DOCUMENTS DE TRAVAIL CEMOI / CEMOI WORKING PAPERS

A SAS macro to estimate Average Treatment Effects with Matching Estimators

Nicolas Moreau[1]

http://cemoi.univ-reunion.fr/publications/

Centre d'Economie et de Management de l'Océan Indien
Université de La Réunion

October 2016

Abstract

This paper presents a SAS macro to estimate the Average Treatment Effect (ATE) and the Average Treatment Effect for the Treated (ATET) with nearest-neighbor matching.

JEL : C210

[1] E-mail: nicolas.moreau@univ-reunion.fr

Introduction

In this paper we present the SAS macro *nn_matching*. *nn_matching* estimates nearest-neighbor matching with replacement for the Average Treatment Effect (ATE) and the Average Treatment Effect for the Treated (ATET). We draw heavily on Abadie et al. (2004) and Abadie and Imbens (2002, 2011). We refer the reader to these articles for a clear and comprehensive presentation of the matching estimator.
Following these authors, *nn_matching* supplies the simple matching estimator and the bias-corrected matching estimator. Variance estimation is done assuming either homoscedasticity of the conditional variance or allowing for heteroscedasticity of the conditional variance. The source code is included in Appendix to this paper and is also available from the author.

Syntax of *nn_matching*

The syntax is %nn_matching(data=,y=,w=,x=,M=,scaling=,covarbias=);.

Data specifies the data set. *y* is the outcome variable. *w* is the binary variable treatment indicator. *x* specifies the list of covariates to be used in the matching.
M specifies the number of matches to be made per observation. It could be any integer between 1 and the minimum of the number of treated and controls in the sample.
If there are ties, if different matched pairs (*i*,*j*) and (*i*,*l*) lead to the same distance $d_{ij}= d_{il}$, the number of matches per unit is greater than *M*.
Scaling specifies the metric for measuring the distance between two vector of covariates. Following Abadie et al. (2004), the default is the diagonal matrix of the inverses of the sample variances of each covariate in *x* when scaling is not specified by the user. If *scaling*=1, the Mahalabonis metric is used. It is the inverse of the sample covariance matrix of the covariates. If *scaling*=2, the identity matrix is used instead. *Covarbias* specifies the list of covariates to be used to compute the bias-corrected matching estimator. The default list is *x* when *covarbias* is not specified by the user.

Note that all variables in *y*, *x*, *w* and *covarbias* must be numeric.

Results presentation and output data files

The average treatment effect and the average treatment effect for the treated are automatically computed. For each estimated parameter, estimated standard errors assuming homoscedasticity of the conditional variance and allowing for heteroscedasticity of the conditional variance are supplied.
The bias-corrected matching estimator developed in Abadie and Imbens (2002, 2011) is automatically computed if the number of "continuous" covariates in *x* is greater than one. In *nn_matching*, all variables that are not binary are considered as continuous.
The first two output tables summarize the model specification and the estimation options. The third one gives summary statistics, such as the number of treated units, the number of controls matched to treated units and so on. The last table shows the main results. The column "Estimate" reports the estimated average treatment effects. The next column shows the corresponding robust standard errors. "z" are the z-statistics to test whether average treatment effects is 0. They are computed as the estimated parameters divided by their standard errors. The column "P-value" reports the p-values for the z-statistics for a two-sided test. The last two columns exhibit the lower and upper bounds of the 95% confidence interval for the z-statistics.

To assess the validity of the balancing hypothesis, *nn_matching* provides normalized differences (see Abadie and Imbens, 2011; Austin, 2009) so as box-plots and empirical distribution functions to

compare the covariate distributions between treatment groups before and after matching. Theses plots are edited for continuous variables only. For binary variables, contingency tables are provided.

Two temporary output data files are created. They need to be stored in a specific folder with a libname statement to become permanent.
Outdata1 includes an internal identification number for observation *i* created by the program and based on the original sort order and called _id_. *nbelements* specifies the number of matches for unit *i*. *count* is the number of times unit *i* is used as a match. *km_i* specifies the number of times unit *i* is used as a matched for any observation *j* of the opposite treatment group weighted by the total number of matches for the given observation *j*. *Outdata1* also includes the outcome variable *y*, the covariates *x* and the treatment group indicator *w*.
Outdata2 includes the list of indices for the *M* closest matches for unit *i*. *_id_* is the internal identification number for observation *i* and *idM* the corresponding identification number of *i*'s closest matches in the opposite treatment group. For each *_id_*, there is one row per match. For instance, if unit 3 is matched with units 5, 6 and 10, there are three rows in outdata2 that correspond to *_id_*=3; the first one with *idM*=5, the second one with *idM*=6 and the last one with *idM*=10. For each matched pair (*i*,*j*), the distance (in absolute value) between unit *i* and unit *j* of the opposite treatment group is stored, so as unit *j*'s outcome value.

An example

Following Abadie et al. (2004), we use the particular data set constructed by Dehejia and Wahba (1999) from Lalonde (1986) to examine the effect of participation in a job-training program on individuals' earnings in 1978.

re78: individual earnings in 1978
treat = 1 if the individual participates to the job-training program, 0 otherwise
educ: years of education
black=1 if Afro-American, 0 otherwise
hisp=1 if Hispanic, 0 otherwise
married=1 if married, 0 otherwise
re74 (re75) : individual earnings in 1974 (1975)
u74 (u75)=1 if unemployed in 1974 (1975), 0 otherwise.

The macro statement:

```
%nn_matching(data=lib.lalonde,y=re78,w=treat,x=age educ black hisp married re74 re75 u74 u75,M=4,scaling=,covarbias=);
```

replicate Abadie et al. (2004)'s results apart from the standard error of ATET with bias-correction under homoscedasticity of the conditional variance which is larger in Abadie et al. (2004). The results are presented below.

ESTIMATING AVERAGE TREATEMENT EFFECTS

Model specification

Outcome variable:	re78
Binary treatment:	Treat
Matching variables:	age educ black hisp married re74 re75 u74 u75

Estimation options

Number of matches requested:	4
Scaling matrix used:	Inverse variances
Covariates used for bias correction:	age educ black hisp married re74 re75 u74 u75

Summary statistics

Number of observations:	445
Number of control units:	260
Number of treated units:	185
Number of treated units matched to controls:	174
Number of control units matched to treated:	239
Number of times a treated unit is used as a match (MIN):	1
Number of times a treated unit is used as a match (MAX):	21
Number of times a control unit is used as a match (MIN):	1
Number of times a control unit is used as a match (MAX):	11

Estimation results assuming homoscedasticity of the conditional variance

	Estimate	Std.Error	z	P-value	L. bound 95% CI	U. bound 95% CI
Average Treatment Effect (ATE)	1903.326	720.215	2.643	0.008	491.731	3314.921
ATE with bias correction	1717.726	720.341	2.385	0.017	305.883	3129.569
Average Treatment Effect for the Treated (ATET)	1994.622	712.729	2.799	0.005	597.699	3391.544
ATET with bias correction	1838.424	712.824	2.579	0.01	441.314	3235.534

Estimation results allowing for heteroscedasticity of the conditional variance

	Estimate	Std.Error	z	P-value	L. bound 95% CI	U. bound 95% CI
Average Treatment Effect (ATE)	1903.326	745.421	2.553	0.011	442.328	3364.324
ATE with bias correction	1717.726	745.421	2.304	0.021	256.729	3178.724
Average Treatment Effect for the Treated (ATET)	1994.622	752.634	2.65	0.008	519.486	3469.757
ATET with bias correction	1838.424	752.634	2.443	0.015	363.289	3313.56

Normalized covariate mean differences are presented in the following table:

Normalized covariate mean differences between treated and controls

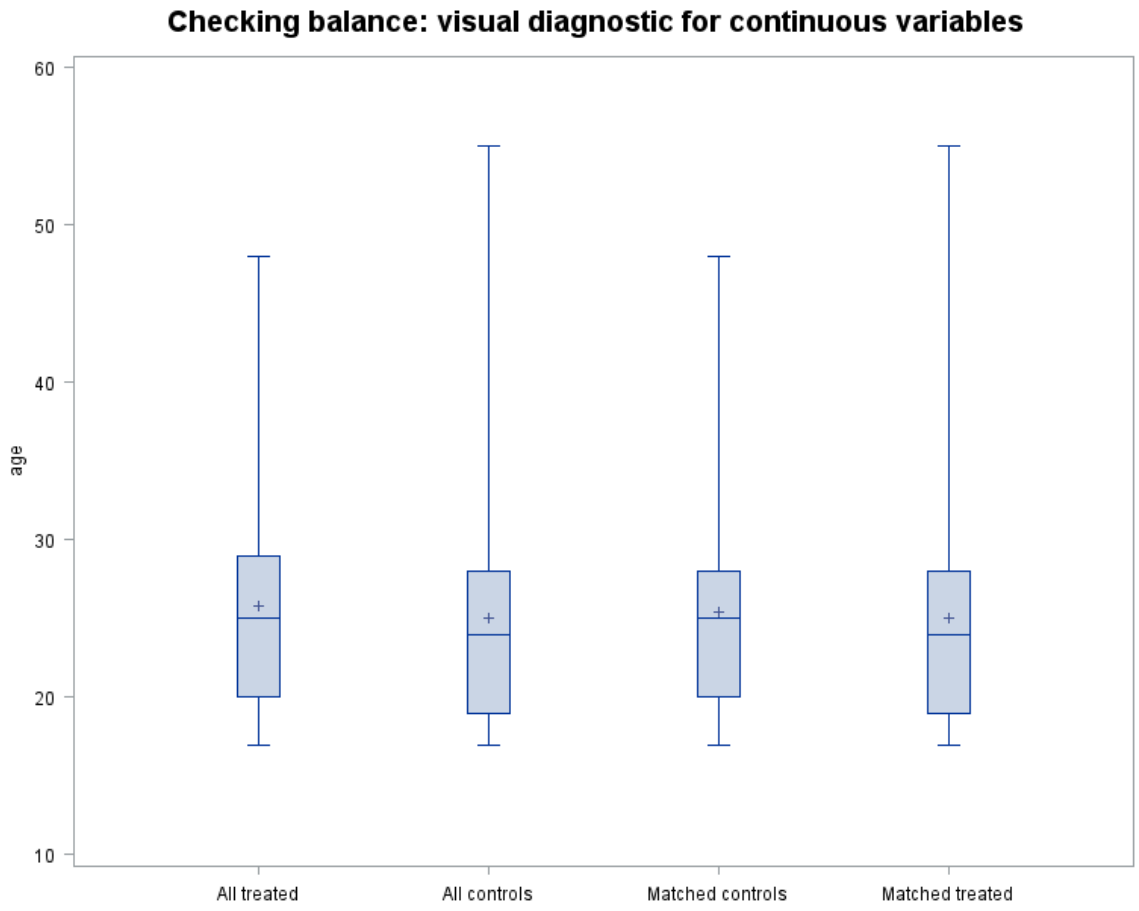
	Unmatched Sample	Matched sample (T)	Matched sample (C)
AGE	0.107	0.115	0.06
EDUC	0.141	0.097	0.105
BLACK	0.044	-0.029	0.081
HISP	-0.175	-0.079	-0.16
MARRIED	0.094	0.091	0.05
RE74	-0.002	0.057	-0.012
RE75	0.084	0.095	0.096
U74	-0.094	-0.092	-0.084
U75	-0.177	-0.154	-0.182

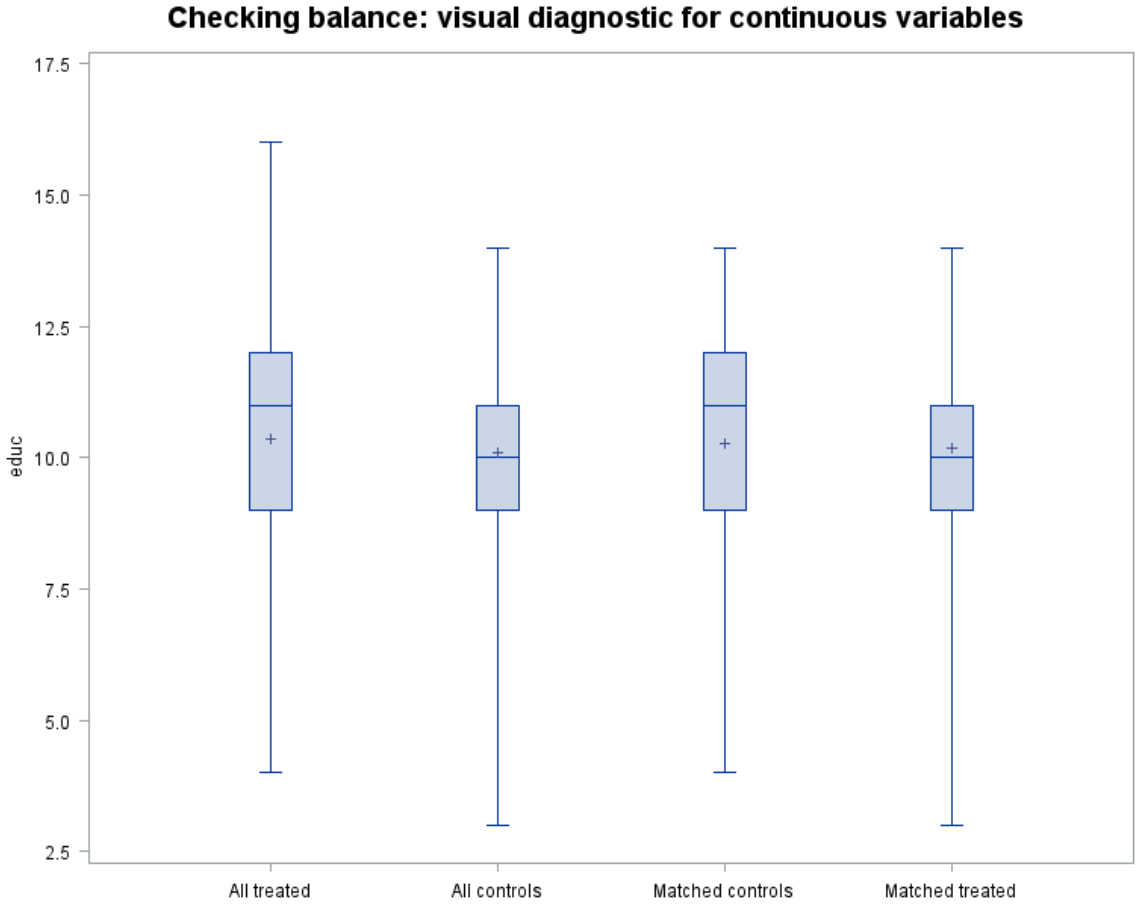
Note: 'Matched sample (T)' is for normalized mean differences between all sample treated and their matches, 'Matched sample (C)' for normalized mean differences between all sample controls and their matches.

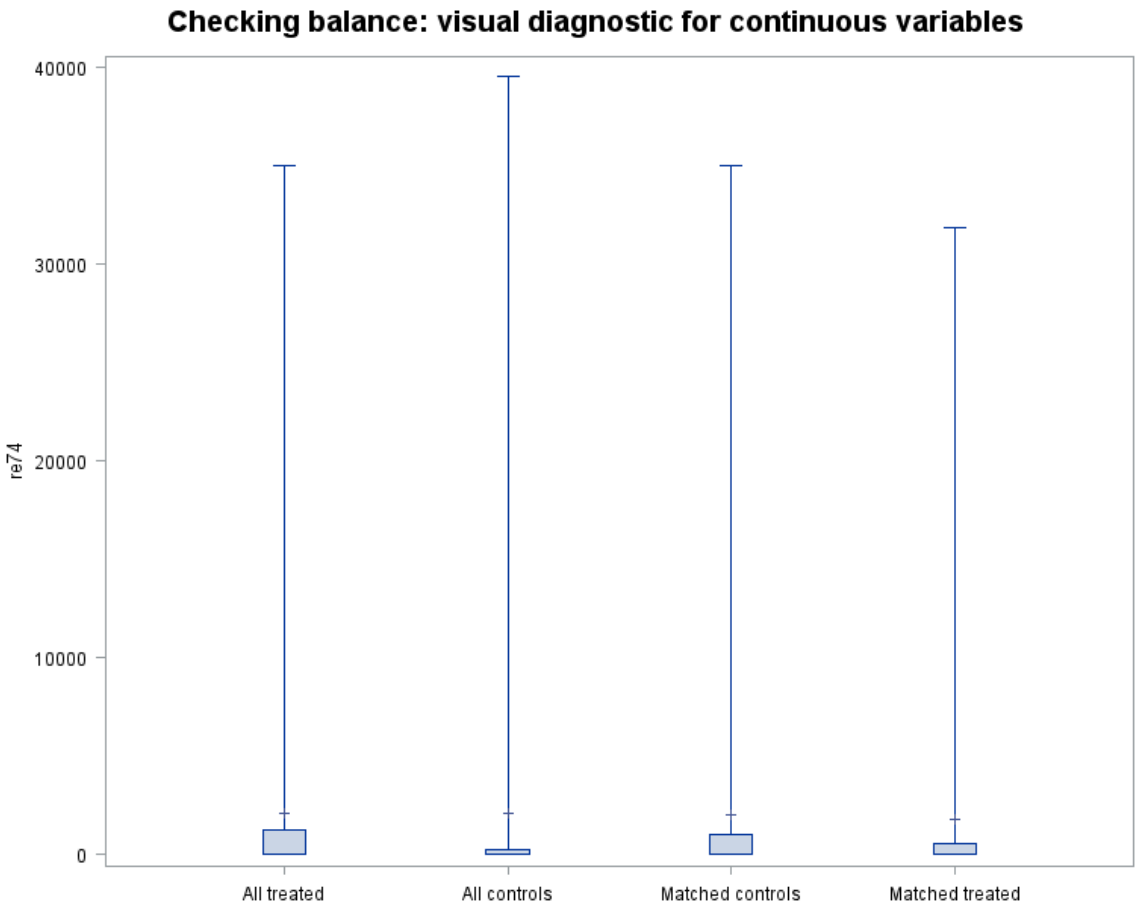
Comparing binary covariate distributions between treatment groups before and after matching

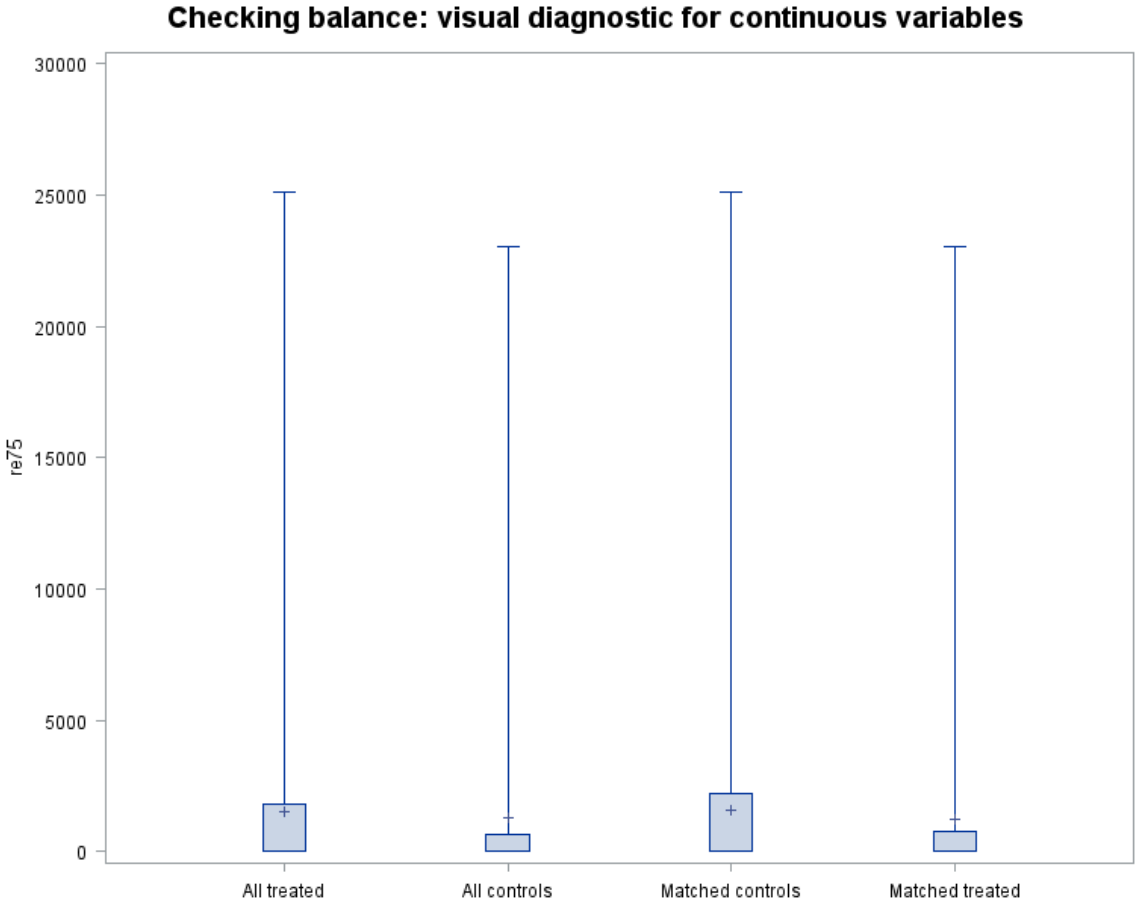
		SampleType			
		All treated	All controls	Matched controls	Matched treated
black					
0	Distribution in %	15.68	17.31	14.37	14.64
1	Distribution in %	84.32	82.69	85.63	85.36
hisp					
0	Distribution in %	94.05	89.23	93.68	92.05
1	Distribution in %	5.95	10.77	6.32	7.95
married					
0	Distribution in %	81.08	84.62	82.76	84.52
1	Distribution in %	18.92	15.38	17.24	15.48
u74					
0	Distribution in %	29.19	25.00	28.74	25.10
1	Distribution in %	70.81	75.00	71.26	74.90
u75					
0	Distribution in %	40.00	31.54	40.23	32.64
1	Distribution in %	60.00	68.46	59.77	67.36

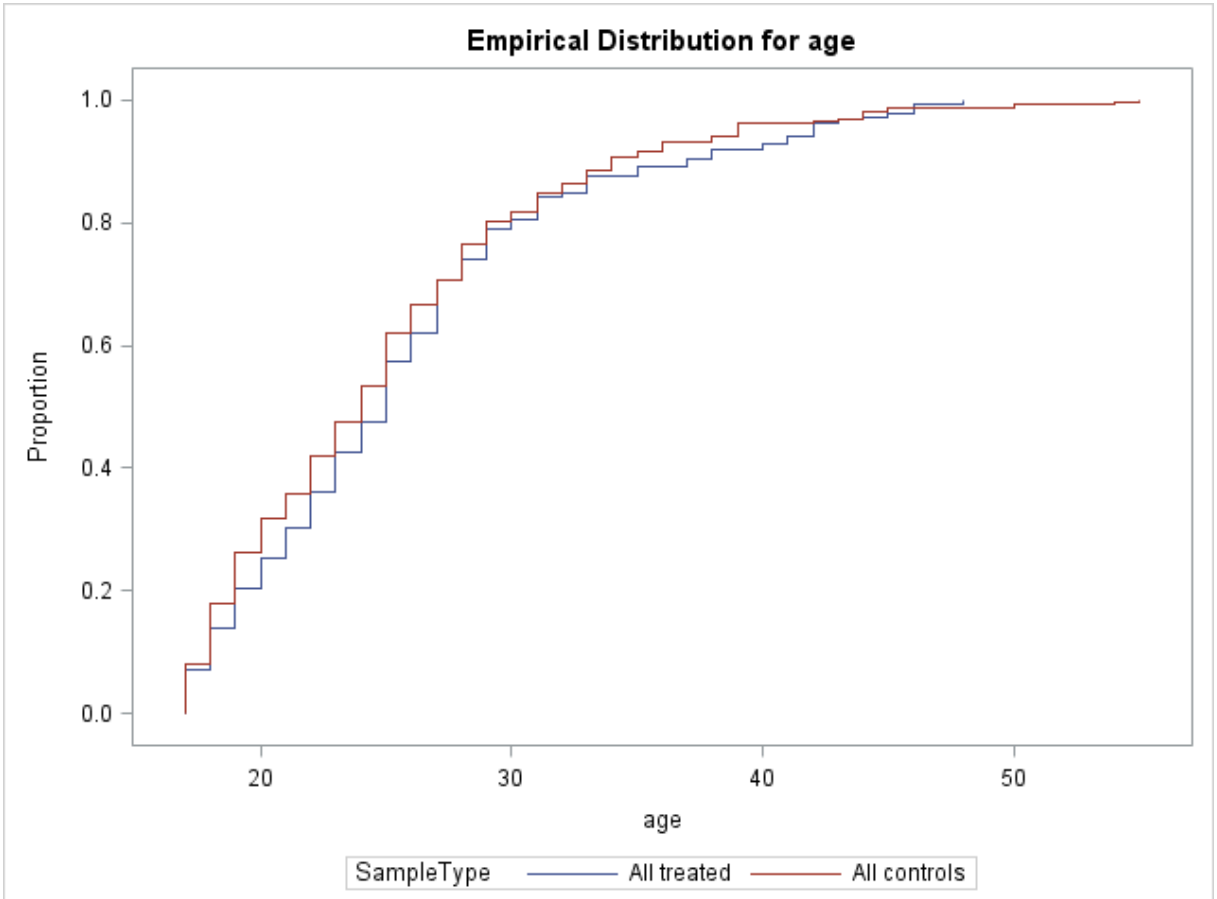
Visual diagnostic to assess the validity of the balancing hypothesis for continuous variables is then provided:

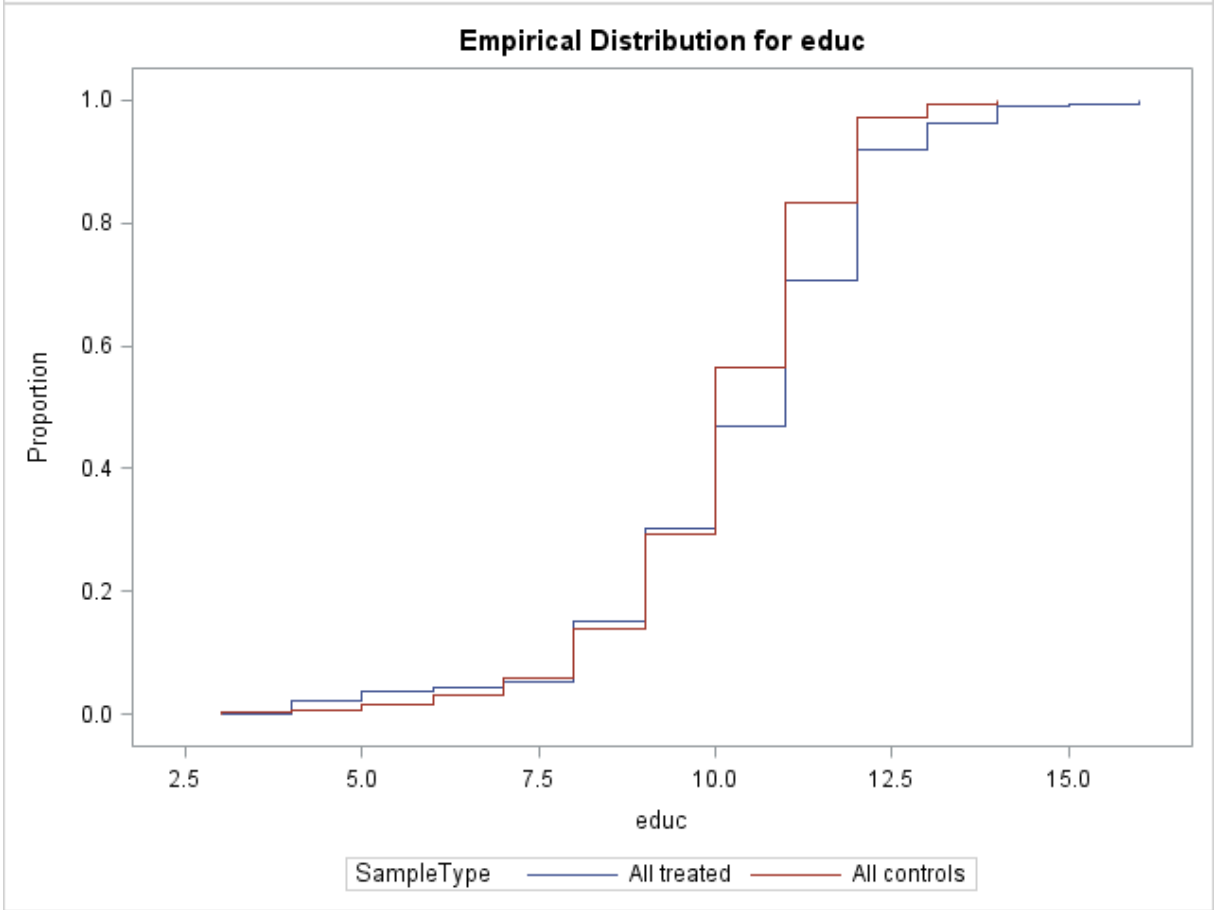

Checking balance: visual diagnostic for continuous variables
age
60
50
40
30
20
10
All treated
All controls
Matched controls
Matched treated

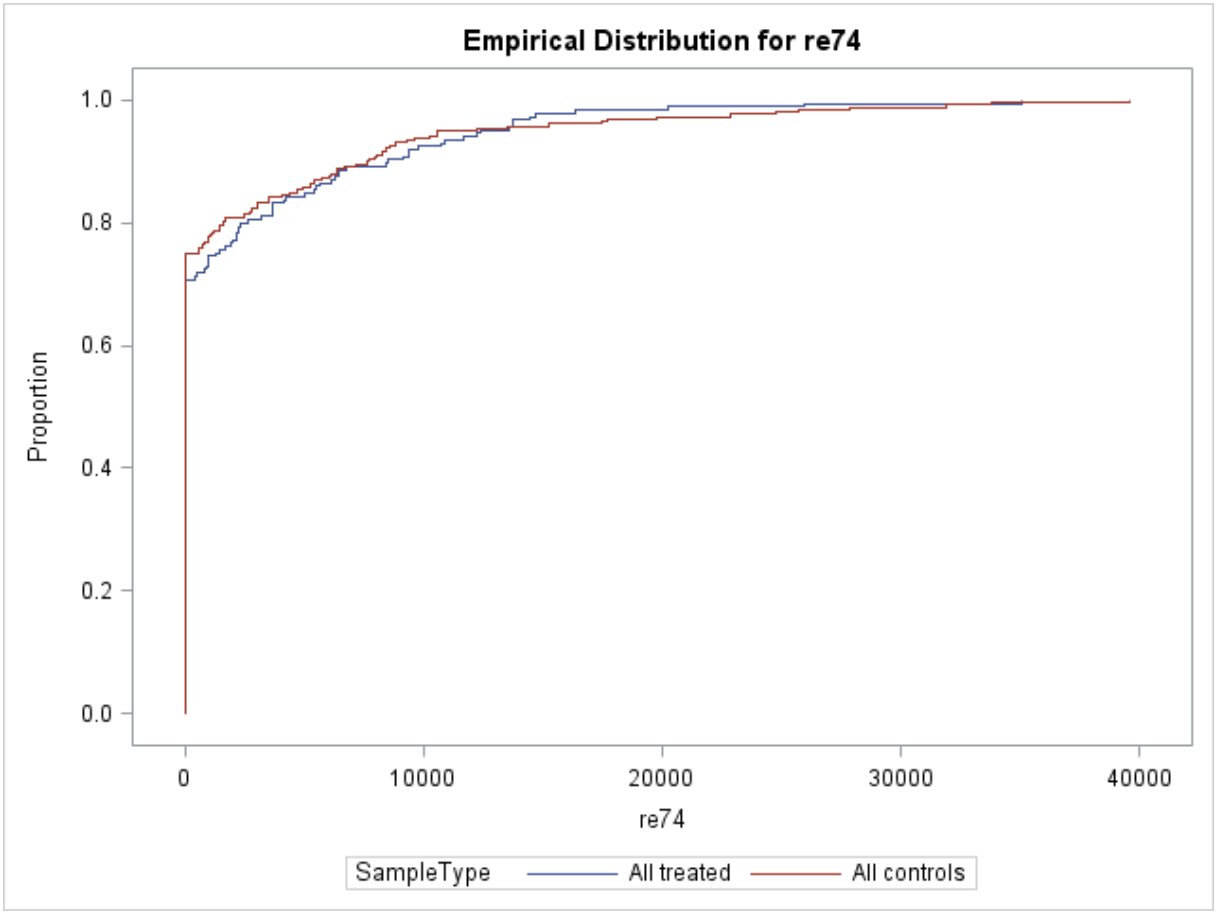

Checking balance: visual diagnostic for continuous variables
educ
17.5
15.0
12.5
10.0
7.5
5.0
2.5
All treated
All controls
Matched controls
Matched treated

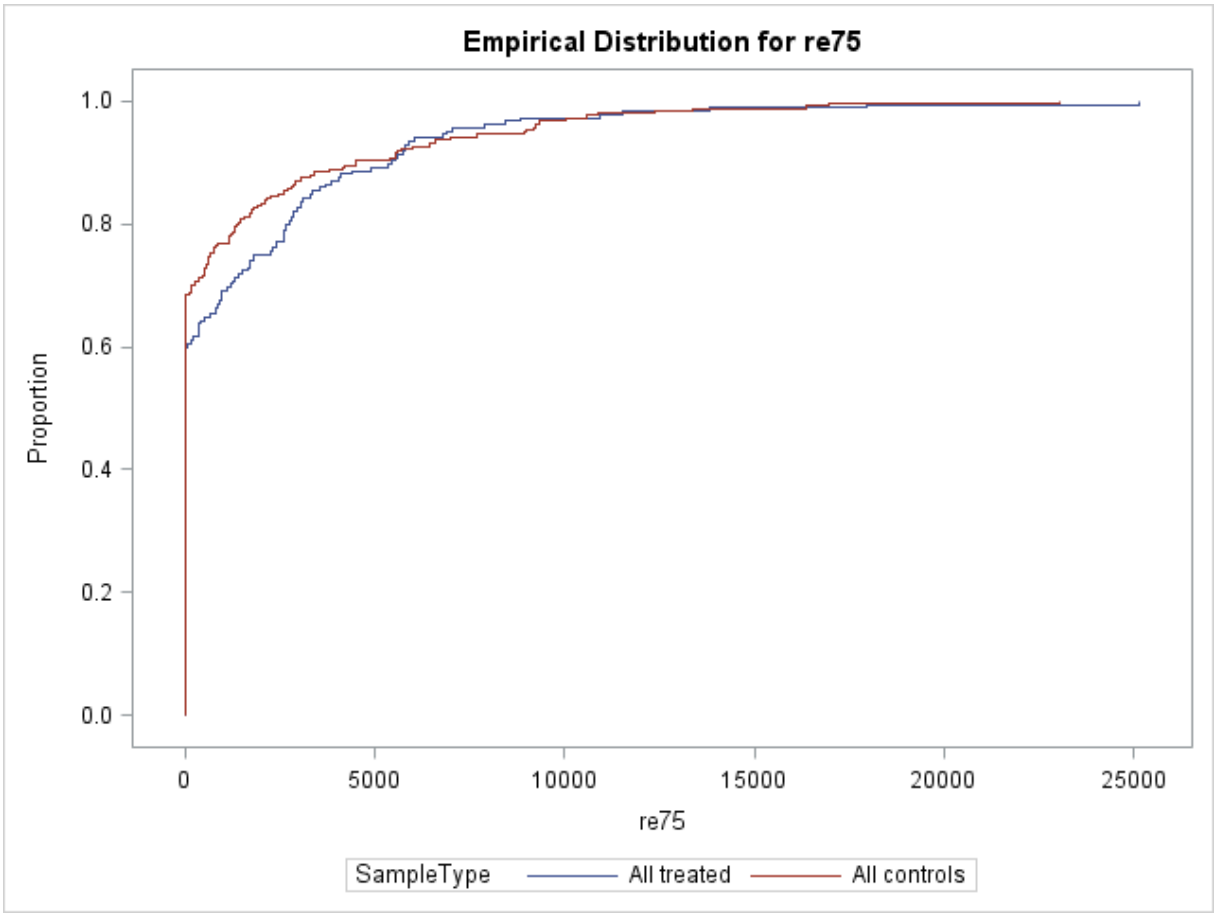

Checking balance: visual diagnostic for continuous variables
re74
40000
30000
20000
10000
0
All treated
All controls
Matched controls
Matched treated

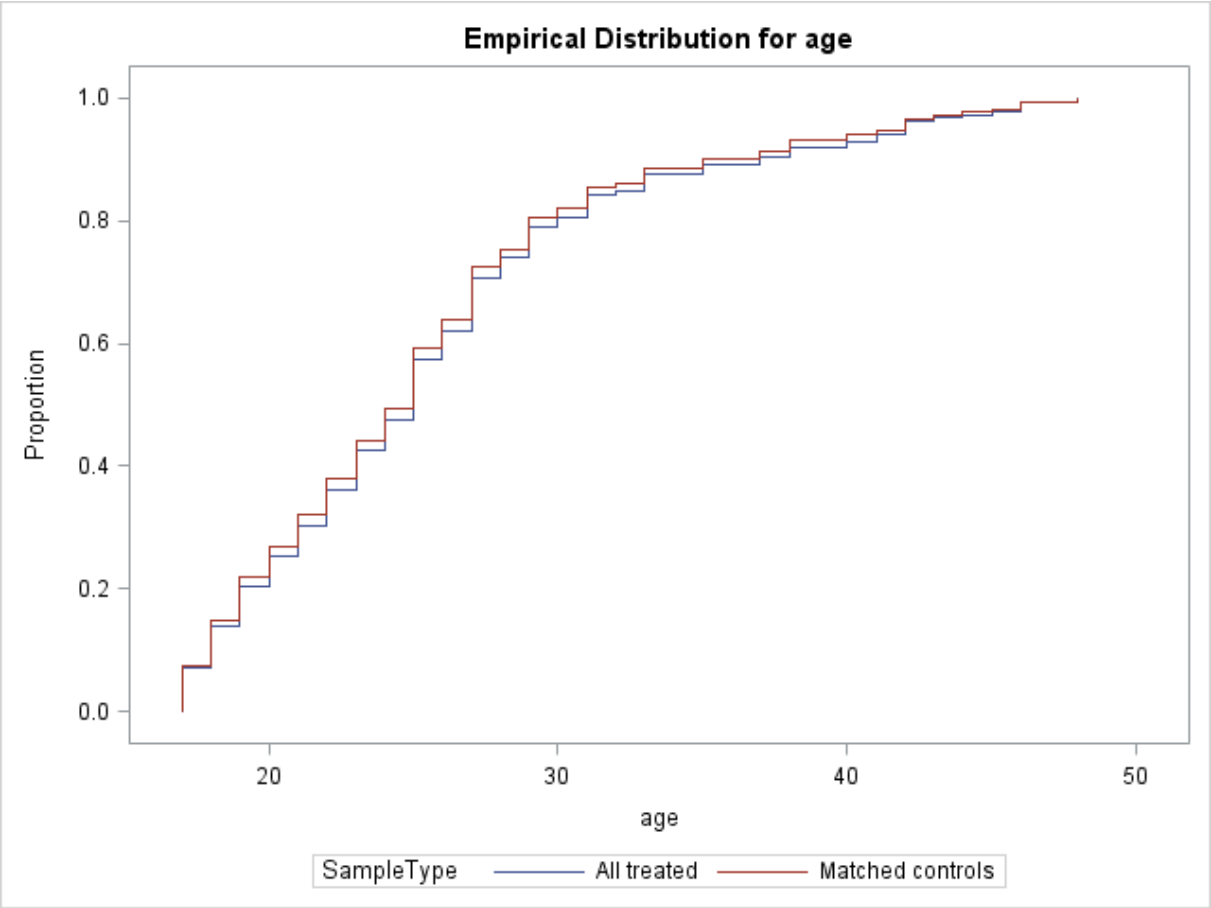

Checking balance: visual diagnostic for continuous variables
re75
30000
25000
20000
15000
10000
5000
0
All treated
All controls
Matched controls
Matched treated

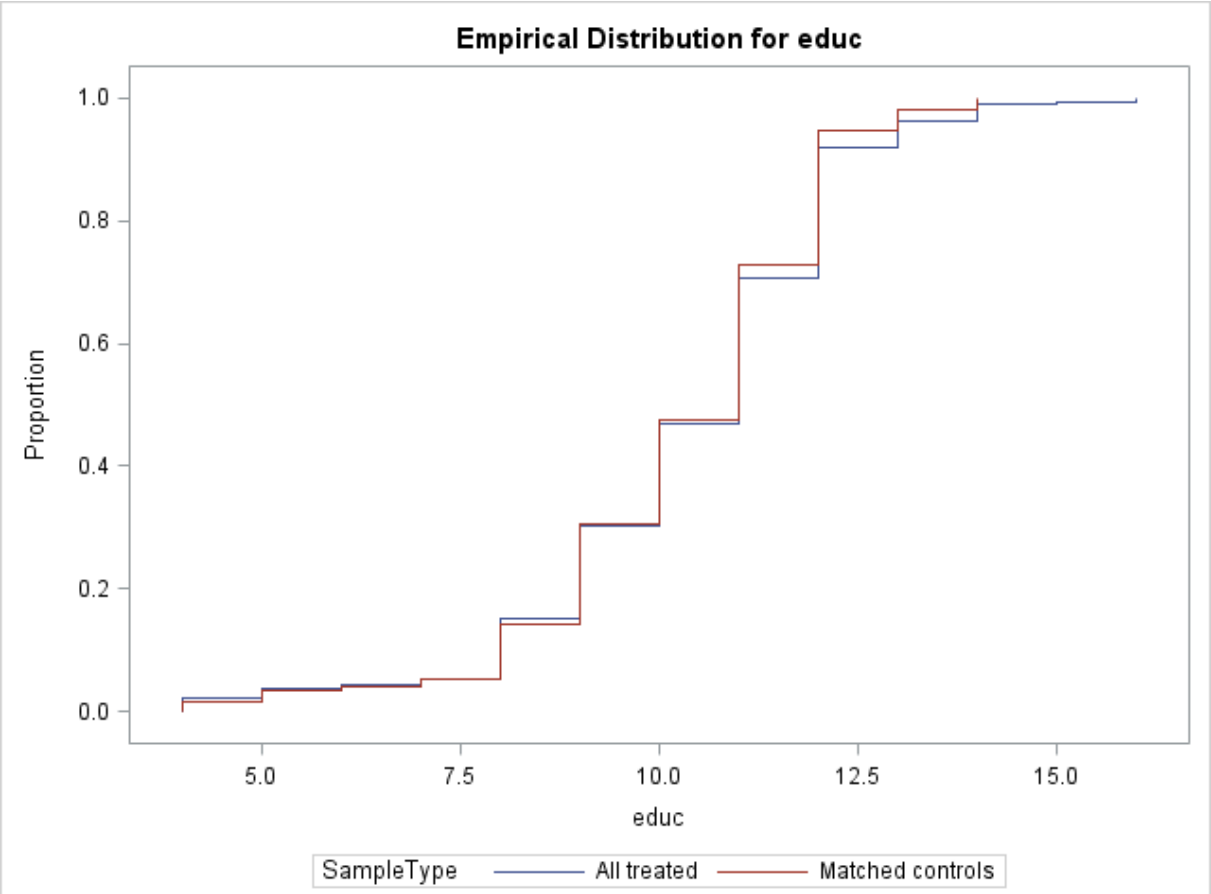

Empirical Distribution for age
Proportion
1.0
0.8
0.6
0.4
0.2
0.0
20
30
40
50
age
SampleType
All treated
All controls

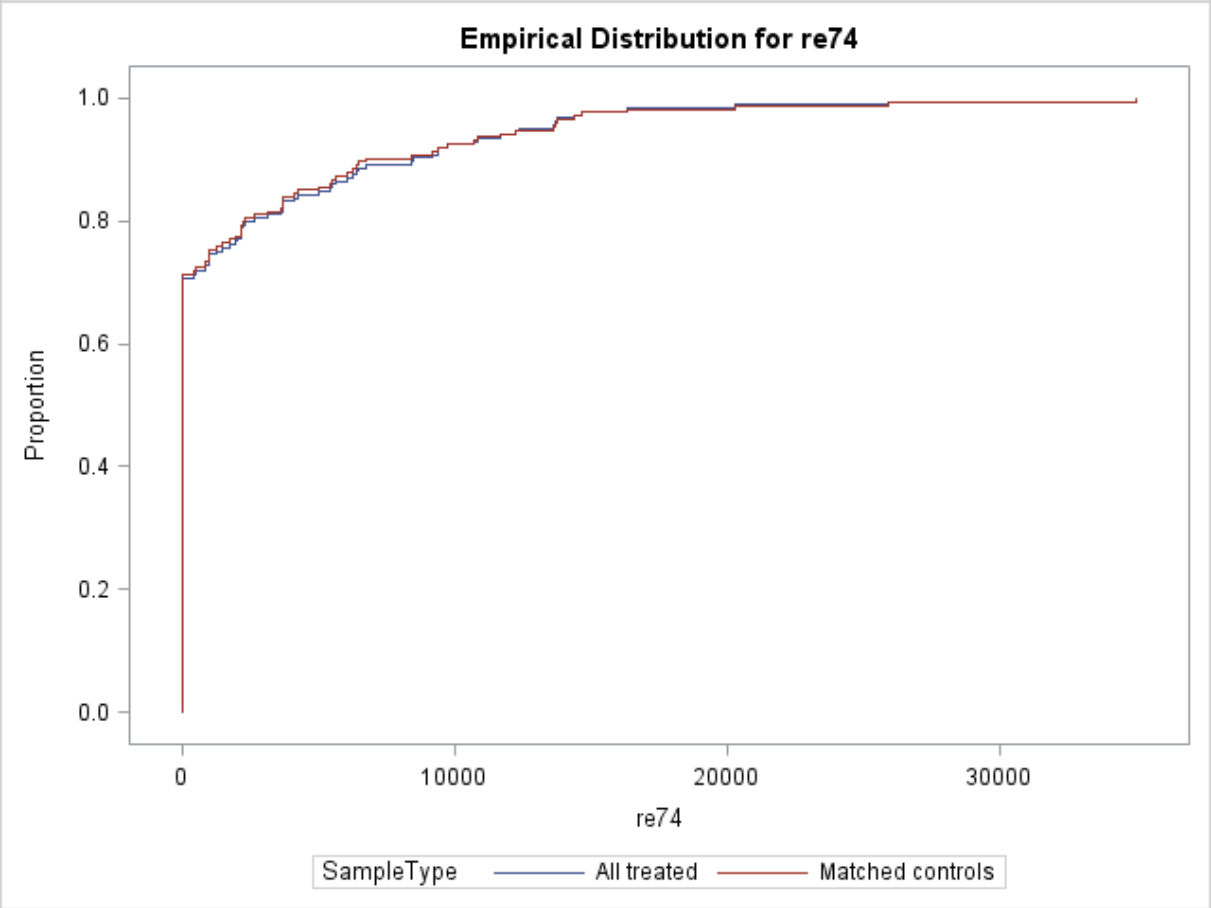

Empirical Distribution for educ
Proportion
1.0
0.8
0.6
0.4
0.2
0.0
2.5
5.0
7.5
10.0
12.5
15.0
educ
SampleType
All treated
All controls

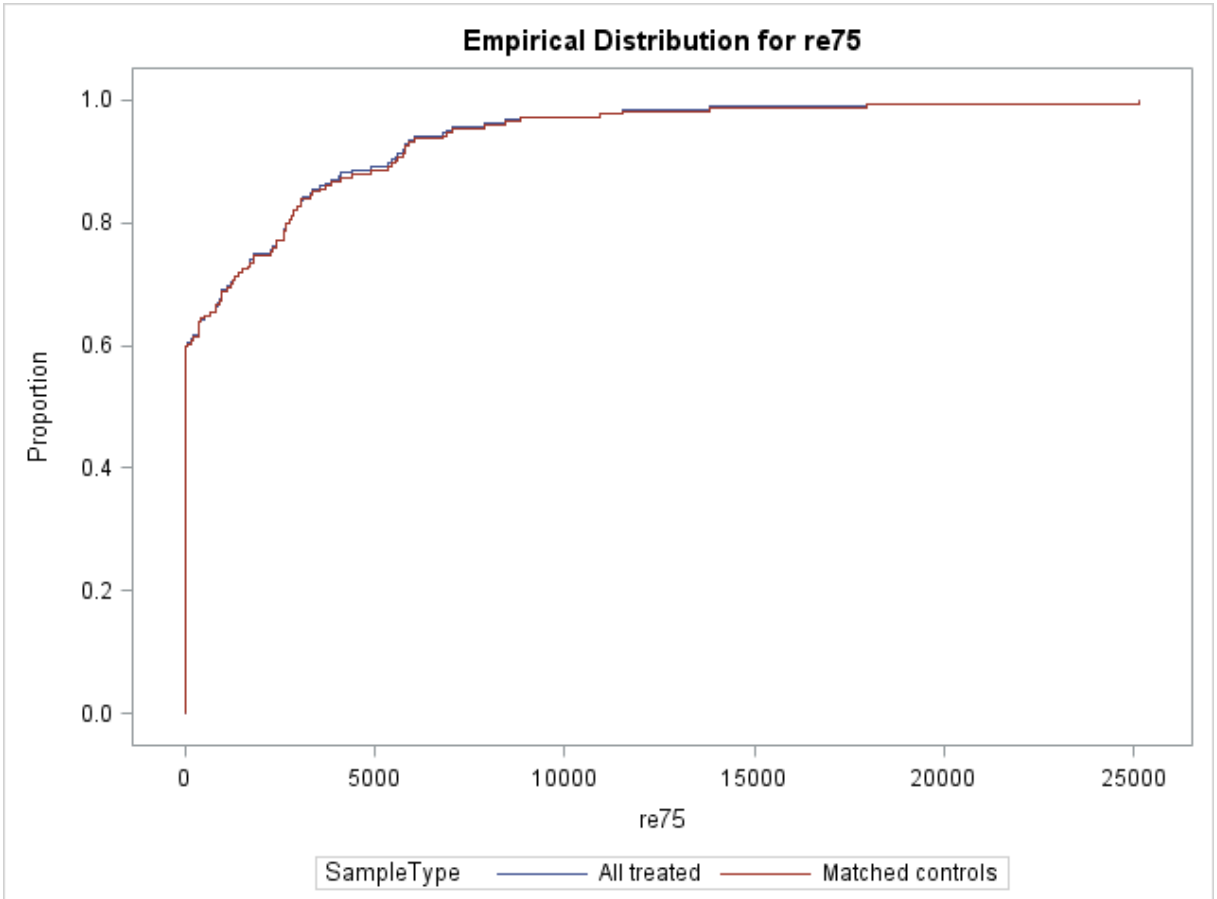

Empirical Distribution for re74
Proportion
1.0
0.8
0.6
0.4
0.2
0.0
0
10000
20000
30000
40000
re74
SampleType
All treated
All controls

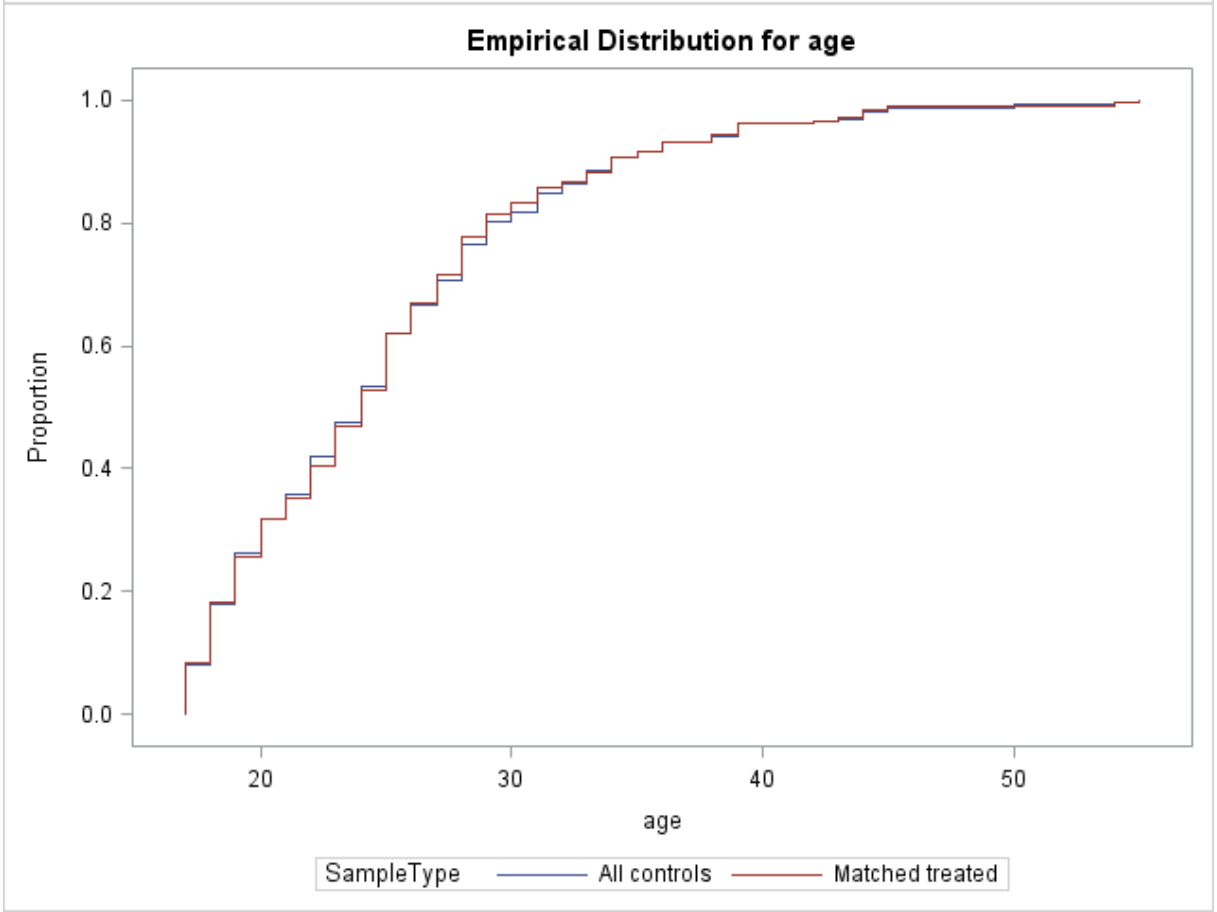

Empirical Distribution for re75
Proportion
1.0
0.8
0.6
0.4
0.2
0.0
0
5000
10000
15000
20000
25000
re75
SampleType
All treated
All controls

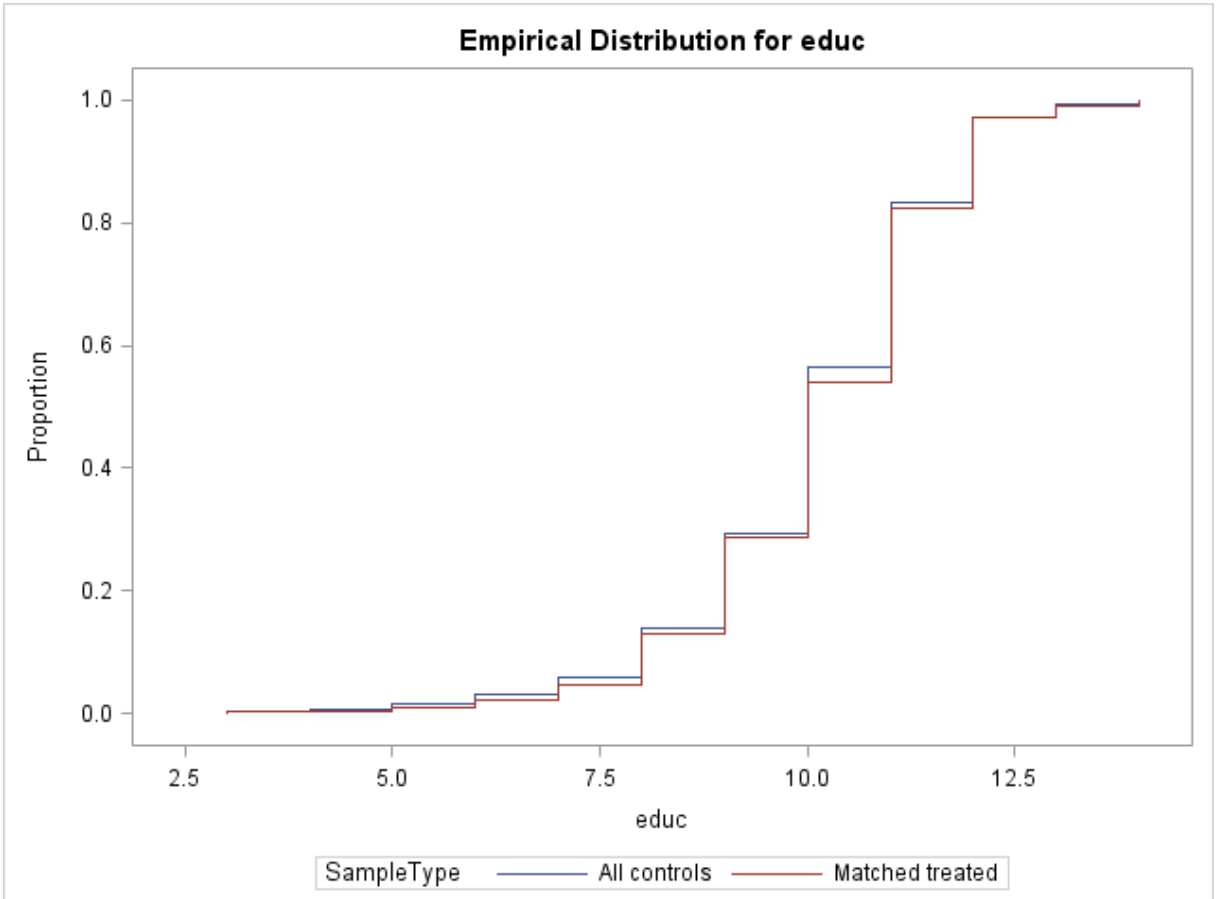

Empirical Distribution for age
Proportion
1.0
0.8
0.6
0.4
0.2
0.0
20
30
40
50
age
SampleType
All treated
Matched controls

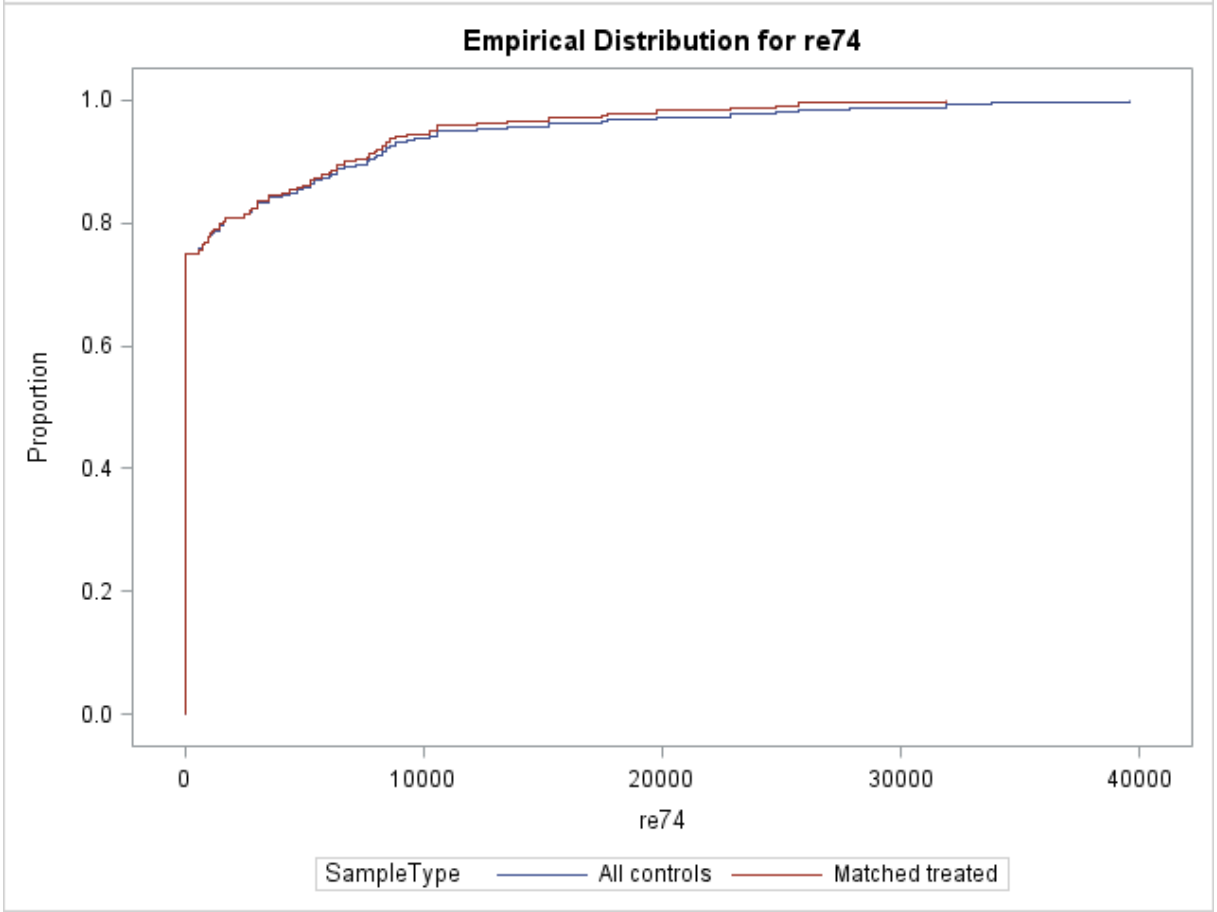

Empirical Distribution for educ
Proportion
1.0
0.8
0.6
0.4
0.2
0.0
5.0
7.5
10.0
12.5
15.0
educ
SampleType
All treated
Matched controls

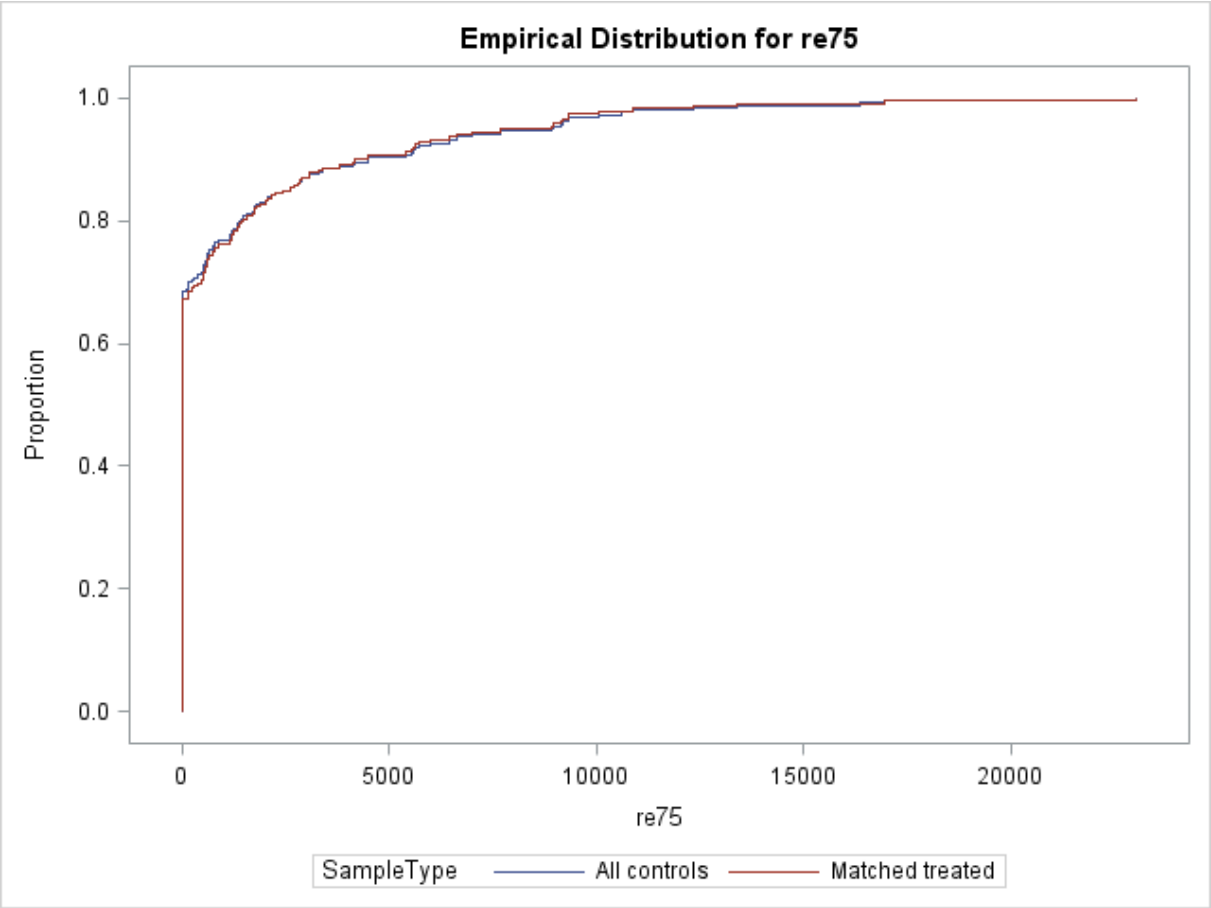

Empirical Distribution for re74
Proportion
1.0
0.8
0.6
0.4
0.2
0.0
0
10000
20000
30000
re74
SampleType
All treated
Matched controls


Empirical Distribution for re75
Proportion
1.0
0.8
0.6
0.4
0.2
0.0
0
5000
10000
15000
20000
25000
re75
SampleType
All treated
Matched controls


Empirical Distribution for age
Proportion
1.0
0.8
0.6
0.4
0.2
0.0
20
30
40
50
age
SampleType
All controls
Matched treated


Empirical Distribution for educ
Proportion
1.0
0.8
0.6
0.4
0.2
0.0
2.5
5.0
7.5
10.0
12.5
educ
SampleType
All controls
Matched treated


Empirical Distribution for re74
Proportion
1.0
0.8
0.6
0.4
0.2
0.0
0
10000
20000
30000
40000
re74
SampleType
All controls
Matched treated


Empirical Distribution for re75
Proportion
1.0
0.8
0.6
0.4
0.2
0.0
0
5000
10000
15000
20000
re75
SampleType
All controls
Matched treated

References

Abadie A., Drukker D., Leber Herr J. and Imbens G. W. (2004), "Implementing matching estimators for average treatment effects in Stata", The Stata Journal, vol. 4, n°3: 290-311.

Abadie A., and Imbens G. W. (2002), "Simple and bias-corrected matching estimators for average treatment effects", NBER Technical Working Paper 283.

Abadie A., and Imbens G. W. (2011), "Bias-Corrected Matching Estimators for Average Treatment Effects", Journal of Business and Economic Statistics, vol. 29, n°1: 1-11.

Austin P. C. (2009), "Balance diagnostics for comparing the distribution of baseline covariates between treatment group in propensity-score matched samples",
Statistics in Medicine, vol. 28: 3083–3107.

Dehehia R. H., and Wabba S. (1999), "Causal effects in non-experimental studies: Re-evaluation of the evaluation of training programs", Journal of the American Statistical Association, vol. 94: 1053-1062.

Lalonde R. J. (1986), "Evaluating the econometric evaluations of training programs", American Economic Review, vol. 74, n°4: 604-620.

Appendix

```
%macro nn_matching(data=,y=,w=,x=,M=,scaling=,covarbias=);

data table;
set &data(keep=&y &x &w);
_id_+1;
run;

data Treatment;
set table(where=(&w=1));
data Control;
set table(where=(&w=0));
run;

/* Sample covariance matrix of the covariates */

proc corr data=&data cov out=covarianceX(where=(_type_="COV")) vardef=N noprint;
var &x;
run;

/* Measuring distances between treated units and control units */

proc iml;
start distance;
use Treatment;
read all var {&x} into XTreated;
read all var {&y} into outcomeT;
read all var {_id_} into idT;
close Treatment;
use Control;
read all var {&x} into XControl;
read all var {&y} into outcomeC;
read all var {_id_} into idC;
close Control;

Ntreated=nrow(XTreated);
Ncontrol=nrow(XControl);

/** Scaling matrix for measuring the distance between the vector of covariates **/

use covarianceX;
read all var {&x} into S;
close covarianceX;

/*scaling*/

%let metric=&scaling;
%if &metric=1 %then %do;
 V=inv(S);
 Scaling={"Mahalanobis"};
%end;
%else %if &metric=2 %then %do;
 V=i(ncol(Xtreated));
 Scaling={"Euclidean"};
%end;
%else %do;
 V=inv(diag(vecdiag(S)));
 Scaling={"Inverse variances"};
%end;

create scaling from scaling;append from scaling;

/** For each treated, measuring all distances between its vector of covariates and those of controls
**/

distT=J(Ntreated#Ncontrol,5,0);
do i=1 to Ntreated;
 disti=J(Ncontrol,5,0);
  do l=1 to Ncontrol;
   disti[l,1]=sqrt( (XTreated[i,]-XControl[l,])*V*(XTreated[i,]-XControl[l,])` );
   disti[l,4]=idC[l];
   disti[l,5]=outcomeC[l];
  end;
  disti[,2]=idT[i];
  disti[,3]=1;
  distT[Ncontrol#(i-1)+1:i#Ncontrol,]=disti;
end;
```

```
varnames= {"Distance" "_id_" "W" "idM" "OutcomeM"};
create distanceT from distT[colname=varnames ];
append from distT ;

free distT;

/** For each control, measuring all distances between its vector of covariates and those of treated
**/

distC=J(Ncontrol#Ntreated,5,0);
do l=1 to Ncontrol;
 distl=J(Ntreated,5,0);
  do i=1 to Ntreated;
   distl[i,1]=sqrt( (XTreated[i,]-XControl[l,])*V*(XTreated[i,]-XControl[l,])` );
   distl[i,4]=idT[i];
   distl[i,5]=outcomeT[i];
  end;
  distl[,2]=idC[l];
  disti[,3]=0;
  distC[NTreated#(l-1)+1:l#NTreated,]=distl;
end;

varnames= {"Distance" "_id_" "W" "idM" "OutcomeM"};
create distanceC from distC[colname=varnames];
append from distC;

free distC Xtreated XControl;

finish distance;
run distance;
quit;

/** Closest matches for treated and controls **/

data distance;
set distanceT distanceC;

proc sort data=distance;
by _id_ distance;

data ClosestM1 ClosestM2;
set distance;
by _id_ distance;
retain nbelements;
if first._id_ then nbelements=0;
nbelements=nbelements+1;
if nbelements<=&M then output ClosestM1;else output ClosestM2;

data tie(drop=nbelements);
set ClosestM1(keep=_id_ distance nbelements rename=(distance=distancetie));
where nbelements=&M;

data ClosestM2(drop=distancetie);
merge ClosestM2 tie;
by _id_;
if distance=distancetie;

data ClosestM;
set ClosestM1(drop=nbelements) ClosestM2(drop=nbelements);

proc summary data=ClosestM nway;
class _id_;
output out=nbelements(keep=_id_ _freq_ rename=(_freq_=nbelements));
run;

proc sort data=ClosestM;
by _id_ distance;
run;

data ClosestM;
merge nbelements ClosestM;
by _id_;
run;

/* Measuring the number of times an observation is used as a match (km_i) for any observation j of
the opposite treatment group weighted by the total number of matches for the given observation j */

proc sort data=ClosestM;
by idM;
```

```
data km(rename=(idM=_id_));
set ClosestM(keep=idM nbelements);
by idM;
retain count km_i;
if first.idM then do;
 count=0;
 km_i=0;
 end;
count=count+1;
km_i=km_i+1/nbelements;
if last.idM then output;
run;

proc sort data=treatment;
by _id_;
proc sort data=control;
by _id_;
proc sort data=km;
by _id_;
run;

data regfile;
merge treatment(keep=_id_ &y &x &w) control(keep=_id_ &y &x &w) km;
by _id_;
if km_i=. then km_i=0;
if count=. then count=0;
run;

/*** Bias-correction : not necessary if the number of continuous covariates is less than 2 ***/

/* Counting binary and continuous covariates */

ods select nlevels;
ods table nlevels=n_levels;
proc freq data=regfile nlevels;
table &x /noprint;
title "Identifying the number of continuous covariates";
run;

title;

data n_levels;
set n levels end=end;
contvar+(nlevels>2); /* if more than 2 categories then "continuous", else binary */
binaryvar+(nlevels=2);
if end then call symputx('nbcont',contvar);
if end then call symputx('nbbinary',binaryvar);
run;

%if &nbcont>=2 %then %do;
 %if &covarbias= %then %let xbias=&x;%else %let xbias=&covarbias;

 /** Weighted least squares on matched treated observations **/

 proc reg data=regfile(where=(&w=1 and km_i>0 )) outest=parmT(keep=intercept &xbias) noprint;
 model &y=&xbias;
 weight km_i;
 run;
 quit;

 /** Weighted least squares on matched controls **/

 proc reg data=regfile(where=(&w=0 and km_i>0)) outest=parmC(keep=intercept &xbias) noprint;
 model &y=&xbias;
 weight km_i;
 run;
 quit;

 /*** OLS estimated and average potential outcomes for treated and controls ***/

 data XT(keep= id  &xbias);
 set regfile(where=(&w=1));
 data XC(keep=_id_ &xbias);
 set regfile(where=(&w=0));

 proc iml;
 start estimate;
 use parmC;
 read all var _all_ into bhatC;
 close parmC;
```

```
use parmT;
read all var _all_ into bhatT;
close parmT;
use XT;
read all var{&xbias} into XT;
read all var {_id_} into idT;
close XT;
use XC;
read all var{&xbias} into XC;
read all var {_id_} into idC;
close XC;

XC=J(nrow(XC),1,1)||XC;
XT=J(nrow(XT),1,1)||XT;

/** Bias-correction for treated **/

yhat_tc=XT*bhatC`; /* OLS estimates on controls imputed to treated */
yhat_cc=XC*bhatC`; /* OLS estimates on controls imputed to controls */

/** Bias-correction for controls **/

yhat_ct=XC*bhatT`; /* OLS estimates on treated imputed to controls */
yhat_tt=XT*bhatT`; /* OLS estimates on treated imputed to treated */

free XC XT bhatT bhatC;

varname1= {"_id_"};
varname2= {"yhat_tc"};
varname3= {"yhat_cc"};
varname4= {"yhat_ct"};
varname5= {"yhat_tt"};

create idt from idt[colname=varname1];append from idT;
create idC from idC[colname=varname1];append from idC;

create yhat_ct from yhat_ct[colname=varname4];append from yhat_ct;
create yhat_tc from yhat_tc[colname=varname2];append from yhat_tc;
create yhat_tt from yhat_tt[colname=varname5];append from yhat_tt;
create yhat_cc from yhat_cc[colname=varname3];append from yhat_cc;

free yhat_tc yhat_cc yhat_ct yhat_tt;
finish estimate;
run estimate;

data yhat_ct;set idc;set yhat_ct;
data yhat_tc;set idt;set yhat_tc;
data yhat_cc;set idc;set yhat_cc;
data yhat_tt;set idt;set yhat_tt;
run;

proc sort data=yhat_ct;by _id_;
proc sort data=yhat_tc;by _id_;
proc sort data=yhat_cc;by _id_;
proc sort data=yhat_tt;by _id_;
run;

data closestM;
merge yhat_cc(rename=(_id_=idM)) yhat_tt(rename=(_id_=idM)) closestM(in=a);
by idM;
if a;

proc sort data=closestM;
by _id_;

data closestM;
merge yhat_tc yhat_ct closestM(in=a);
by _id_;
run;

/** Estimating ATE and ATET **/

data outcome_hat(keep=_id_ nbelements outcome_hat outcome_hatbias_corr);
set ClosestM;
by  id ;
if yhat_tc=. then yhat_tc=0;
if yhat_tt=. then yhat_tt=0;
if yhat_ct=. then yhat_ct=0;
if yhat_cc=. then yhat_cc=0;
retain sum sumbias_corr;
```

```
if first._id_ then do;
 sum=0;sumbias_corr=0;
end;
bias_corr=w*(yhat_tc-yhat_cc)+(1-w)*(yhat_ct-yhat_tt);
sum=sum+outcomeM;
sumbias_corr=sumbias_corr+outcomeM+bias_corr;
if last._id_ then do;
 outcome_hat=sum/nbelements;
 outcome_hatbias_corr=sumbias_corr/nbelements;
 output;
end;
run;

data outcome_hat;
merge outcome_hat(in=x) control(keep=_id_ &y &w) treatment(keep=_id_ &y &w);
by _id_;
difoutcome=&w*(&y-outcome_hat)+(1-&w)*(outcome_hat-&y);
difoutcomebias_corr=&w*(&y-outcome_hatbias_corr)+(1-&w)*(outcome_hatbias_corr-&y);

/** Averaging the differences to compute ATE and ATET **/

proc means data=outcome_hat noprint;
var difoutcome difoutcomebias_corr;
output out=ATE(keep=ATE ATE_withoutBIAS) mean=ATE ATE_withoutBIAS;
run;

proc means data=outcome_hat(where=(&w=1)) noprint;
var difoutcome difoutcomebias_corr;
output out=ATET(keep=ATET ATET_withoutBIAS) mean=ATET ATET_withoutBIAS;
run;

/**** Variance estimation ****/

/** 1: Outcome variance estimation under homoscedasticity **/

data ate;set ate;cst=1;
data atet;set atet;cst=1;
data ClosestM;set ClosestM;cst=1;

data ClosestM;
merge ate(keep=cst ate ate_withoutbias) atet(keep=cst atet atet_withoutbias) ClosestM;
by cst;

proc sort data=ClosestM;
by _id_;

data sigma2;
merge ClosestM(in=a) treatment(keep=_id_ &y &w) control(keep=_id_ &y &w);
by _id_;
if a;
retain sumATE sumATE_withoutbias sumATET sumATET_withoutbias N0 N1;
if first. id_ then do;
 sumATE=0;
 sumATE_withoutbias=0;
 sumATET=0;
 sumATET_withoutbias=0;
 N0=0;
 N1=0;
end;
difoutcome=&w*(&y-outcomeM)+(1-&w)*(outcomeM-&y);
sumATE=sumATE+(difoutcome-ate)**2;
sumATE_withoutbias=sumATE_withoutbias+(difoutcome-ate_withoutbias)**2;
sumATET=sumATET+&w*(&y-outcomeM-ateT)**2;
sumATET_withoutbias=sumATET_withoutbias+&w*(&y-outcomeM-ateT_withoutbias)**2;
if last._id_ then do;
 weightedsumATE=sumATE/nbelements;
 weightedsumATE_wbias=sumATE_withoutbias/nbelements;
 weightedsumATET=sumATET/nbelements;
 weightedsumATET_wbias=sumATET_withoutbias/nbelements;
 N0=(&w=0);
 N1=(&w=1);
 output;
end;
run;

proc summary data=sigma2;
var weightedsumATE weightedsumATE_wbias weightedsumATET weightedsumATET_wbias N0 N1;
output out=sigma2(drop=_type_ rename=(_freq_=N)) sum=;
```

```
data sigma2;
set sigma2;
sigma2=0.5*weightedsumATE/N;
sigma2_wbias=0.5*weightedsumATE_wbias/N;

sigma2T=0.5*weightedsumATET/N1;
sigma2T_wbias=0.5*weightedsumATET_wbias/N1;
run;
%end;
%else %do;
  /** Estimating ATE and ATET **/
proc sort data=closestM;
by _id_;
data outcome_hat(keep=_id_ nbelements outcome_hat);
set ClosestM;
by _id_;
retain sum;
if first._id_ then sum=0;
sum=sum+outcomeM;
if last._id_ then do;
 outcome_hat=sum/nbelements;
 output;
end;
run;

data outcome_hat;
merge outcome_hat(in=x) control(keep=_id_ &y &w) treatment(keep=_id_ &y &w);
by  id ;
difoutcome=&w*(&y-outcome_hat)+(1-&w)*(outcome_hat-&y);

/** Averaging the differences to compute ATE and ATET **/

proc means data=outcome_hat noprint;
var difoutcome;
output out=ATE(keep=ATE) mean=ATE;
run;

proc means data=outcome_hat(where=(&w=1)) noprint;
var difoutcome;
output out=ATET(keep=ATET) mean=ATET;
run;

/**** Variance estimation ****/

/** 1: Outcome variance estimation under homoscedasticity **/

data ate;set ate;cst=1;
data atet;set atet;cst=1;
data ClosestM;set ClosestM;cst=1;

data ClosestM;
merge ate(keep=cst ate) atet(keep=cst atet) ClosestM;
by cst;

proc sort data=ClosestM;
by _id_;

data sigma2;
merge ClosestM(in=a) treatment(keep=_id_ &y &w) control(keep=_id_ &y &w);
by _id_;
if a;
retain sumATE sumATET N0 N1;
if first. id_ then do;
 sumATE=0;
 sumATET=0;
 N0=0;
 N1=0;
end;
difoutcome=&w*(&y-outcomeM)+(1-&w)*(outcomeM-&y);
sumATE=sumATE+(difoutcome-ate)**2;
sumATET=sumATET+&w*(&y-outcomeM-ateT)**2;
if last._id_ then do;
 weightedsumATE=sumATE/nbelements;
 weightedsumATET=sumATET/nbelements;
 N0=(&w=0);
 N1=(&w=1);
 output;
end;
run;
```

```
 proc summary data=sigma2;
 var weightedsumATE weightedsumATET N0 N1;
 output out=sigma2(drop=_type_ rename=(_freq_=N)) sum=;

 data sigma2;
 set sigma2;
 sigma2=0.5*weightedsumATE/N;
 sigma2T=0.5*weightedsumATET/N1;
 run;
%end;

data sigma2;set sigma2;cst=1;

 /**** 2. Outcome variance estimation allowing for heteroscedasticity *****/

 /**** Matching controls with controls and matching treated with treated ****/

proc iml;
start distance;
use Treatment;
read all var {&x} into XTreated;
read all var {&y} into outcomeT;
read all var {_id_} into idT;
close Treatment;
use Control;
read all var {&x} into XControl;
read all var {&y} into outcomeC;
read all var {_id_} into idC;
close Control;

Ntreated=nrow(XTreated);
Ncontrol=nrow(XControl);

/** Scaling for measuring the distance between the vector of covariates **/

use covarianceX;
read all var {&x} into S;
close covarianceX;

%let metric=&scaling;
%if &metric=1 %then %do;
 V=inv(S);  /* Mahalanobis distance */
 %end;
%else %if &metric=2 %then %do;
 V=i(ncol(Xtreated)); /* Euclidean distance */
%end;
%else %do;
 V=inv(diag(vecdiag(S))); /* Inverse variances, the default choice */
%end;

/** Treated with treated **/

distT=J(Ntreated#Ntreated,3,0);
do i=1 to Ntreated;
 disti=J(Ntreated,3,0);
  do l=1 to NTreated;
   disti[l,1]=sqrt( (XTreated[i,]-XTreated[l,])*V*(XTreated[i,]-XTreated[l,])` );
   disti[l,3]=outcomeT[l];
  end;
  disti[,2]=idT[i];
  distT[Ntreated#(i-1)+1:i#Ntreated,]=disti;
end;

varnames= {"DistanceM" "_id_" "OutcomeM"};
create distanceT_ from distT[colname=varnames ];
append from distT ;

free distT;

/** Controls with controls **/

distC=J(Ncontrol#Ncontrol,3,0);
do l=1 to Ncontrol;
 distl=J(Ncontrol,3,0);
  do i=1 to Ncontrol;
   distl[i,1]=sqrt( (Xcontrol[i,]-XControl[l,])*V*(Xcontrol[i,]-XControl[l,])` );
   distl[i,3]=outcomeC[i];
  end;
  distl[,2]=idC[l];
  distC[Ncontrol#(l-1)+1:l#Ncontrol,]=distl;
```

```
end;

varnames= {"DistanceM" "_id_" "OutcomeM"};
create distanceC_ from distC[colname=varnames];
append from distC;

free distC Xtreated XControl;

finish distance;
run distance;
quit;

/** Closest matches for treated with treated and controls with controls**/

data distance_;
set distanceT_ distanceC_;

proc sort data=distance_;
by _id_ distanceM;

data ClosestM1_ ClosestM2_;
set distance_;
by _id_ distanceM;
retain nbelements;
if first._id_ then nbelements=0;
nbelements=nbelements+1;
if nbelements<=&M+1 then output ClosestM1_;else output ClosestM2_;

data tie_(drop=nbelements);
set ClosestM1 (keep= id_ distanceM nbelements rename=(distanceM=distancetie));
where nbelements=&M+1;

data ClosestM2_(drop=distancetie);
merge ClosestM2_ tie_;
by _id_;
if distanceM=distancetie;

data ClosestM_;
set ClosestM1_(drop=nbelements) ClosestM2_(drop=nbelements);

proc summary data=ClosestM_ nway;
class _id_;
output out=nbelements(keep=_id_ _freq_ rename=(_freq_=nbelements));
run;

proc sort data=ClosestM_;
by _id_ distanceM;
run;

data ClosestM_;
merge nbelements ClosestM_;
by _id_;
run;

/* Conditional variances */

proc summary data=ClosestM_ nway vardef=df;
class _id_;
var outcomeM;
output out=var_(keep=_id_ Sigma2i) var=Sigma2i;
run;

data regfile;
merge var_ regfile;
by _id_;
run;

/** File with all variances and standard errors **/

data regfile;set regfile;cst=1;

%if &nbcont>=2 %then %do;
 data variance;
 merge sigma2 regfile;
 by cst;
 retain Variance Variance_wbias VarianceT VarianceT_wbias Variance_h VarianceT_h;
 if first.cst then do;
  Variance=0;
  Variance_wbias=0;
  VarianceT=0;
```

```
  VarianceT_wbias=0;
  Variance_h=0;
  VarianceT_h=0;
 end;
 Variance=Variance+sigma2*(1+km_i)**2;
 Variance_wbias=Variance_wbias+sigma2_wbias*(1+km_i)**2;
 VarianceT=VarianceT+sigma2T*(&w-(1-&w)*km_i)**2;
 VarianceT_wbias=VarianceT_wbias+sigma2T_wbias*(&w-(1-&w)*km_i)**2;
 Variance_h=Variance_h+sigma2i*(1+km_i)**2;
 VarianceT_h=VarianceT_h+sigma2i*(&w-(1-&w)*km_i)**2;
 if last.cst then do;
  variance=variance/(N0+N1)**2;
  variance_wbias=variance_wbias/(N0+N1)**2;
  varianceT=varianceT/N1**2;
  varianceT_wbias=varianceT_wbias/N1**2;
  variance_h=variance_h/(N0+N1)**2;
  varianceT_h=varianceT_h/N1**2;
  stderr_ATE=sqrt(variance);
  stderr_ATEwbias=sqrt(variance_wbias);
  stderr_ATET=sqrt(varianceT);
  stderr_ATETwbias=sqrt(varianceT_wbias);
  stderr_ATE_h=sqrt(variance_h);
  stderr_ATET_h=sqrt(varianceT_h);
  output;
 end;
%end;
%else %do;
 data variance;
 merge sigma2 regfile;
 by cst;
 retain Variance VarianceT Variance_h VarianceT_h;
 if first.cst then do;
  Variance=0;
  VarianceT=0;
  Variance_h=0;
  VarianceT_h=0;
 end;
 Variance=Variance+sigma2*(1+km i)**2;
 VarianceT=VarianceT+sigma2T*(&w-(1-&w)*km_i)**2;
 Variance_h=Variance_h+sigma2i*(1+km_i)**2;
 VarianceT_h=VarianceT_h+sigma2i*(&w-(1-&w)*km_i)**2;
 if last.cst then do;
  variance=variance/(N0+N1)**2;
  varianceT=varianceT/N1**2;
  variance_h=variance_h/(N0+N1)**2;
  varianceT_h=varianceT_h/N1**2;
  stderr_ATE=sqrt(variance);
  stderr_ATET=sqrt(varianceT);
  stderr_ATE_h=sqrt(variance_h);
  stderr_ATET_h=sqrt(varianceT_h);
  output;
 end;
%end;

/** Editing results **/

data result;
set ATE;set ATET;set variance;

proc iml;
start editing;
%if &nbcont>=2 %then %do;
 use result;
 read all var{ATE ATE_withoutBIAS ATET ATET_withoutBIAS} into coef;
 read all var{stderr_ATE stderr_ATEwbias stderr_ATET stderr_ATETwbias} into StdErr;
 read all var{stderr_ATE_h stderr_ATET_h} into StdErr_h;
 read all var{N0} into N0;
 read all var{N1} into N1;
 close result;
%end;
%else %do;
 use result;
 read all var{ATE ATET} into coef;
 read all var{stderr_ATE stderr_ATET} into StdErr;
 read all var{stderr_ATE_h stderr_ATET_h} into StdErr_h;
 read all var{N0} into N0;
 read all var{N1} into N1;
 close result;
%end;
z=coef`/StdErr`;
```

```
%if &nbcont>=2 %then %do;
 rowcol={1 1 0 0,0 0 1 1};
 StdErr_h=StdErr_h*rowcol;
%end;
z_h=coef`/StdErr_h`;

pvalue=round(2*(1-probnorm(abs(z))),.001);
pvalue_h=round(2*(1-probnorm(abs(z_h))),.001);

Lbound=round(coef`-quantile('normal',.975)#StdErr`,.001);
Ubound=round(coef`+quantile('normal',.975)#StdErr`,.001);

Lbound_h=round(coef`-quantile('normal',.975)#StdErr_h`,.001);
Ubound_h=round(coef`+quantile('normal',.975)#StdErr_h`,.001);

coef_=round(coef,.001);
z_=round(z,.001);
z_h_=round(z_h,.001);
StdErr_=round(StdErr,.001);
StdErr_h_=round(StdErr_h,.001);

Results=coef_`||StdErr_`||z_||pvalue||LBound||UBound;
Results_h=coef_`||StdErr_h_`||z_h_||pvalue_h||LBound_h||UBound_h;

use regfile;
read all var {&w count};
read all var {count} into countT where(&w=1 & count>0);
read all var {count} into countC where(&w=0 & count>0);
close regfile;

nbmatchesT=nrow(countT);
nbmatchesC=nrow(countC);

maxT=countT[<>];
minT=countT[><];
maxC=countC[<>];
minC=countC[><];

use scaling;
read all var _all_ into scaling;
close scaling;

N=N0+N1;

ModelSpecification={"&y","&w","&x"};
SummaryStatistics=N//N0//N1//nbmatchesT//nbmatchesC//minT//maxT//minC//maxC;

TitleResult={"Estimate","Std.Error","z","P-value","L. bound 95% CI","U. bound 95% CI"};
%if &nbcont>=2 %then %do;
 EstimationOptions={"&M"}//Scaling//{"&xbias"};
 Effect={"Average Treatment Effect (ATE)","ATE with bias correction",
         "Average Treatment Effect for the Treated (ATET)","ATET with bias correction"};
%end;
%else %do;
 EstimationOptions={"&M"}//Scaling//{"Not any ! Bias-correction not needed"};
 Effect={"Average Treatment Effect (ATE)", "Average Treatment Effect for the Treated (ATET)" };
%end;

TitleModel={"Outcome variable:","Binary treatment:","Matching variables:"};
TitleOptions={"Number of matches requested:","Scaling matrix used:","Covariates used for bias
correction:"};
TitleSummary={"Number of observations:","Number of control units:","Number of treated units:",
              "Number of treated units matched to controls:",
               "Number of control units matched to treated:",
               "Number of times a treated unit is used as a match (MIN):",
               "Number of times a treated unit is used as a match (MAX):",
               "Number of times a control unit is used as a match (MIN):",
               "Number of times a control unit is used as a match (MAX):"};

print "ESTIMATING AVERAGE TREATEMENT EFFECTS",,
      ModelSpecification [rowname=TitleModel label="Model specification"],,
      EstimationOptions [rowname=TitleOptions label="Estimation options"],,
      SummaryStatistics [rowname=TitleSummary label="Summary statistics"],,
      Results [colname=TitleResult rowname=Effect label="Estimation results assuming
homoscedasticity of the conditional variance"],,
      Results_h [colname=TitleResult rowname=Effect label="Estimation results allowing for
heteroscedasticity of the conditional variance"] ;
finish editing;
run editing;
```

```
/*** Checking balance : standardized differences for comparing means between controls and treated
before and after matching. Formulas for "continuous" and binary variables are different (see Austin,
P. C. (2009)). ***/

proc iml;
start balance;

use regfile;
read all var {&x} where(&w=0) into XC;
read all var {&x} where(&w=1) into XT;
read all var {&x} where(&w=0 & km_i>0) into XCmatched; /* covariates for controls used as matches
for treated */
read all var {&x} where(&w=1 & km_i>0) into XTmatched; /* covariates for treated used as matches for
controls */
close regfile;

use n_levels;
read all var {Nlevels} into Nlevels;
close n_levels;

binary=(Nlevels=2);
k=ncol(xc);

MeanXC=XC[:,];
MeanXT=XT[:,];

meanXCmatched=XCmatched[:,];
meanXTmatched=XTmatched[:,];

VarXC=Var(XC);
VarXT=Var(XT);

VarXCmatched=Var(XCmatched);
VarXTmatched=Var(XTmatched);

 /** Formulas for continuous and binary variables **/

Norm Diff Before=J(k,1,0);
Norm_Diff_TCmatches=J(k,1,0);
Norm_Diff_CTmatches=J(k,1,0);

do i=1 to k;
 if binary[i]=0 then do;
  Norm_Diff_Before[i]=round( t((meanXT[,i]-meanXC[,i])/sqrt((varXT[,i]+varXC[,i])/2)) ,.001);
  Norm_Diff_TCmatches[i]=round( t((meanXT[,i]-
meanXCmatched[,i])/sqrt((varXT[,i]+varXCmatched[,i])/2)) ,.001);
  Norm Diff CTmatches[i]=round( t((meanXTmatched[,i]-
meanXC[,i])/sqrt((varXTmatched[,i]+varXC[,i])/2)) ,.001);
 end;
 else do;
  Norm_Diff_Before[i]=round( t( (meanXT[,i]-meanXC[,i])/sqrt( (meanXT[,i]#(1-
meanXT[,i])+meanXC[,i]#(1-meanXC[,i]))/2 ) ) ,.001);
  Norm_Diff_TCmatches[i]=round( t( (meanXT[,i]-meanXCmatched[,i])/sqrt( (meanXT[,i]#(1-
meanXT[,i])+meanXCmatched[,i]#(1-meanXCmatched[,i]))/2 ) ) ,.001);
  Norm Diff CTmatches[i]=round( t( (meanXTmatched[,i]-meanXC[,i])/sqrt( (meanXTmatched[,i]#(1-
meanXTmatched[,i])+meanXC[,i]#(1-meanXC[,i]))/2 ) ) ,.001);
 end;
end;

Normalized_Differences=Norm_Diff_Before||Norm_Diff_TCmatches||Norm_Diff_CTmatches;

Variables={&x}`;
Title={"Unmatched Sample" "Matched sample (T)" "Matched sample (C)"};
print Normalized_Differences [colname=Title rowname=Variables label="Normalized covariate mean
differences between treated and controls"],,
      "Note: 'Matched sample (T)' is for normalized mean differences between all sample treated and
their matches, 'Matched sample (C)' for
       normalized mean differences between all sample controls and their matches.";
finish balance;
run balance;

/** Checking balance : visual diagnostics for continuous variables and contingency tables for binary
covariates **/

 data sampleT0;
 set regfile(keep=&w &x where=(&w=1));
 SampleType=1;
 data sampleC0;
 set regfile(keep=&w &x where=(&w=0));
 SampleType=2;
```

```
 data sampleTmatched;
 set regfile(keep=&w &x km_i where=(&w=1 and km_i>0));
 SampleType=3;
 data sampleCmatched;
 set regfile(keep=&w &x km_i where=(&w=0 and km_i>0));
 SampleType=4;

 data sample;
 set sampleT0 sampleC0 sampleTmatched sampleCmatched;
 proc sort data=sample;
 by SampleType;
 run;

 proc format;
 value sample
      1="All treated"
         2="All controls"
         3="Matched controls"
         4="Matched treated";
 run;
/* Identifying continuous and binary variables and picking up their names */
/* See http://support.sas.com/kb/45/626.html "Using Macro to create new variable names from variable
values" */

%if &nbbinary>0 %then %do;
 data binaryvar;
 set n_levels(where=(nlevels=2));
 run;
 data _null_;
 set binaryvar end=end;
 count+1;
 call symputx('varb'||left(count),TableVar);
 if end then call symputx('maxb',count);
 run;
 %macro binaryvars;
 data binaryvar;
 set binaryvar end=end;
 %do i = 1 %to &maxb;
  if _n_=&i then do;
    &&varb&i=0;
    retain &&varb&i;
    keep &&varb&i;
  end;
 %end;
 if end then output;
 %mend;

 %binaryvars;

 proc contents data=binaryvar out=binaryvar(keep =varnum name) noprint;
 run;
 proc sql noprint;
 select name
 into :xbinary separated by " " from binaryvar order by varnum;
 quit;
 title "Comparing binary covariate distributions between treatment groups before and after
matching";
 proc tabulate data=sample;
 class sampletype &xbinary;
 table (&xbinary)*COLPCTN,sampletype;
 keylabel COLPCTN='Distribution in %';
 format sampletype sample.;
 run;
%end;
%if &nbcont>0 %then %do;
 data contvar;
 set n_levels(where=(nlevels>2));
 run;
 data _null_;
 set contvar end=end;
 count+1;
 call symputx('var'||left(count),TableVar);
 if end then call symputx('max',count);
 run;
 %macro contvars;
 data contvar;
 set contvar end=end;
 %do i = 1 %to &max;
  if _n_=&i then do;
    &&var&i=0;
```

```
    retain &&var&i;
    keep &&var&i;
  end;
 %end;
 if end then output;
 %mend;

 %contvars;

 proc contents data=contvar out=contvar(keep =varnum name) noprint;
 run;
 proc sql noprint;
 select name
 into :xcont separated by " " from contvar order by varnum;
 quit;

 ODS GRAPHICS OFF;
 TITLE;
 proc boxplot data=sample;
 plot (&xcont)*sampleType/ nohlabel;
 title "Checking balance: visual diagnostic for continuous variables";
 format sampleType sample.;
 run;
 ODS GRAPHICS ON;
 proc npar1way edf plots(nostats)=edfplot data=sample(where=(sampleType in (1,2)));
 class sampleType ;
 var &xcont;
 title "Checking balance through empirical distribution functions";
 title2 "Original sample of treated vs original sample of controls";
 format sampleType sample.;
 run;
 proc npar1way edf plots(nostats)=edfplot data=sample(where=(sampleType in (1,3)));
 class sampleType ;
 var &xcont;
 title "Checking balance through empirical distribution functions";
 title2 "Original sample of treated vs final sample of matched controls";
 format sampleType sample.;
 run;
 proc npar1way edf plots(nostats)=edfplot data=sample(where=(sampleType in (2,4)));
 class sampleType ;
 var &xcont ;
 title "Checking balance through empirical distribution functions";
 title2 "Original sample of controls vs final sample of matched treated";
 title;
 format sampleType sample.;
 run;
 quit;
%end;

/** Output data files **/

data outdata1;
set regfile(keep=_id_ &y &w &x km_i count nbelements);

data outdata2;
set closestM(keep=_id_ idM outcomeM distance);
run;
%mend nn_matching;
```