5 minutes – Four Ways of Merging PATDATA with other Datasets

presented by David Franklin, January 11, 2006

Merging variables from one dataset into another is one of the basic data manipulation tasks that a SAS programmer has to do. The most common way to merge on data is using the MERGE statement in the DATA step but there are three other ways that can help. First though, some data:

```
Dataset: PATDATA
SUBJECT  TRT_CODE
124263      A
124264      A
124266      B

Dataset: ADVERSE
SUBJECT  EVENT
124263   HEADACHE
124266   FEVER
124266   NAUSEA
```

Each of the examples below will merge the TRT_CODE from the PATDATA dataset onto the ADVERSE dataset. Using the DATA step the code for merging would typically be:

```
DATA alldata0;
  MERGE adverse (in=a)
        patdata (in=b);
  BY subject;
  IF a;
RUN;
```

Another way would be to use SQL, so the code would look something like:

```
PROC SQL;
  CREATE TABLE alldata0 AS
    SELECT a.*, b.trt_code
    FROM adverse a
           LEFT JOIN
         patdata b
    ON a.subject=b.subject;
  QUIT;
RUN;
```

A third way is using the KEY= option in the SET statement as shown in the following example:

```
DATA alldata0;
  SET adverse;
  SET patdata KEY=subject /UNIQUE;
  DO;
    IF _IORC_ THEN DO;
      _ERROR_=0;
      trt_code='';
    END;
  END;
RUN;
```

Quick Tip

To set all missing numeric values within a dataset to zero, the following code fragment is useful:

```
drop __i;
array nv{*} _numeric_;
do __i=1 to dim(nv);
  if missing(nv{__i}) then nv{__i}=0;
end;
```

Before the third example is run the dataset PATDATA must have an index created inside it, using either the INDEX statement inside a DATASETS or SQL procedure, or INDEX option inside a DATA step.

The forth example creates a format from the data and sets the treatment from the created format:

```
DATA fmt;
  RETAIN fmtname 'TRT_FMT' type 'C';
  SET patdata;
  RENAME subject=start trt_code=label;
RUN;
PROC FORMAT CNTLIN=fmt;
RUN;
DATA alldata0;
  SET adverse;
  trt_code=put(subject,$trt_fmt.);
RUN;
```

There are other ways of merging data but the above examples show the most common. No one method is better than the other - it depends on the size of the data being merged. The following table gives a guide on the ratio of time spent when compared with a number of observations in the ADVERSE dataset (PATDATA has 1000 patients/records), using the MERGE statement as a relative standard measure of 1:

Number of Observations	MERGE Statement	PROC SQL	KEY= Option	PROC FORMAT
IK	1.00	2.19	0.25	1.25
1M	1.00	1.05	0.59	0.27
5M	1.00	0.51	0.38	0.33

Again, the table above is only a guide.