/\*Where XX is written, remember to input a number that applies to your need. \*/

%LET agerange = XX; /\*The agerange is how much of a difference you will allow in age for your cases and eligible controls. Remember to set agerange according to how age is reported. If age is reported as years, then agerange is as well. \*/

%LET ratio = XX; /\*insert the number of controls you want for each case\*/

/\* ‘population’ is your data set containing both cases and possible controls. You must create a variable called ‘casecontrol’ which is marked ‘1’ for cases. A variable called ‘uniqueid’ is needed to distinguish the observations/participants from each other; it must be unique for everyone. Besides these two variables, you will need to have the matching variables. In this form of the code, we have ‘gender‘ and ‘age’.\*/

DATA cases controls;

SET population;

IF casecontrol = 1 THEN OUTPUT cases;

ELSE OUTPUT controls;

RUN;

PROC FREQ NOPRINT DATA=cases;

TABLES age\*gender/OUT=caseout; /\*The controls are matched on age and gender, which are also the names of the variables \*/

RUN;

%MACRO sample(v\_age, v\_gender, v\_count); /\*The following section is a program within the code, here called ‘sample’. The program ends with the %MEND-statement\*/

DATA qualify1;

SET controls;

WHERE (&v\_age-&agerange <= age <= &v\_age+&agerange)

AND

(gender = "&v\_gender");

case\_age = &v\_age;

case\_gender = "&v\_gender";

SEED = RANUNI(0);

PROC SORT;

BY SEED;

DATA qualify2;

SET qualify1 NOBS=totobs;

IF \_N\_ <= &v\_count\*&ratio;

IF &v\_count\*&ratio <= totobs THEN tag = 'yes';

ELSE tag = 'no';

PROC APPEND BASE=matches DATA=qualify2 force; /\*new data set ‘matches’ will contain the matched controls\*/

PROC SORT DATA=qualify2 OUT=temp1 (KEEP=uniqueid);

BY uniqueid;

PROC SORT DATA=controls OUT=temp2;

BY uniqueid;

DATA controls; /\* controls already matched are removed and can not be matched again \*/

MERGE temp1(IN=in1) temp2(IN=in2);

BY uniqueid;

IF in2 AND NOT in1;

%MEND sample;

DATA \_NULL\_; /\*This data step calls the macro.\*/

SET caseout;

CALL EXECUTE ('%sample('||age||','||gender||','||count||')');

RUN;

/\*if you need to run the code anew, due to errors for instance, you may use this following step to remember to delete the following datasets. Otherwise, DO NOT RUN THIS DATA STEP, since it will delete your matches\*/

proc datasets nolist library=work;

delete temp1 temp2 qualify1 qualify2 matches controls;

run;

/\* The next part is for testing if any of the cases have not received the required number of controls \*/

PROC FREQ NOPRINT DATA=matches;

TABLES case\_age\*case\_gender/OUT=con\_out;

PROC SORT DATA = caseout(RENAME=

(age=case\_age gender=case\_gender count=case\_cnt));

BY case\_age case\_gender;

PROC SORT DATA = con\_out (RENAME= (count=con\_cnt));

BY case\_age case\_gender;

DATA final (DROP=percent);

MERGE caseout con\_out;

BY case\_age case\_gender;

con\_need = case\_cnt\*&ratio;

IF con\_cnt = . THEN con\_cnt = 0;

diff = con\_cnt-con\_need;

PROC PRINT DATA = final; /\*creates a table showing characteristics of the cases who have not received enough matches and how many matches they are missing\*/

WHERE diff < 0;

TITLE 'Insufficient Matches';

RUN;