

Matching Data Using SAS® Programming and Human Judgment

Imelda C. Go, Richland County School District One, Columbia, SC

ABSTRACT

Matching data is not that difficult especially when reliable data is involved and when a computer does the matching. However, various problems arise when the variables used to match the data (e.g., social security numbers and names) are inaccurate. Even if computer-based algorithms are used to match most records, more records are sometimes found to match when a person reviews the data or uses human judgment. This paper describes the last stage of a matching process that requires human judgment. The unmatched data from two data sets are interleaved in an electronic data file that is inspected and revised by a person. The information in this revised file is used to find additional matching records that were matched based on human judgment.

EXAMPLE

For the purpose of providing a simple example, assume that there are two data sets (called FIRST and SECOND). They both contain the same three variables (SSN, LNAME, and FNAME) and their other variables do not need to be renamed if both data sets are merged by the following three variables:

Description	Variable Name
social security number	SSN
last name	LNAME
first name	FNAME

Variables SSN, LNAME, and FNAME uniquely identify the records in both data sets. That is, no two records from the same data set have the same values on these variables. This example only involves one-to-one matching between two data sets.

Suppose that data sets FIRST and SECOND already exist and their observations are numbered/indexed sequentially by variables FIRSTID and SECONDID respectively. That is, FIRSTID=i for the ith record in data set FIRST. The variable SECONDID for data set SECOND is similarly defined. Both data sets are also sorted by the variables (SSN, LNAME, and FNAME) used for merging records.

```
data first;
  set first;
  firstid=_n_;

proc sort data=first;
  by ssn lname fname;

data second;
  set first;
  secondid=_n_;

proc sort data=second;
  by ssn lname fname;
```

Merge the two data sets. Data set MATCHED contains all records that matched while the data set NOMATCH contains all other records.

```
data matched nomatch;
  merge first (in=infirst) second (in=insecond);
  by ssn lname fname;
  if infirst and insecond then output matched;
  else output nomatch;
```

None of the records in data set NOMATCH matched with each other. Hence, none of them have the same value for all three variables SSN, LNAME, and FNAME.

In reality, there might be records from data sets FIRST and SECOND that should have matched but did not. Reasons for this could include errors in data collection and data entry, and inconsistent use and spelling of names. Except for gross name changes, sorting the unmatched records by LNAME and FNAME should help identify records from both data sets that should really match with each other. This sorting results in interleaved or alternating unmatched records from both data sets.

```
proc sort data=nomatch;
  by lname fname ssn;
```

Take the unmatched records from data sets FIRST and SECOND and write them out to an external file named manual.txt.

```
data _null_;
  set nomatch;
  file 'a:\manual.txt';
  put @1 firstid @10 secondid @20 ssn @30 fname
    @50 lname;
```

Records that have a non-missing FIRSTID value are from data set FIRST and records that have a non-missing SECONDID value are from data set SECOND. The interleaved data may look like the following.

23	.	247889046	Alston	Edith
.	13	247889047	Alston	Edith
.	35	243899056	Milligan	Ofelia
33	.	243899056	Milligan	Ofeli
47	.	261789995	Santa Cruz	Roger

The 1st line is the 23rd record (FIRSTID=23) from data set FIRST. The 2nd line is the 13th record (SECONDID=13) from data set SECOND. The names for these two records are the same, but the social security numbers are off by one digit. Human judgment indicates these two records should probably match.

The 3rd and 4th lines of the file have the same social security numbers and last name. The first names are off by one letter. Human judgment indicates these two records should also probably match.

The file can be revised based on those observations:

23	13	247889046	Alston	Edith
.	13	247889047	Alston	Edith
.	35	243899056	Milligan	Ofelia
33	35	243899056	Milligan	Ofeli
47	.	261789995	Santa Cruz	Roger

The number '13' is typed where SECONDID was missing on the 1st line. The number '35' is typed where SECONDID was missing on the 4th line. The missing SECONDID value is replaced with a number corresponding to a record (from data set FIRST) that is believed to match with the record from data set SECOND.

```
data manual;
  infile 'a:\manual.txt';
  input @1 firstid @10 secondid;
  if n(firstid,secondid)=2;
```

```
proc sort data=manual; by firstid;
proc sort data=first; by firstid;

data match2;
  merge manual (in=inmanual) first(in=infirst);
  by firstid;
  if inmanual and infirst;
```

```
proc sort data=match2; by secondid;
proc sort data=second; by secondid;
data human;
    merge match2 (in=inmatch2)
           second (rename=(ssn=ssn2 lname=lname2
                           fname=fname2)in=insecond);
    by secondid;
    if inmatch2 and insecond;
    match='by hand';
```

SELECTED VARIABLES FOR DATA SET HUMAN											
FIRST ID	SECOND ID	NAME	AGE	SEX	STATUS	RELATIONSHIP	EDUCATION	RELIGION	ETHNICITY	BY	HAND
23	13	2478899046	Alton	Edith	2478899047	Alton	Edith	by	hand		
33	35	243899056	Millian	Ofelia	243899056	Millian	Ofelia	by	hand		

The HMF can be revised directly while viewing its contents on the computer screen. Revising the HMF only on the computer, or only on its paper copy first is a matter of preference. Both methods may have their own advantages.

As the HMF is being created, a special character can be used (e.g., *) to identify those 15 records. Use computer search commands to locate the special characters in the file. In the SAS program editor (version 6.12), the FIND and subsequently the RFIND (repeat find) commands are useful in locating these records. (It is useful to assign the RFIND command to a function key so the RFIND command can be issued by pressing a single key.) An example follows:

Had the data from the two data sets not been interleaved, two lists could be used to manually determine matches. The following examples are based on the previous examples:

```
Unmatched records from data set SECOND:
13      247889047 Alston      Edith
35      243899056 Milligan    Ofelia
```

Human judgment can be incorporated into the matching process. Using human judgment can identify matches that are otherwise not identified by computer programming.

Tel.: (803) 733-6079
Fax: (803) 929-3873
icqo@juno.com

