

Navigating Between Data Levels: A Tutorial to Working with Unit of Analysis Differences in Survey Research Data

Chris Becker, Abt Associates Inc, Chicago IL

ABSTRACT

Working with multiple data sets can be a challenge when the data are not all on the same level. For example, imagine two data sets, the first one containing information about households and the second one containing information about persons in those households. Because households may be comprised of more than one person, a household record may have several person records associated with it. Thus, the relationship between the household data and the person data is one-to-many. Extracting data from one data set with the intent of combining it with the other data set can be tricky in this situation. Valuable information can be lost in the process. This paper highlights how SAS makes the navigation from one level to the other possible. A variable in the person-level data set can be converted to a household level variable by first creating a subset, sorting the subset on the appropriate variables, and then using the special SAS *first.dot* or *last.dot* variables. The resulting data set can then be merged with the household file. In turn, a variable from the household data set can be extracted and combined with the person-level data set with the proper usage of merging.

INTRODUCTION

In survey research, data are often collected about households and about persons living in households. This data can be stored in two files, one containing information on the households and the other containing information on persons in the households. Since households can be comprised of more than one person, the person level dataset is likely to have more records than the household dataset. So the relationship between these two datasets is not a one-to-one, but instead is a one-to-many. For every household record, there is the potential to have multiple person records. When information from one dataset needs to be combined with the other dataset, the difference in levels must be taken into account. Manipulating datasets that are on different levels can be tricky. SAS allows users to work with these datasets in a meaningful way. To help illustrate how navigation between levels is possible, this paper will make reference to two datasets, *household.sasdat* and *person.sasdat*. The household dataset contains the following variables:

- HHID: household identification number
- INCOME: household income
- ZIPCODE: zipcode of household
- TELEPHONE: number of working telephone lines in household

The person dataset contains these variables:

- HHID: household identification number
- PERSONID: person identification number within household
- AGE: age in years of person
- EDUC: highest educational level completed

This paper will demonstrate how to take the ZIPCODE variable from the household dataset and merge it onto the person dataset, such that each person record will have a zipcode value associated with it. Also, we will create a new variable in the household dataset that contains the age of the oldest person living in the household. That information will be extracted from the person dataset and merged onto the household dataset.

HOUSEHOLD LEVEL TO PERSON LEVEL NAVIGATION

The household dataset contains the variable ZIPCODE. We are interested in taking this variable and adding it to the person dataset such that each record in the person dataset would have a zipcode value. There are two basic steps needed to accomplish this task. First, the zipcode data must be extracted from the household dataset. Second, the zipcode data must be combined with the person dataset.

CREATING A SUBSET

Extracting ZIPCODE from the household dataset involves creating a subset. A subset of a dataset is a new dataset that contains only a portion of the data from the original dataset. For instance, a subset can contain only a few of the variables or certain observations, i.e. records, from the original dataset. We want to create a subset of the household dataset containing HHID and ZIPCODE. HHID is needed later on to correctly match the zipcodes with the appropriate person records. The subset is created with the data step and the *keep=* option.

```
data household_subset;  
set household (keep=HHID ZIPCODE);  
run;
```

The new dataset *household_subset.sasdat* contains the variables HHID and ZIPCODE. The *keep=* option tells SAS to only retain the variables listed when executing the data step. All variables contained in *household.sasdat* that are not listed in the *keep=* option will not be copied into the new dataset. Depending on how many variables will be retained, the *drop=* option can be used instead of *keep=*. This option tells SAS to exclude the listed variables during the creation of the new dataset. When working with datasets containing few variables, the *keep=* and *drop=* options can be used interchangeably. The following code will yield the same result as the above code.

```
data household_subset;  
set household (drop=INCOME TELEPHONE);  
run;
```

MERGING TWO DATASETS

The relationship between `household.sasdat` and `person.sasdat` is one-to-many because for every household record, there can be many person records. To combine the person dataset with the newly created household subset, we will use the one-to-many data step merge. In order to merge two datasets, they must contain a common variable. Our datasets both have the `HHID` variable, which is the household identification number. Before datasets are merged, they need to be sorted on their common variable. `Proc sort` will accomplish that task. The `by` statement specifies which variable to sort the dataset on. By default, `proc sort` will sort the dataset in ascending order.

```
proc sort data=person;
  by HHID;
run;
proc sort data=household_subset;
  by HHID;
run;
```

Once the datasets are properly sorted, we can merge them.

```
data new_person;
  merge person (in=a) household_subset (in=b);
  by HHID;
  if a;
run;
```

The `if` statement is used to specify that only records with an `HHID` contained in `person.sasdat` will be written to the new dataset. If an `HHID` is found only in the household dataset, we do not want a new record created in `new_person.sasdat` that would only have values in `HHID` and `ZIPCODE`. The `new_person` dataset will contain the same number of observations as the person dataset, but there will be one extra variable, `ZIPCODE`. Each observation that has the same `HHID` will have the same value in `ZIPCODE`. Data from the household level dataset has successfully been extracted and transferred to the person level dataset by means of subsetting and merging. We will now demonstrate how the navigation from the person level to the household level is possible.

REVERSE NAVIGATION: GOING FROM PERSON TO HOUSEHOLD LEVEL

Our next goal is to create a new variable in the household dataset that contains the age of the oldest person living in the household. Three steps are required this time to complete the task. First, the `AGE` variable must be extracted from the person dataset. Next, the person level `AGE` variable needs to be transformed into a household level variable. Finally, the new age data must be combined with the household dataset.

CREATING A SUBSET

As was done earlier with the household dataset, a subset of `person.sasdat` is created by using the `keep=` option in the data step. The two variables that are kept are `HHID` and `AGE`.

```
data person_subset;
  set person (keep=HHID AGE);
run;
```

The new dataset `person_subset.sasdat` contains `HHID` and `AGE`.

TRANSFORMING A PERSON LEVEL VARIABLE TO A HOUSEHOLD LEVEL VARIABLE

For every unique `HHID` in `person_subset.sasdat`, we want to keep the highest value of `AGE`. In order to accomplish this task, we will sort our dataset on two variables. `Person_subset.sasdat` needs to be ordered not only by `HHID` but also by `AGE`. The ages within each household will be sorted in ascending order, from youngest to oldest. We can get our dataset in the desired order by using only one `proc sort` because SAS allows multiple variables to be listed in the `by` statement. Since we want the sorting to take place first on `HHID` and then on `AGE`, `HHID` is listed before `AGE` in the `by` statement. The following code is used to sort `person_subset.sasdat`.

```
proc sort data=person_subset;
  by HHID AGE;
run;
```

Partial listing of `person_subset.sasdat` after sorting:

HHID	AGE
1	54
1	57
2	24
3	6
3	35
3	37

If we examine the partial listing of `person_subset.sasdat`, we notice that common `HHIDs` are grouped together and that within each unique `HHID`, the ages of household members are sorted in ascending order. Thus, the last observation of each `HHID` contains the age of the oldest person in the household. We will use that knowledge to transform the `AGE` variable into a household level variable with the help of the `last.dot` variable. This temporary variable can be used in conjunction with an `if` statement to keep only the last observation of each `HHID`. A `by` statement must be used for the `last.dot` variable to be created. During the processing of the data step, when SAS reaches the last observation of each `HHID`, the value of `last.dot` becomes 1. The `if` statement tells SAS to write out the current observation only when the value of `last.dot` is true, i.e., when `last.dot` equals 1.

```
data oldest;
  set person_subset;
```

```

if last.HHID;
by HHID;
run;

```

Partial listing of oldest.sasdat:

HHID	AGE
1	57
2	24
3	37

We see that the new dataset oldest.sasdat has one observation per HHID and that the value of AGE is the age of the oldest person in the household.

MERGING TWO DATASETS

The data sets oldest.sasdat and household.sasdat can now be combined using one-to-one merging. Before merging is possible, both data sets need to be sorted by HHID.

```

proc sort data=household;
by HHID;
run;
proc sort data=oldest;
by HHID;
run;

```

The data step merge is then performed as the final step in the creation of our household level variable AGE.

```

data new_household;
merge household (in=a) oldest (in=b);
by HHID;
if a;
run;

```

The variable AGE can be renamed in household.sasdat so that it will not be confused with the AGE variable in person.sasdat. The following code will rename AGE.

```

data new_household;
set new_household (rename=(AGE=OLDEST_AGE));
run;

```

CONCLUSION

As was just demonstrated, it is possible to navigate between datasets that are on different levels. Properly using subsetting, sorting, and merging allows us to transfer data from one level to another. In our examples, during the transfer of AGE from the person level to the household level, not all values of AGE could be kept. It is important to check that the desired information is retained during the subsetting of a dataset. SAS only provides the tools necessary to navigate between levels. It is up to the user though to know how those tools should be used. Bon voyage!

CONTACT INFORMATION

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

Chris Becker
Abt Associates Inc.
640 N. LaSalle, Suite 400
Chicago, IL 60657
Work Phone: (312) 867-4084
Fax: (312) 867-4200
Email: chris_becker@abtassoc.com