

Finding matched controls for a case-control study using your own criteria: SAS Macro programming

Designed by Muriel TAFFLET
OUTCOMEREA biostatistical department
France
tafflet@outcomerea.org

The macro is designed with SAS Software: the aim is to match one or several control patients (or unexposed patients) with each case patient (or exposed patient) on potential confounding variables for the study. The macro allows you to choose as many controls as you want per case, to optimize the power of the study.

Macro design

First, the macro calculates **for each case patient** the total number of controls that can be matched according to your own criteria (the potential confounding variables) that you have identified previously. At that time, the control patients can be the same for several case patients. Then, the macro sorts the case patients on this number to identify those case patients who are hardest to match with controls.

Finally, you choose the number of controls to be matched per case (n), and the macro selects n controls matched on your criteria; in contrast to the previous step, a control can be selected for only one case.

Executing the macro

Note that the macro must be executed by a person who has sufficient experience using SAS. First, you include the macro in your computer by running the script of the SAS macro, which is located at [“Script of the macro”](#) p.4.

Important: your database must have one line per **patient** defined by a number named “obs” (this is mandatory). There is a single line per patients **patient??**. Your database may also contain a variable indicating patient status (1 for cases and 0 for controls). To increase macro execution speed, we recommend that only variables of interest be kept in the database (i.e., obs, status, and matching criteria).

Then, you create your matching criteria at [“Your Matching Criteria”](#) p.10.

Finally, and only once the previous two steps are completed, you run the macro line command located at [“Running the macro command”](#) p.11.

Example

Suppose that you have a database containing 232 patients exposed to a factor of interest and that you want to study the excess risk of mortality associated with this factor. You will match the exposed patients with unexposed patients (i.e., patients without the factor of interest) on possible confounding factors, for example the probability of death as calculated using the TRIO score (PROBTRIO) and length of exposure to the factor. You decide that you need two unexposed patients for each exposed patient.

The steps needed to produce the database with exposed patients matched to unexposed patients are as follows:

- 1) Run the script of the SAS macro: [“Script of the macro”](#).

- 2) Create your own matching criteria: for example (1) a PROBTRIO difference no greater than 5% in either direction and (2) follow-up in unexposed patients longer than time to occurrence of the event defining exposure in exposed patients.

The variable criterion1 in “Your Matching Criteria” is PROBTRIO and criterion2 is EXPOSURELENGTH.

Therefore you must write the following program (the criteria to be modified are in red and the comments in green):

```
%macro conditions ();
    data selection; set _last_;
    if &criteria1*0.95<=&criteria1.ctrl<=&criteria1*1.05 and
    &criteria2<=&criteria2.ctrl;
    /* &criteria1 and &criteria2 are the macro-variables designating
    PROBTRIO and EXPOSURELENGTH in cases, respectively ; &criteria1.ctrl and
    &criteria2.ctrl designate PROBTRIO and EXPOSURELENGTH in the controls,
    respectively (before this procedure, you must create the EXPOSURELENGTH
    variable:
    data mydata2; set mydata;
    if exposed=1 then EXPOSURELENGTH =Day of occurrence of the event defining
    exposure;
    if exposed=0 then EXPOSURELENGTH =Length of follow-up of the patient;
    run;)* /
    run;

/* You can add a supplementary procedure that sorts the last database on
the absolute PROBTRIO difference between exposed and unexposed patients */
    data selection; set _last_;
    difftrio=abs(&criteria1-&criteria1.ctrl);
    run;
    proc sort;
    by difftrio;
    run;
%mend;
```

- 3) Run the macro command (only words in red need to be modified):

To increase the speed of the matching process, we recommend that you keep only the variables of interest in your database. As an example :

```
data mydata2; set mydata2;
keep obs exposed probtrio exposurelength;
run;

%MATCHING (database=mydata2, /*mydata2 is the name of the database*/
case=exposed, /*exposed is the variable designating the exposed patient*/
nbcase=232, /*232 is the total number of exposed patients in the database*/
nbctrl=3, /*3 is the desired number of unexposed patients per exposed
patient +1: it corresponds to the size of the stratum*/
criteria1=PROBTRIO, /*PROBTRIO is the variable name of the 1st matching
criterion*/
criteria2=EXPOSURELENGTH, /*EXPOSURELENGTH is the variable name of the 2nd
matching cr
criteria4=., /*write a . if there is no 4th criterion, as in our example*/
case_unmatched=exp_unmatched, /*exp_unmatched is the name of the database
created by the macro for cases that are not selected (because of non
correspondance or there are no more controls available for matching)*/
```

selection=selection, /*selection is the name of the database created by the macro with the observation number of cases and the corresponding observation number of the matched controls.

Form of the database "selection":

Obs (obs number of cases)	Obsctrl (obs number of matched controls)
1001	1002
1001	1003
1004	1005
...	...

*/

casectrl=expnotexp); /*expnotexp is the name of the database created by the macro with one line per observation, with the variables "obs number", "number of the corresponding stratum" and "case or control".

Form of the database "expnotexp":

Obs (obs number of patients)	Exposed (1 for exposed patient and 0 for unexposed)	Numstrate (Number of the stratum)
1001	1	1
1002	0	1
1003	1	2
1004	0	2
...

This example can be run in 30 minutes, depending on the speed of your processor. The time varies with the number of cases and controls.

Script of the macro

```
/****** MAIN macro *****/

%macro MATCHING (database,sortvar,case,nbcase,nbctrl,criterial,criteria2,
                criteria3,criteria4,case_unmatched,selection,casctrl);

/** first step: creation of sort order according to difficulty of matching
***/

/* for that, the macro selects all possible controls per case, and the same
controls can be selected for several cases*/

%sortorder;

/* creation of a database containing all cases for whom no matched control
can be found */
proc sort data=selectiontot; by obs; run;
data first; set selectiontot;
by obs;
if first.obs;
run;
proc sort data=&database; by obs; run;
data &case_unmatched; merge &database (in=a where=(&case=1)) first (in=b);
by obs;
if a and not b;
run;

/* calculation of the frequency of controls per cases and frequency of
cases per controls */
proc freq data=selectiontot order=freq noprint;
where obs ne obsctrl;
tables obs / out=tab;
tables obsctrl / out=tab2;
run;

proc sort data=tab; by obs;
proc sort data=tab2; by obsctrl;
data &database.NV; merge tab (rename=(count=countcase) drop=percent) tab2
(rename=(obsctrl=obs count=countctrl) drop=percent) &database;
by obs;
run;

/* calculation of the new number of cases to match */
proc means data=tab noprint;
var obs;
output out=out n=n;
run;
data _null_; set out;
call symput('nbcase',n);
run;

/** 2nd step: selection of n controls per case ***/

%selection;

%mend;
```

```

/***** SUPPLEMENTARY macros *****/

%macro sortorder();

/* database of the cases */
data Kleft; set &database;
if &case=1;
run;

/* database of the controls to be matched to one case */
data controls; set &database (where=(&case=0));
rename obs=obsctrl &criteria1=&criteria1.ctrl;
id=1;
run;
%if &criteria2 ne . %then %do;
data controls; set controls
(rename=(&criteria2=&criteria2.ctrl));
run;
%end;
%if &criteria3 ne . %then %do;
data controls; set controls
(rename=(&criteria3=&criteria3.ctrl));
run;
%end;
%if &criteria4 ne . %then %do;
data controls; set controls
(rename=(&criteria4=&criteria4.ctrl));
run;
%end;

/* loop on each case to match */
%do i=1 %to &nbcase;

/* database of case number i to match */
data caseni; set Kleft;
if _N_=1;
id=1;
run;

/* deletion of this case number i, to obtain the case in the next
step of the loop */
data Kleft; set Kleft;
if _N_=1 then delete;
run;

/* selection of all potential controls and case number i */
data controls1case; merge caseni (drop=&case) controls;
by id;
drop id;
run;

/* selection of possible controls with the condition of the user */

%conditions;

/* set of the controls and cases selected in the current step with
those of the previous step*/
%if &i=1 %then %do;
data selectiontot; set _last_;

```

```

        run;
        %end;
    %else %do;
        data selectiontot; set selectiontot _last_;
        run;
        %end;

%end; /* end of the loop */

%mend;

%macro selection();

/* preliminary step: delete the cases and controls that cannot be matched,
and sort the new table by frequency order of cases and frequency order of
controls */

data &database.NV; set &database.NV;
if countcase=. and &case=1 then delete; /* cases that cannot be matched */
if countctrl=. and &case=0 then delete; /* controls that cannot be matched
*/
run;
proc sort data=&database.NV; by countctrl countcase &sortvar; run; /* first
sort by countctrl because it is missing values before */

title 'New cases and control patients on which the macro will work!';
title2 see table &database.NV;
proc freq;
tables &case;
run;

/* database with all controls and cases */
data Kleft; set &database.NV;
run;

/* database with only cases */
data case; set &database.NV;
if &case=1;
run;

/* loop1: loop of number of controls wanted per case */
%do j=1 %to %eval(&nbctrl-1);
%if &j>=2 %then %do;
    %let pas=%eval(&j-1);
    data selectionpasn&pas; set _last_;
    run;
    %end;

/* database of the cases */
data Kleft; set case;
run;

        /* loop2: loop on each case */
        %do i=1 %to &nbcase;

        /* database of case number i to match */
        data caseni; set Kleft;

```

```

if _N_=1;
id=1;
run;

/* deletion of this case number i, to obtain the case in the
next step of the loop */
data Kleft; set Kleft;
if _N_=1 then delete;
run;

/* database of the controls to match with one case */
data controls; set KTleft;
rename obs=obsctrl;
id=1;
run;

%if &criterial ne . %then %do;
    data controls; set controls
(rename=(&criterial1=&criterial1.ctrl));
    run;
    %end;
%if &criteria2 ne . %then %do;
    data controls; set controls
(rename=(&criteria2=&criteria2.ctrl));
    run;
    %end;
%if &criteria3 ne . %then %do;
    data controls; set controls
(rename=(&criteria3=&criteria3.ctrl));
    run;
    %end;
%if &criteria4 ne . %then %do;
    data controls; set controls
(rename=(&criteria4=&criteria4.ctrl));
    run;
    %end;

data controls1case; merge caseni (drop=&case) controls;
by id;
drop id;
if &case=1 and obs ne obsctrl then delete; /* keep only the
case number i */
run;

user /*
/* selection of possible controls with the condition of the
%conditions;

/* selection of the first control among all possible controls:
this is why sorting is important*/

proc sort; by &case;
data last_selection; set _last_;
if _N_=1;
run;

/* deletion of controls selected in the current step, to make
sure that each control is selected only once */
proc sort data=last_selection ; by obsctrl;
proc sort data=KTleft; by obs; run;

```

```

        data KTleft; merge last_selection (in=a keep=obsctrl
rename=(obsctrl=obs)) KTleft (in=b);
        if b and not a;
        by obs;
        run;

        proc sort data=KTleft;
        by countctrl countcase &sortvar;
        run;

        /* set of the controls and cases selected in the current step
with those of the previous step*/
        %if &i=1 %then %do;
            data selectiontot2; set last_selection;
            run;
            %end;
        %else %do;
            data selectiontot2; set selectiontot2 last_selection;
            run;
            %end;
        %end; /* end of loop2 */

%if &j>=2 %then %do;
    data selectiontot2; set selectiontot2 selectionpasn&pas;
    run;
    %end;
%end; /* end of loop1 */

title 'Number of cases unselected by the macro';
title2 see table &case_unmatched;

%if &nbctrl=2 %then %do;
    data &case_unmatched; set &case_unmatched selectiontot2
(where=(obs=obsctrl));
    run;
    %end;
%else %if &nbctrl>2 %then %do;
    proc sort data=selectiontot2; by obs;
    data temp; merge first (in=a) selectiontot2 (where=(obs=obsctrl)
in=b);
    by obs;
    if a and not b;
    run;
    data &case_unmatched; set &case_unmatched temp (where=(obs=obsctrl));
    by obs;
    run;
    %end;

proc freq;
tables &case;
run;

title Database of match cases and controls: &Selection;
title2 'Database with one line per patient, with the number of the
stratum';
title3 and with status of the patient: &casctrl;
data &selection; set selectiontot2;
where obsctrl ne obs;
run;
proc print noobs N='Number of matched cases: ';
var obs obsctrl;

```

```
run;
data temporaire; set &selection (keep=obs obsctrl);
retain suj id_strate 0;
if suj ne obs then do;
suj=obs;
id_strate=id_strate+1;
end;
run;

proc sort data=temporaire; by obs;
data case; set temporaire (keep=obs id_strate);
by obs;
if first.obs;
&case=1;
run;
data control; set temporaire (keep=obsctrl id_strate rename=(obsctrl=obs));
&case=0;
run;
data &casectl; set case control;
run;
proc sort data=&casectl; by id_strate; run;
```

Your Matching Criteria

/* these matching criteria are provided as an example
you can choose 1 to 4 matching criteria and assign conditions to these
criteria */

```
%macro conditions ();  
    data selection; set _last_;  
    /* you can change the following selection criteria as you want*/  
    if (&criteria1*0.95<=&criteria1.ctrl<=&criteria1*1.05) and  
        ((&criteria2*0.8<=&criteria2.ctrl<=&criteria2*1.2)  
            or (&criteria2=1 and &criteria2.ctrl in (1 2))  
            or (&criteria2=2 and &criteria2.ctrl in (1 2 3))  
            or (&criteria2=3 and &criteria2.ctrl in (2 3 4))  
            or (&criteria2=4 and &criteria2.ctrl in (3 4 5)));  
    run;  
%mend;
```

Running the macro command

Simply modify the information in red and run the line of command.

```
%MATCHING (  
database=mydata, /*select the name of your database*/  
sortvar=obs, /*select the sorting variable for your database*/  
case=status, /*select the name of the status variable of the patient (case  
or exposed) ; 1 for cases and 0 for controls*/  
nbcase=200, /*select the total number of case (or exposed) patients of your  
database*/  
nbctrl=5, /*select the desired number of controls per case patients +1:  
this corresponds to the size of the stratum (ex: if you want a case control  
1:1 write the number 2)*/  
criterial=sapsII, /*select the 1st matching criterion*/  
criteria2=lengthofstay, /*select the 2nd matching criterion or . if there is  
no other criterion*/  
criteria3=ventilation, /*select the 3rd matching criterion or . if there is  
no other criterion  
  
other criterion */  
case_unmatched=case_unmatched, /*select the name of the database created by  
the macro for cases that are not selected (because of non corresponding or  
no more controls available for matching)*/  
selection=selection, /*select the name of the database created by the macro  
with the observation number of cases and the corresponding observation  
number of matched controls */  
casectrl=casectl); /*select the name of the database created by the macro  
with one line per observation, with variables "observation number of the  
patient", "status of the patient (case or control)", "number of the obtain  
stratum"*/
```