ABSTRACT
This paper introduces an extremely fast and simple implementation of the survey cell collapsing process. Prior implementations had used either several SQL queries or numerous DATA step arrays, with multiple data reads. This new approach uses a single hash object with a maximum of two data reads. The hash object provides an efficient and convenient mechanism for quick data storage and retrieval (sub-second total run time).

INTRODUCTION
While working at the US Census Bureau, a friend and colleague from a different division at the Census Bureau sent me an email, to my personal email address, seeking help with implementing Survey Cell Collapsing solution.

All I got from him are the following collapsing requirements with a spreadsheet of artificial survey results data.

- Calculate the Adjustment factor = popct/wegt.
- Collapse any cell that has < 35 persons or adjustment factor < .67 or > 4 with the cell that is closest scale value.
- Collapse only within sex and race. Do not collapse <15 with 15+
- Collapse means adding the two cells unwegt and weighted cells and calculate the new scale value.

New scale value = ((cell1 scale_value*cell1 weight) + (cell2 scale_value*cell2 weight))/(weight1+weight2)
<table>
<thead>
<tr>
<th>Cell Number</th>
<th>Sex</th>
<th>Race</th>
<th>Age</th>
<th>Scale Value</th>
<th>unwegt</th>
<th>popct</th>
<th>wegt</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Male</td>
<td>White Alone</td>
<td>&lt;1</td>
<td>601</td>
<td>716</td>
<td>2503603</td>
<td>1552194</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>1</td>
<td>603</td>
<td>12</td>
<td>41959.82682</td>
<td>26014.42458</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>16</td>
<td>Male</td>
<td>White Alone</td>
<td>18-19</td>
<td>18</td>
<td>741</td>
<td>2698875</td>
<td>861641</td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td>20-21</td>
<td>27</td>
<td>683</td>
<td>2737338</td>
<td>1915384</td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
<td>22-24</td>
<td>29</td>
<td>1065</td>
<td>4427703</td>
<td>4179955</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>32</td>
<td></td>
<td></td>
<td>85+</td>
<td>106</td>
<td>260</td>
<td>1446173</td>
<td>358819</td>
</tr>
<tr>
<td>33</td>
<td>Female</td>
<td>White Alone</td>
<td>&lt;1</td>
<td>701</td>
<td>358</td>
<td>1122112</td>
<td>504566</td>
</tr>
<tr>
<td>34</td>
<td></td>
<td></td>
<td>1</td>
<td>703</td>
<td>321</td>
<td>1201937</td>
<td>1779974</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>48</td>
<td>Female</td>
<td>White Alone</td>
<td>18-19</td>
<td>118</td>
<td>694</td>
<td>2725271</td>
<td>2224527</td>
</tr>
<tr>
<td>49</td>
<td></td>
<td></td>
<td>20-21</td>
<td>127</td>
<td>625</td>
<td>2646054</td>
<td>1996393</td>
</tr>
<tr>
<td>50</td>
<td></td>
<td></td>
<td>22-24</td>
<td>129</td>
<td>964</td>
<td>3953523</td>
<td>1148473</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>64</td>
<td></td>
<td></td>
<td>85+</td>
<td>206</td>
<td>475</td>
<td>2613759</td>
<td>2111431</td>
</tr>
<tr>
<td>65</td>
<td></td>
<td></td>
<td>0-4</td>
<td>821</td>
<td>35</td>
<td>192592.7684</td>
<td>155579.1263</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>71</td>
<td>Male</td>
<td>Black Alone</td>
<td>30-34</td>
<td>345</td>
<td>34</td>
<td>188427.0097</td>
<td>175348.7282</td>
</tr>
<tr>
<td>72</td>
<td></td>
<td></td>
<td>35-39</td>
<td>347</td>
<td>489</td>
<td>2023880</td>
<td>585685</td>
</tr>
<tr>
<td>73</td>
<td></td>
<td></td>
<td>40-44</td>
<td>349</td>
<td>302</td>
<td>1011554</td>
<td>106363</td>
</tr>
<tr>
<td>74</td>
<td></td>
<td></td>
<td>45-49</td>
<td>355</td>
<td>22</td>
<td>73689.36424</td>
<td>7748.298013</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>77</td>
<td></td>
<td></td>
<td>65+</td>
<td>365</td>
<td>378</td>
<td>1288817</td>
<td>542540</td>
</tr>
</tbody>
</table>

*Figure 1: Artificial survey results table*
SOLUTION

Considering the requirements I was given, I implemented a SAS® macro “collapse_data” that treats some of the listed requirements as input parameters, in order to generalize it and allow it to be used with other requirements/conditions.

Figure 2: High-level logic flow of the collapse_data macro

MACRO COLLAPSE_DATA DETAILS

%macro collapse_data (  
p_inDsName= /* Input Survey Table */  , p_outDsName= /* Output Collapsed Table */  , p_classVars= /* By Group Variables */  , p_prntVar= /* By Group Boundary variable */  , p_minLimitCondition= /* Collapsing Condition */  , p_mapDsName= /* Collapsed Cells Mapping Output Table */  );

%LOCAL  
  l_eTime  
  l_msg  
  l_rc  
  l_rTime  
  l_sTime  
  l_needCollapse  
  l_needCollapseCnt
Using a SAS® Hash Object to Speed and Simplify
the Survey Cell Collapsing Process

l_hashKeyVars
l_currVarsLengthStmt
l_prevVarsLengthStmt
l_currVarsAssignStmt
l_prevVarsAssignStmt
l_currVarsAssignStmt2
l_prevVarsAssignStmt2
;

/********** BEGIN -- Main Macro Processing **********/

/** Record Starting Time **/
%let l_sTime = %sysfunc(time());

/* First - Examine we need to Perform the Collapsing Process */
%let l_needCollapseCnt = 0;
PROC SQL NOPRINT;
  SELECT STRIP(PUT(COUNT(*),best.))
    ,STRIP(PUT(cell_number,BEST.))
  INTO :l_needCollapseCnt
    ,:l_needCollapse separated by ' ' 
  FROM &p_inDsName 
  WHERE &p_minLimitCondition
  ;
QUIT;
%put ;
%put >>> ----------------------------------------------- ;
%put >>> Cells must be Collapsed Count = &l_needCollapseCnt;
%PUT >>> Cells must be Collapsed = &l_needCollapse;
%put >>> ----------------------------------------------- ;
%put ;
%if(&l_needCollapseCnt EQ 0) %then
  %do;
    %let l_rc = 1;
    %let l_msg = ERROR>>> collapse_data : Input table did not contain collapsable records! Process terminated.;
    %goto exit;
  %end;
/* Initialize Dynamic Variables and Statements to be used later */
%let l_libName = WORK;
%let l_dsName  = %UPCASE(%superq(p_inDsName));
%if (%INDEX(%superq(l_dsName),%str(.)) GT 0) %then
  %do;
    %let l_libName = %SCAN(%superq(l_dsName),1,%str(.));
    %let l_dsName  = %SCAN(%superq(l_dsName),2,%str(.));
  %end;
PROC SQL NOPRINT;
SELECT CATS('curr_',name)||' '|| CASE WHEN type = 'char' then '$' ELSE '' END || STRIP(PUT(length,best.)) ,CATS('prev_',name)||' '|| CASE WHEN type = 'char' then '$' ELSE '' END || STRIP(PUT(length,best.)) ,CATS('curr_'.name,' = ',name,',';') ,CATS('prev_'.name,' = ',name,',';') ,CATS(name,' = curr_'.name,';') ,CATS(name,' = prev_'.name,';') INTO :l_currVarsLenghtStmt separated by '' ,:l_prevVarsLenghtStmt separated by '' ,:l_currVarsAssignStmt separated by '' ,:l_prevVarsAssignStmt separated by '' ,:l_currVarsAssignStmt2 separated by '' ,:l_prevVarsAssignStmt2 separated by '' FROM DICTIONARY.COLUMNS WHERE LIBNAME = "&l_libName" AND MEMNAME = "&l_dsName" ; QUIT;

/* scale_Value unwegt popct wegt origCellNumber */
%let l_hashKeyVars= %str(%')%sysfunc(tranwrd(%superq(p_classVars), %str( ),%str('%',%')))%str('%');

DATA _NULL_;
/* Define all the variables in the PDV */
if 0 then SET &p_inDsName;

LENGTH &l_currVarsLenghtStmt &l_prevVarsLenghtStmt usedCellsToCollapseWith $3000 recN 8;

RETAIN usedCellsToCollapseWith;

FORMAT scale_Value unwegt popct wegt 16.;

if (_n_ = 1) then
  do;
    /* declare hash, and load the data set into it */
    dcl hash recs (dataset:="#p_inDsName",ordered:'a',hashexp: 16) ;
    recs.definekey (%unquote(&l_hashKeyVars)) ; /* key table with KEY variable */
    recs.definedata (all:'yes') ; /* store record data */
    recs.definedone () ; /* check validity and instantiate object */

    declare hiter hiRecs("recs");
end;

call missing(of _all_);

DO recN=1 by 1 UNTIL(LAST.&p_prntVar);

SET &p_inDsName END=eof;
BY     &p_classVars;

/* Load an updated copy of the record */
rc = recs.find();

&l_currVarsAssignStmt

if (MISSING(origCellNumber)) then
  origCellNumber = STRIP(PUT(cell_number,BEST.));
if (MISSING(curr_origCellNumber)) then
  curr_origCellNumber = STRIP(PUT(curr_cell_number,BEST.));
if (_N_ GT 1) then
do;
  /* Reset the Hash Iterator position to current record */
  if (hiRecs.setcur() = 0) then
    do;
      rc = hiRecs.PREV(); /* Load previous record into the PDV */
        &l_prevVarsAssignStmt
    end;
  end;
/* Check the Collapsing Condition */
if (&p_minLimitCondition) then
do;
  /* Reset the Hash Iterator position to current record */
  if (hiRecs.setcur() = 0) then
    do;
      rc = hiRecs.next(); /* Load next record into the PDV */
        if (MISSING(origCellNumber)) then
          origCellNumber = STRIP(PUT(cell_number,BEST.));
    end;
/* Find the suitable collapsing cell based on closest scale_value.*/
if (rc = 0) then
    next_diff = abs(scale_Value - curr_scale_Value);
prev_diff = abs(curr_scale_Value - prev_scale_Value);

if (((prev_&p_prntVar = curr_&p_prntVar) AND (prev_diff LT next_diff) OR
    ((prev_diff EQ next_diff) AND (prev_unwegt LT unwegt))
) then
    do;
        prev_origCellNumber = CATX(‘,’ ,prev_origCellNumber,curr_origCellNumber);
        prev_scale_value = ((curr_scale_value*curr_wegt)+(prev_scale_value*prev_wegt))/(curr_wegt+prev_wegt);
        prev_unwegt = SUM(curr_unwegt,prev_unwegt);
        prev_wegt = SUM(curr_wegt,prev_wegt);
        prev_popct = SUM(curr_popct,prev_popct);
end;

&l_prevVarsAssignStmt2

end;
else if (((curr_&p_prntVar = &p_prntVar) AND (prev_diff GT next_diff) OR
    ((prev_diff EQ next_diff) AND (prev_unwegt GT unwegt))
) then
    do;
        origCellNumber = CATX(‘,’ ,origCellNumber,curr_origCellNumber);
        scale_value = ((curr_scale_value*curr_wegt)+(scale_value*wegt))/(curr_wegt+wegt);
        unwegt = SUM(curr_unwegt,unwegt);
        wegt = SUM(curr_wegt,wegt);
        popct = SUM(curr_popct,popct);
end;

&l_currVarsAssignStmt2

end;

/* Add/Replace the record in the Hash Object */
recs.replace();

/* Need to Remove Currently collapsed record from the Hash Object */
if (((prev_&p_prntVar = curr_&p_prntVar) OR (curr_&p_prntVar = &p_prntVar)) then
    do;
        &l_currVarsAssignStmt2
        if (recs.find() = 0) then
            do;
                rc= recs.remove();
            end;
        end;
end;

end;
Using a SAS® Hash Object to Speed and Simplify
the Survey Cell Collapsing Process  

else
do;
   /* Replace the record in the Hash Object */
   recs.replace();
end;
END;

if (eof) then
   recs.output(dataset:"&p_outDsName");
RUN;

/* Produce Cell_number & Collapse Index mapping data set */
DATA &p_mapDsName(keep=index);
LENGTH index 3;
SET &p_outDsName(keep=cell_number origCellNumber);

index = _n_;  
i=1;
do until(scan(origCellNumber,i,',' eq "");
   OUTPUT;
   i+1;
end;
RUN;

%goto finished;

%exit:
   %put *** collapse_data ***;
   %put *** l_RC must be zero (0). ***;
   %put *** l_RC= &l_RC. ***;
   %put *** &l_MSG ***;
   %put *** collapse_data ***;

%finished:
   /** Record Finish Time **/
   %let l_eTime = %sysfunc(time());

   %if (%superq(l_sTime) NE ) %then
   %do;
      /* Calculate Run Time and display it */
      %let l_rTime = %sysfunc(putn(%sysevalf(&l_eTime - &l_sTime),time12.2));
      %put;
      %put >>> -----------------------------------------------;
      %put >>> collapse_data :>>> Total RunTime = &l_rTime;
      %put >>> -----------------------------------------------;
      %put;
   %end;
Using a SAS® Hash Object to Speed and Simplify the Survey Cell Collapsing Process

%mend collapse_data;

/* Sample macro execution call */
%cparse_data (p_inDsName=myLib.input, p_outDsName=work.lev1
, p_classVars=%STR(cell_number sex age), p_prntVar=sex
, p_minLimitCondition=%str(unwegt LT 35 OR (popct/wegt) LT 0.67 OR (popct/wegt) GT 4)
, p_mapDsName=work.map1);

/* Sample Output */
>>> ┌─────────────────────────────────────────────────────────────────────────┐
>>> │ Cells must be Collapsed Count = 25                                 │
>>> │ Cells must be Collapsed = 2 6 10 14 20 32 39 41 42 43 54 59 60 66 71 73 74 75 80 84 87 93 98 102 107 │
>>> └─────────────────────────────────────────────────────────────────────────┘

NOTE: There were 108 observations read from the data set MYLIB.INPUT.
NOTE: The data set WORK.LEV1 has 83 observations and 9 variables.
NOTE: There were 108 observations read from the data set MYLIB.INPUT.
NOTE: DATA statement used (Total process time):
    real time 0.02 seconds
    cpu time 0.03 seconds

NOTE: There were 83 observations read from the data set WORK.LEV1.
NOTE: The data set WORK.MAP1 has 108 observations and 1 variables.
NOTE: DATA statement used (Total process time):
    real time 0.01 seconds
    cpu time 0.01 seconds

>>> collapse_data :>>> Total RunTime = 0:00:00.10

>>> └─────────────────────────────────────────────────────────────────────────┘
### Using a SAS® Hash Object to Speed and Simplify the Survey Cell Collapsing Process

<table>
<thead>
<tr>
<th>Cell_Number</th>
<th>Sex</th>
<th>Race</th>
<th>Age</th>
<th>Scale_Value</th>
<th>unwegt</th>
<th>popct</th>
<th>wegt</th>
<th>origCellNumber</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>&lt;1</td>
<td>601.033</td>
<td>728</td>
<td>254563</td>
<td>1578208,1</td>
<td>2</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Male</td>
<td>White</td>
<td>4</td>
<td>611.0277</td>
<td>625</td>
<td>2635370</td>
<td>1475543,5</td>
<td>6</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Male</td>
<td>White</td>
<td>8</td>
<td>620.1548</td>
<td>323</td>
<td>1360688</td>
<td>470413.7,9</td>
<td>10</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Male</td>
<td>White</td>
<td>16-17</td>
<td>15.8706</td>
<td>995</td>
<td>3951597</td>
<td>1236326,15</td>
<td>14</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Male</td>
<td>White</td>
<td>25-29</td>
<td>47.44487</td>
<td>3316</td>
<td>13798844</td>
<td>3474992,19</td>
<td>20</td>
</tr>
<tr>
<td>21</td>
<td>Male</td>
<td>White</td>
<td>35-39</td>
<td>57</td>
<td>1573</td>
<td>5866860</td>
<td>6241955</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>Male</td>
<td>White</td>
<td>80-84</td>
<td>104.6845</td>
<td>702</td>
<td>3161600</td>
<td>1048419,31</td>
<td>32</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>Female</td>
<td>White</td>
<td>5</td>
<td>712</td>
<td>303</td>
<td>1211654</td>
<td>954199</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>Female</td>
<td>White</td>
<td>7</td>
<td>716.9633</td>
<td>980</td>
<td>3930691</td>
<td>962505.2,40,39,42,41</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>Female</td>
<td>White</td>
<td>12 to13</td>
<td>729.0286</td>
<td>1678</td>
<td>6465308</td>
<td>9633235,44,43</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>Female</td>
<td>White</td>
<td>14</td>
<td>733</td>
<td>335</td>
<td>1248982</td>
<td>615495</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>Female</td>
<td>White</td>
<td>30-34</td>
<td>149</td>
<td>1834</td>
<td>6524873</td>
<td>2809038</td>
<td></td>
</tr>
<tr>
<td>53</td>
<td>Female</td>
<td>White</td>
<td>35-39</td>
<td>157.1725</td>
<td>3382</td>
<td>13191510</td>
<td>9003230,53,54</td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>Female</td>
<td>White</td>
<td>45-49</td>
<td>163</td>
<td>1887</td>
<td>7439013</td>
<td>2740975</td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>Female</td>
<td>White</td>
<td>50-54</td>
<td>165</td>
<td>2026</td>
<td>7818318</td>
<td>2342809</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 3**: Collapsed artificial survey results table

### CONCLUSION

The Hash Object and the Hash Iterator Object are two component objects provided by SAS® for use in a Data Step. They enable us to rapidly store, search, and retrieve data based on lookup keys efficiently there by cutting down the run time.

The predefined attributes and methods of these two component objects, provide great functionality out of the box, simplify and reduce the length of the SAS® programs, thereby enhance their maintainability.
REFERENCES


CONTACT INFORMATION

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

Ahmed Al-Attar
AnA Data Warehousing Consulting, LLC.
McLean, VA 22101
Cell Phone: 703-477-7972
E-mail: ahmed.al-attar@anadwc.com
Web: www.anadwc.com

TRADEMARKS

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.