Paper 74

Use PROC SQL to Collapse Cells – It's Easy

Tamara Hagemeier, Yue-Hwa Chang, Bureau of Labor Statistics[1]

I. Introduction

During data processing it may become necessary to group data records in order to perform a procedure. The grouped records form cells that may be used to test some general condition about the grouped records. If the quantity of data within a cell is insufficient then cells are combined until sufficient data exist for valid inference. The Consumer Expenditure Survey at the Bureau of Labor Statistics has developed a program to collapse cells by using SAS® PROC SQL. The PROC SQL program collapses cells in an order defined by the user, keeps a record of all collapsed cells, and retains the original and the collapsed cell values.

There are three types of input parameters to the collapsing program: classification variables, sufficiency criteria, and collapsing order. All parameters are supplied by the user and easily changed. Survey data is put into cells defined by classification variables such as, family size, tenure, and race. The sufficiency criteria may take on many forms such as, the number of records or units in a cell, the total weight of the cell, the ratio of one type of record to another, or any combination of above. The collapsing order indicates which cells should be combined if a cell fails the sufficiency criteria.

II. Cell Defining Variables

For the Consumer Expenditure Survey application, each record is placed into cells based on tenure, family size, and race. Table 1 shows the 16 possible cells. Each record should be categorized into one and only one of the sixteen cells.

Table 1

Tenure		Owner				Renter			
Family Size		1	2	3-4	5+	1	2	3-4	5+
Race	Black	1	3	5	7	9	11	13	15
	Non-Black	2	4	6	8	10	12	14	16

The data step below classifies a record into a cell by formatting family size, tenure, and race.

```
DATA DEFINCEL (DROP = TENURECU CUSIZE
CURACE);
     SET &FMLYIN;
     TENURE = PUT (TENURECU, $TENU.);
     FAMSIZE = PUT (CUSIZE, SIZE.);
     RACE = PUT (CURACE, $RACE.);
RUN;
```

III. Sufficiency Criteria

For the Consumer Expenditure Survey application, cells are combined if the total number of units within the cell, or a weighted ratio exceeds a given value. Cells are collapsed in a ***specific hierarchical order***. The first cell defining variable to be collapsed is family size. Once all the family size cells have been established, the tenure cells within the remaining family size cells are reviewed for collapsibility. Finally, the race cells within the remaining family size/tenure cells are reviewed, and the insufficient cells are collapsed. Cells should be collapsed if any one of the follow (4) conditions is met:

1. The cell contains more than a total of 20 units and the weighted ratio is greater than 2.00.

2. The cell contains a total of 11-20 units and the weighted ratio is greater than 1.75.

3. The cell contains less than or equal to a total of 10 units and the weighted ratio is greater than 1.55.

4. If the weighted ratio results in division by zero the cell should be collapsed.

[1] Any opinions in this paper are those of the authors and do not constitute policy of the Bureau of Labor Statistics.

The total number of units and weighted ratio are recalculated for each set of collapsed cells until none of the collapsing criteria are met. Only then does the process move to the next step. Hence, this is an iterative process.

The sufficiency criteria (except #4) have been implemented by SAS® format, which is originally stored as a Microsoft® Excel spreadsheet and exported into SAS®.

FMTNAME	START	END	LABEL
MAXRATIO	1	10	1.55
	11	20	1.75
	21	999999999	2.00

IV. Collapsing Process

To identify cells that should be collapsed, calculate the total number of units and the weighted ratio for each of the sixteen cells. Use PROC SQL to select the appropriate cell defining variables (from a set up table) and GROUP BY the same cell defining variables. For example, use the SUM function to sum individual units for family size 1, tenure 1, and race 1 to get the total number of units for cell 1. The set up table should contain the cell defining variables (i.e. family size, tenure, and race) and variables used to calculate total units and weight ratio. Here we have added the variables NUMUNITS, NUMERWT, and DENOMWT to the data set DEFINCEL. The code to assign cell defining values is shown in section II above.

To calculate the total number of units and the weighted ratio for each of the sixteen cells:

```
PROC SQL;
CREATE TABLE GROUP16 AS
SELECT
DISTINCT TENURE, FAMSIZE, RACE,
SUM (NUMUNITS) AS TOTUNITS,
SUM (NUMERWT) AS TOTNUMWT,
SUM (DENOMWT) AS TOTDENWT
FROM DEFINCEL
GROUP BY TENURE, FAMSIZE, RACE;
QUIT;
```

Compare the total number of units and weighted ratio to sufficiency criteria:

```
IF (TOTUNITS > 0) THEN
  DO;
     WTRATIO = TOTNUMWT / TOTDENWT;
     IF (WTRATIO >
INPUT(PUT(TOTUNITS,MAXRATIO.),6.5)) THEN
       DO;
         OUTPUT CELLCLPS;
/* CELL SHOULD BE COLLAPSED */
       END;
```

Collapsing is necessary if any of the cells meet one of the above sufficiency criterion. If at least one of the cells meets the above criteria then calculate the overall number of units for cells 1-16 combined and the corresponding weighted ratio, which is done by performing the summary calculations without the GROUP BY statement.

```
PROC SQL;
CREATE TABLE GROUPALL AS
SELECT
SUM (NUMUNITS) AS TOTUNITS,
SUM (NUMERWT) AS TOTNUMWT,
SUM (DENOMWT) AS TOTDENWT
FROM DEFINCEL;
QUIT;
```

If the overall calculations do not meet any of the collapsing criteria then calculate the number of units and weighted ratio within family size, this is done by performing the summary calculations with the GROUP BY family size statement.

```
PROC SQL;
CREATE TABLE GRPSIZE AS
SELECT
DISTINCT FAMSIZE,
SUM (NUMUNITS) AS TOTUNITS,
SUM (NUMERWT) AS TOTNUMWT,
SUM (DENOMWT) AS TOTDENWT
FROM DEFINCEL
GROUP BY FAMSIZE;
QUIT;
```

If collapsing is necessary, then follow the steps outlined below.

If any of the family size calculations meet the collapsing criteria then combine the appropriate family sizes, ***keeping race and tenure fixed***, together using the following instructions:

- If family size 1 should be collapsed, combine with size 2. If further collapsing is needed combine with size 3-4, and finally combine with size 5+.
- If family size 2 should be collapsed, combine with size 1. If further collapsing is needed combine with size 3-4, and finally combine with size 5+.
- If family size 3-4 should be collapsed, combine with size 5+. If further collapsing is needed combine with size 2, and finally combine with size 1.

- If family size 5+ should be collapsed, combine with size 3-4. If further collapsing is needed combine with size 2, and finally combine with size 1.

For example, if the calculations for the combined cells 1, 2, 9, and 10 result in 25 total units and weighted ratio 2.55 then cell 1 is combined with cell 3, cell 2 is combined with cell 4, cell 9 is combined with cell 11, and cell 10 is combined with cell 12 (keeping race and tenure fixed), as shown in table 2.

Table 2

Tenure		Owner			Renter		
Family Size		1-2	3-4	5+	1-2	3-4	5+
Race	Black	1 3	5	7	9 11	13	15
	Non-Black	2 4	6	8	10 12	14	16

To combine the appropriate family sizes means the individual records that were classified into different cells (i.e. family size 1 and family size 2) will be merged into one cell (i.e. family size 1-2). There are several ways to combine cells. Here we combined cells by forcing family sizes to be the same. For example, if family size 1 should be collapsed, then family size 2 will be changed to 1, if further collapsing is needed, family size 3-4 will be changed to 1. Then use PROC SQL GROUP BY to perform calculations within the newly created cell.

While family sizes are changed during collapsing, we must trace the original value of each record within the collapsed cell so the final weight ratio can be assigned to individual records properly. We solved this by creating additional columns: transaction family size, transaction tenure, and transaction race prior to any collapsing process.

```
DATA BEFCLPS;
    LENGTH TRANSIZE TRANTENU TRANRACE $1;

    SET CELLCLPS;
    TRANSIZE = FAMSIZE;
    TRANTENU = TENURE;
    TRANRACE = RACE;
RUN;
```

The transaction values are set to be the same value as the original family size, tenure, and race and are updated during the collapsing process. The original family size, tenure, and race become a bridge between individual record and the transaction cells. The transaction columns are also used for bookkeeping. It is possible that the last value of the variable will need to be collapsed. If previous values have already been collapsed the final value should be collapsed into the combined cell. With these additional columns, it is easy to find the collapsed family size from the original family size.

Link the final weight ratio to the original 16 family size, tenure, and race cells through transaction family size, transaction race, and transaction tenure (if cells are combined, the final weight ratio should be the same for all records in the combined cells):

```
PROC SQL NOPRINT;
  CREATE TABLE CELFINWT AS
    SELECT DISTINCT A.TENURE, A.FAMSIZE,
         A.RACE, B.WTRATIO
       FROM BACOLPS A, FINWTRAT B
       WHERE A.TRANSIZE = B.TRANSIZE AND
             A.TRANTENU = B.TRANTENU AND
             A.TRANRACE = B.TRANRACE;
QUIT;
```

Note: Above BACOLPS (Before After Collapsing) SAS® data set in the FROM clause contains records with the before collapsing cell values and the after collapsing cell values. The SAS® data set FINWTRAT contains only one record for each unique cell after collapsing is complete. Each FINWTRAT record contains the value of the weight ratio for the final combined cells.

The cell final weight ratio is then merged with individual records by matching the original family size, tenure, and race values.

To collapse family size in the specified order, we store the collapsing order in a SAS® data set with variable name suffix as the order. For example, the SAS® data set will contain rows as family size = ‘1’, clps1=’2’, clps2=’3-4’, clps3=’5+’; family size=’2’, clps1=’1’; clps2=’3-4’;clps3 =’5+’; etc.

```
DATA CLPSORDR;
  SIZE = '1';
  CLPS1 = '2';
  CLPS2 = '3';
  CLPS3 = '4';
  OUTPUT CLPSORDR;

  SIZE = '2';
  CLPS1 = '1';
  CLPS2 = '3';
```

```
              CLPS3 = '4';
              OUTPUT CLPSORDR;

              SIZE = '3';
              CLPS1 = '4';
              CLPS2 = '2';
              CLPS3 = '1';
              OUTPUT CLPSORDR;

              SIZE = '4';
              CLPS1 = '3';
              CLPS2 = '2';
              CLPS3 = '1';
              OUTPUT CLPSORDR;
            RUN;
```

If family size 2 needs to be collapsed, select the family size to combine with family size 2 (from clps&order; and loop order from 1 until the sufficiency criteria are met) from the SAS® data set where family size = family size that needs collapsing (i.e. ‘2’). We also avoided reprocessing. For example, family size 1 and 2 are both insufficient, while family size 1 and family size 2 data are combined during the processing of family size 1, family size 2 should not be processed again. Reprocessing is avoided by using PROC SQL join and PROC SQL delete.

```
LET ORDER = 1;
LET MORECLPS = YES;

DO WHILE ("MORECLPS" = "YES");

/* FIND CU SIZE TO COLLAPSE WITH */
/* CLPSZ = CUCOLP [&TARGETSZ, &ORDER];*/

  PROC SQL NOPRINT;
  SELECT CLPS&ORDER INTO : CLPSZ
  FROM CLPSORDR
  WHERE CUSIZE = "&NEEDCLP";

/* USE TRANSITION SIZE TO COLLAPSE */
/* SIZE MAY HAVE BEEN COMBINED PREVIOUSLY */

  SELECT TRANSIZE INTO : CMBSIZE
  FROM BACOLPS
  WHERE FAMSIZE = "&CLPSZ";

/* COMBINE SIZES BY UPDATING TO BE THE */
/*SAME VALUE */
  UPDATE BACOLPS
  SET TRANSIZE = "&NEEDCLP"
  WHERE TRANSIZE IN ("&CMBSIZE");

/* CHECK IF THE COMBINED WITH SIZE MEETS */
/* THE COLLAPSING CRITERIA IF TRUE,      */
/* DELETE IT TO AVOID REPROCESSING.      */

  DELETE
  FROM MEETCLPS
  WHERE TRANSIZE = "&CMBSIZE";
```

Once none of the family size calculations meet the above criteria then calculate the total number units and weighted ratio within tenure. If any of the tenure calculations meet the collapsing criteria then combine tenure within family size while ***keeping race fixed***. This is a similar process but is simpler because the variable tenure contains only two values, owner and renter.

Once all required collapsing has been completed, calculate the total number of units and weighted ratio within race. If any of the race calculations meet the collapsing criteria then combine the race cells within family size and tenure. This is a similar process but is simpler because the variable race contains only two values, black and non-black.

VI. Conclusion

While there are many software tools available it is important to select the tool that will not only accomplish the task but is easy to use and maintain. Before PROC SQL was available, there was no easy way to collapse cells. PROC SQL is flexible and easy to maintain which should please both the user and the programmer.

VII. Contact

For more information contact:

Tamara Hagemeier
2 Massachusetts Ave. NE
Suite 3650
Washington, DC 20212
hagemeier_t@bls.gov

Yue-Hwa Chang
2 Massachusetts Ave. NE
Suite 5935
Washington, DC 20212
chang_y@bls.gov