Paper SY-02

Where's The Match?

Jennifer L. Waller, Verna C. Brantley, and Robert H. Podolsky
Medical College of Georgia, Department of Biostatistics, Augusta, GA

ABSTRACT

Statisticians, while not data managers, are often asked to perform data management tasks. One task is to match a study group to a comparison group one-to-one on several variables. While easy for small datasets and matching only categorical variables, matching thousands of study subjects to thousands of comparison subjects where data come from large administrative databases and the matching variables can be categorical and continuous is difficult. How do you match continuous variables with the comparison subject being within some range of the study subject (e.g. age within +/- 2 years)? The utilization of SAS™ / Macros is one solution. The created macro uses a study subject data set, a control data set, and outputs a data set containing the one-to-one matched subjects. The macro reads in a study subject, scans through the comparison subject data set until a match is found, writes the match to the matched data set, deletes the comparison subject from the comparison data set, and proceeds to the next study subject. If no match is found the macro proceeds to the next study subject. The resulting matched data set contains those study subjects and comparison subjects that were matched one-to-one on the necessary variables.

INTRODUCTION

Statisticians are not data managers, nor do we claim to be. However, in many instances the lines between being a statistician and being a data manager become blurred in the eyes of other investigators. Many investigators believe that because statisticians analyze data, they know how to manage, organize, manipulate, create, and find data. To some extent statisticians can perform simple data management tasks (merging data sets, creating new variables, sub-setting data sets, etc.), but generally statisticians prefer to rely on those individuals who were trained to do data management for more complex tasks (matching patients and controls, identifying records in large administrative data bases who meet specific criteria for inclusion, doing data integrity checks and data cleansing, etc).

Statisticians are trained differently than data managers and statisticians and data managers recognize this difference. Statisticians also recognize what a good data manager can offer a research project in terms of data integrity, organization, and quality. But the fact still remains that investigators believe statisticians can do it all, and sometimes, due to available resources, they have to find a way to access the data the investigator needs to use.

One particular data management task that is repeatedly requested by investigators is to match one-to-one a set of patients in one data set to a set of controls in another. While

easy for small datasets and matching only categorical variables, matching thousands of study subjects to thousands of comparison subjects where data come from large administrative databases and the matching variables can be categorical and continuous is difficult.

The One-To-One Matching Programming Problem – Personal Experience

In the recent past, an investigator asked that a data set containing patients with a particular disease be matched one-to-one with three other control data sets on several factors. "No problem," I said, "Piece of cake." And then they said that a couple of the matching variables were continuous and they wanted the controls to be matched within +/- some given range (e.g. age+/- 2 years). "No can do," I said. The level of SAS™ programming needed to do something like this was beyond what I knew. But the matching was necessary and so I had to try and find a solution. My first thought was to send the data back to the source and have them perform the match, but time was of the essence and the source wasn't going to be able to provide the match in a timely fashion. A second thought for a solution was to categorize the continuous variables and then by sorting the data sets on the categorical variables merge them together to get the matches. However, that solution wasn't attractive to the investigator due to the nature of the disease of interest. The next thought was to ask the data manager in my unit if she had done one-to-one matching with continuous variables previously. While she hadn't previously performed one-to-one matching with continuous variables, the problem was interesting and had several applications to many research projects in our unit. So she and another colleague developed a solution by utilizing the SAS™/Macro facility.

The Solution to the One-To-One Matching Programming Problem

Following is the SAS code used to match one-to-one a case data set to a control data set on several variables of interest. The variables can be either categorical (e.g. sex, race, marital status) or continuous (e.g. age, screening score). In each data set, the case data set and the control data set, an identifier and the matching variables are included. Other variables can be in the data sets, but for a little more efficiency, normally extraneous variables are left out. In this example there are three matching variables. Matchvar1 is categorical and matchvar2 and matchvar3 are continuous. The specific matching criteria set by the investigator is that a control must match a case exactly on matchvar1, be within +/- 2 units on matchvar2, and be within +/- 1 unit on matchvar3. The macro does the following:

1) Reads in the case data set (dataf) and determines the number of cases (nobs) in the case data set that need to be matched.
2) Reads in the control data set (datal).
3) For the each case, the macro determines the number of controls available in the control data set (conobs) to match. In the first pass through the control data set, all observations are available. In subsequent passes through the control data set, this number changes.

4) A matching indicator (match) is set to zero, the variables in the control data set are renamed, and the matching criteria are coded.
5) For the first observation in the case data set, the first observation in the control data set is assessed as a potential match.
6) If a match is found, the matching indicator is set to 1, the matched case and control are appended to the matched data set (outdata), and the matched control is then deleted from the control data set. The macro then proceeds to the next case.
7) If the first control observation does not match the case, the macro proceeds to the next control observation.
8) If all controls are assessed for potential matches and none of the controls match the case, the program sets the matching indicator to 1 to exit out of the do loop and proceeds to the next case.
9) After all cases have been examined, the macro prints the matched data set. This can be suppressed.

After the macro performs the matching, the matched data set is checked to make sure no duplicate controls exist in the matched data set.

SAS Macro for One-To-One Matching

```
ods html close;
*****************************************************************************
**   Creation of the case and control data sets to match one-to-one. **
*****************************************************************************;
libname in 'path to SAS data sets to match';

data case;
  set in.casedataset;

proc sort data=case nodupkey; by studyid;

data control;
  set in.controldataset;

proc sort data=control nodupkey; by studyid;

proc sort data=case; by id matchvar1 matchvar2 matchvar3;
proc sort data=control; by id matchvar1 matchvar2 matchvar3;
run;

*****************************************************************************
** Begin the Macro                                                         **
** dataf = the case data set                                               **
** datal = the control data set                                            **
** outdata = the matched data set to output                                **
*****************************************************************************;
%macro matches(dataf=,datal=,outdata=);

/* Determine if the output data set exists.  If it does exist.  Delete it
from the library. */
%if %sysfunc(exist(&outdata)) %then %do;
```

```
proc datasets;
  delete &outdata;
Run;
%end;

 /* Determine the number of observations in the case data set. */
 data _null_;
  dsid=open("&dataf",'i');
  nobs=attrn(dsid,'nobs');
  call symput('nobs',nobs);
  Run;

 data datalast;
  set &datal;
  Run;

 %put nobs = &nobs;

 %do n=1 %to &nobs;
  /* Determine the number of observations in the control data set. */
  data _null_;
   dsid=open("datalast",'i');
   nobs=attrn(dsid,'nobs');
   call symput('conobs',nobs);
   Run;

   %let mmatch=1;
   data matches1;
    choose=&n;
    set &dataf point=choose;
    output;
    stop;
    Run;
   /* The match indicator variable is set to zero, the variables in the
      control data set are renamed, and the matching criteria is
      coded. */
   data matches1;
    set matches1;
    i=1;
    match=0;
    do until(match=1);
     /* Set the control data set and rename the variables*/
     set datalast(rename=(id=idb
                          matchvar1=matchvar1b
                          matchvar2=matchvar2b
                          matchvar3=matchvar3b)
                          ) point=i;
     /* If the control matches the case, set match=1 to allow for this
        matched control to be appended to the match data set and
        deleted from the control data set. */
     if matchvar1=matchvar1b and
       ((matchvar2-2) le matchvar2b le (matchvar2+2)) and
       ((matchvar3-1) le matchvar3b le (matchvar3+1))
     then do;
       match=1;
       call symput("matchn",i);
```

```
        output;
      end;
      /* If the control does not match the case and all controls have
         been examined as potential matches, set match=1 to exit the do loop. */
      if match=0 and i=&conobs then do;
        call symput("mmatch",match);
        match=1;
      end;
     /* If the control does not match the case and all controls have not
        been examined, set match=0 and go to the next control. */
     if match=0 then i=i+1;
    end;
    stop;
    Run;
   /* Appends the matches to the matched data set and deleted the matched
      control from the control data set. */
   %if &mmatch=1 %then %do;
      proc append base=&outdata data=matches1;
      Run;

      data datalast;
        set datalast;
        if _n_=&matchn then delete;
        Run;
    %end;
 %end;

 proc print data=&outdata;
 title 'Matched Data Set';
 Run;
%mend;

%matches(dataf=case,datal=control,outdata=match);

**********Check for duplicate controls********************;
proc sort data=match out=adjust nodupkey; by idb;

run;
```

CONCLUSIONS

This one-to-one matching macro has many applications. The macro has been used to match prescription drug claims to office visit claims within a specified date range. It has also been used to match multiple controls to cases (e.g. matching cases 1:3 to controls) by simple calling the macro several times and using the outputted matched data set at the new “case” data set. It has been successfully used to match 700 cases to multiple controls.

While this solution to the problem of matching one-to-one is not the most elegant and efficient, the macro works and can serve as the basis for further modification. For example, the macro could be programmed to allow for more matching variables. The macro could be generalized so that calling the macro depends on the three data sets (case, control, and matched), variable names, types of variable (categorical or

continuous), and the matching criteria for the matching variables. Such programming would also ensure that the user wasn't changing code within the macro as is done currently.

Potential additions to the macro that are being developed are:

1) Streamlining the call to the macro so that the user doesn't change the code in the macro as mentioned above.
2) Expand the macro to determine a set of potential controls that satisfy the matching criteria and randomly sampling one of potential controls to match the case.

Contact Information

Jennifer L. Waller, Verna C. Brantley, and Robert H. Podolsky
Medical College of Georgia
Department of Biostatistics
AE-3031
Augusta, GA 30912-4900
(706) 721-3785
Fax: (706) 721-6294
E-mail: jwaller@mcg.edu, vbrantley@mcg.edu, rpodolsky@mcg.edu

SAS and all other SAS Institute Inc. product of service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.