsing value in opposite direction is conducted for stratacd='d' group. Thus, 6+5+1=12 and 1+6+2=9.

### Next Higher Grouping Option

<table>
<thead>
<tr>
<th>trtcd</th>
<th>stratacd</th>
<th>Frequency</th>
<th>Percent</th>
<th>Cumulative Frequency</th>
<th>Cumulative Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 a</td>
<td>3</td>
<td>11.11</td>
<td>3</td>
<td>11.11</td>
<td></td>
</tr>
<tr>
<td>0 c</td>
<td>6</td>
<td>44.44</td>
<td>15</td>
<td>55.56</td>
<td></td>
</tr>
<tr>
<td>1 a</td>
<td>3</td>
<td>11.11</td>
<td>18</td>
<td>66.67</td>
<td></td>
</tr>
<tr>
<td>1 c</td>
<td>9</td>
<td>33.33</td>
<td>27</td>
<td>100.00</td>
<td></td>
</tr>
</tbody>
</table>

Finally, you can call the macro by using &type=next_low.

```plaintext
%combstrata(type=next_low,indata=one,outname=test3,caseid=pt,
exclude_mistrt=Y,exclude_misstrata=Y,anavar=event,txgroup=trtcd,pooln=2,
strata=stratacd); 
```

The above macro call will pool the stratacd='b' and 'd' to stratacd='a' and 'c' respectively. Thus, 6+3=9, 3+1=4, 5+1=6, and 6+2=8.

### Next Lower Grouping Option

<table>
<thead>
<tr>
<th>trtcd</th>
<th>stratacd</th>
<th>Frequency</th>
<th>Percent</th>
<th>Cumulative Frequency</th>
<th>Cumulative Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 a</td>
<td>9</td>
<td>33.33</td>
<td>9</td>
<td>33.33</td>
<td></td>
</tr>
<tr>
<td>0 c</td>
<td>6</td>
<td>22.22</td>
<td>15</td>
<td>55.56</td>
<td></td>
</tr>
<tr>
<td>1 a</td>
<td>4</td>
<td>14.81</td>
<td>19</td>
<td>70.37</td>
<td></td>
</tr>
<tr>
<td>1 c</td>
<td>8</td>
<td>29.63</td>
<td>27</td>
<td>100.00</td>
<td></td>
</tr>
</tbody>
</table>

Once you have collapsed the data set, you can run further statistical procedures by using newly generated data set. You can also test the impact of different collapsing methods easily by selecting the different collapsing methods followed by the same statistical procedure several times. The results of sensitivity analysis can provide more information for final decision.
on selecting the final collapsing strategy if collapsing method has not been already defined in SAP since you may not have anticipated this event in the design stage.

ALTERNATIVE METHOD

If you do not use the proposed macro, you will need to use proc freq to check the cross frequency distribution of "treatment by strata" to make sure which strata groups with counts less than &POOLN. Then, you will need to hard code the strata groups by using the "if" and "else if" for one kind of collapsing method. If you want to conduct sensitivity analysis for another collapsing method, you will need to hard code them also. This manual process is not only time consuming but also is prone to errors.

CONCLUSION

This macro is very easy to use and can save time for statisticians or programmers. The sensitivity analysis can be easily done by calling the macro with different &TYPE selections. It also will save programming time to generate final production version of data collapsing when many variables need to be analyzed under the same approach.

ACKNOWLEDGMENTS

Many thanks to Minying Royston for providing an opportunity to use similar macro in Amgen production programs. Many thanks for Dr. Xuena Wang and Dr. Nan Zhang for giving some inputs about collapsing data strategies on this paper. Many thanks for Dr. Mandyam Srirama and Mr. Rajesh Venkataaram for reviewing the paper.

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

REFERENCE

http://sda.berkeley.edu/HELPDOCS/semethod.htm, Methods Used by SDA 3.1 for Computing Standard Errors of Percentages and Means for Complex Samples.


CONTACT INFORMATION

Sunil Gupta, Associate Director, Statistical Programming
Quintiles, Inc.
325 E. Hillcrest Drive Suite 200
Thousand Oaks, CA 91360
Work Phone: 805-557-7774
E-mail: Sunil.Gupta@quintiles.com

Linfeng Xu, MSPH, Senior SP
Quintiles, Inc.
325 E. Hillcrest Drive Suite 200
Thousand Oaks, CA 91360
Work Phone: 805-557-7727
E-mail: Linfeng.xu@quintiles.com