Data Resource Centre, University of Guelph

SAS – AGGREGATING MULTIPLE RECORDS INTO ONE WIDE RECORD

In some cases raw data files have multiple records that all belong to one larger overall unit. In the example below, there is one record in the raw data file for each member of the larger overall unit: *household*. A common requirement is to combine these multiple records into one single "wide" record that contains all the information for the overall unit. In the example for this tip sheet the requirement is to create one "wide" record containing all the information for all the members of the household. In SAS, this operation is referred to as *aggregation*.

In the example below: the SAS dataset **households** has information on the Household ID (variable: householdID), the Household member (variable: householdMember), the Household members' gender (variable: gender), and the Household member's age (variable: age). Note there are multiple observations for a given Household in the raw data.

This program illustrates how to create a new SAS dataset named: **singleRecord**, which has all the information for all the members of the household in one "wide" observation. In SAS terminology, the program aggregates the individual records for the Household into one wide record. Please refer to the "Code Notes" section below for an explanation of what the code is doing in those sections identified by a number in a **red-filled** circle.

```
* Aggregation: combining multiple observations into one wide record in SAS */

data households ;
      input householdID householdMember :$30.  gender :$1. age ;
      datalines ;
      23914 head F 47
      23914 child F 8
      11654 child M 6
      11654 spouse M 31
      11654 head F 30
      11654 child F 2
      91203 head M 57
      91203 spouse F 61
      ;
run ;
```

Step 1:
Create the SAS dataset: **households**

```
proc print data = households;
      title "Information on a sample of Canadian Households";
      title2 "Source file has one record for each member of each household" ;
run;
```

Step 2:
Print the dataset to illustrate it has multiple records for a household unit.

```
proc sort data = households ;
       by householdID ;
run ;
```

Step 3:
Sort the dataset by the overall unit identifier to group all records for a given household together.

```
data singleRecord ;
       set households ;
       by householdID ;

       /* if this is the first record for the household then initialize all variables for this household */
1      if first.householdID then do ;
             headGender = ' ';
             headAge = . ;
             spouseGender = ' ' ;
             spouseAge = . ;
             numberOfChildren = 0 ;
       end;

2      if householdMember = 'head' then do ;
             headGender = gender ;
             headAge = age ;
       end ;

2      if householdMember = 'spouse' then do ;
             spouseGender = gender ;
             spouseAge = age ;
       end ;

3      if householdMember = 'child' then do ;
             numberOfChildren = numberOfChildren + 1 ;
       end ;
```

Step 4:
Create the “wide” dataset **singleRecord**.
(this step is continued on the next page).

See the **Code Notes** section of this document for an explanation of what the code is doing in those areas identified with **red-filled** circles.

4

5

```
      /* if the last record for this household has been processed,
         output the single wide record to the dataset */

      if last.householdID then output ;

      retain numberOfChildren headGender headAge spouseGender spouseAge;
      keep householdID headGender headAge spouseGender spouseAge numberOfChildren ;
run ;
```

Step 4 continued

```
proc print data = singleRecord ;
      title "Information on a sample of Canadian Households";
      title2 "One record for the entire household";
run;
```

Step 5:
Print the "wide record" dataset to confirm the aggregation worked as expected.

```
proc means data = singleRecord ;
      title "Descriptive information on Canadian Households" ;
      var headAge spouseAge numberOfChildren ;
run ;
```

Step 6:
This step illustrates an optional procedure (PROC MEANS) that would be difficult to perform on "multiple records per unit" dataset *households*.

Output produced by this program's PROCS (PROCEDURES):

```
              Information on a sample of Canadian Households
       Source file has one record for each member of each household

                    household    household
              Obs      ID         Member      gender    age

               1      23914       head          F        47
               2      23914       child         F         8
               3      11654       child         M         6
               4      11654       spouse        M        31
               5      11654       head          F        30
               6      11654       child         F         2
               7      91203       head          M        57
               8      91203       spouse        F        61
```

Output produced by Step 2
(Print the dataset to illustrate it has multiple records for a household unit.)

```
              Information on a sample of Canadian Households
                    One record for the entire household

                                                                 number
          household    head      head    spouse    spouse          Of
    Obs      ID       Gender     Age     Gender     Age       Children

     1      11654       F         30       M         31           2
     2      23914       F         47                  .           1
     3      91203       M         57       F         61           0
```

Output produced by Step 5
(Print the "wide record" dataset to confirm the aggregation worked as expected.)

```
                          Descriptive information on Canadian Households


                                  The MEANS Procedure

Variable              N          Mean         Std Dev         Minimum         Maximum
headAge               3    44.6666667      13.6503968      30.0000000      57.0000000
spouseAge             2    46.0000000      21.2132034      31.0000000      61.0000000
numberOfChildren      3     1.0000000       1.0000000               0       2.0000000
```

Output produced by Step 6 (This step illustrates an optional procedure (PROC MEANS) that would be difficult to perform on "multiple records per unit" dataset *households*.)

Code Notes: (explanation of what the code is doing in those areas identified by a number in a red-filled circle)

1. The code in this section is executed when the first record for any household is encountered. In this section you do two things: (a) name all the variables to appear on the "wide record" and (b) initialize these variables – i.e set them all to missing.

2. Add additional IF code blocks for each possible value of *householdMember* you wish to map to variables in the wide record. In this example we have two IF code blocks. One is mapping information on the head of the household to variables *headGender* and *headAge.*, The second IF code block is mapping information on the spouse in the household to variables *spouseGender* and *spouseAge*.

3. In this example, we're just counting the number of children in the household, not mapping each child's information as a separate set of variables on the wide record as we did in area 2 above.

4. Write out the wide observation to the dataset **singleRecord** when you have processed the last observation for this household from the **households** dataset.

5. The RETAIN statement tells SAS to remember the values of these variables as a new observation is read in from the dataset **households**.

6. The KEEP statement names the variables that should appear in the new dataset being constructed (**singleRecord**). Any variables in the data step not named on the KEEP statement are not written to the dataset.

Tips:

- All records in the multiple-records-per-unit dataset (dataset: **households** in this example) must have a common variable that uniquely identifies the overall unit the record belongs to. (in this example, the variable is **householdID**).
- All records in the multiple-records-per-unit dataset (dataset: **households** in this example) must have a common variable that uniquely identifies the role the observation plays in the larger overall unit. (in this example, this variable is **householdMember –** this variable indicates whether the observation is for the head of the household, the spouse, a child, etc.).
- See Step 3: the multiple-records-per-unit dataset (dataset: **households** in this example) must be sorted by the overall unit identifier (in this example, the variable **householdID**) for the code in Step 4 to work.
- It is helpful to print the aggregated dataset (**singleRecord**) to verify that the aggregation worked as expected (see Step 5 above).
- Always examine the SAS log for any error messages or warnings.