



Working Paper no. **169**

**PSMatching: A SAS Macro for Treatment Effect Estimation and  
Missing Data Imputation Based on Propensity Score Matching**

Roberto Leombruni  
University of Torino and LABORatorio R. Revelli

Michele Mosca  
LABORatorio R. Revelli and Technos Italia

July, 2019

# PSMatching: A SAS Macro for Treatment Effect Estimation and Missing Data Imputation Based on Propensity Score Matching

---

Roberto Leombruni – University of Turin and Laboratorio Revelli  
Michele Mosca – Laboratorio Revelli and Technos Italia

## Abstract

Matching estimators based on the propensity score are widely used in the field of treatment effect evaluation and a viable technique also for missing data imputation. This paper describes an implementation of the technique in SAS®, a statistical software where only limited implementations of it are currently available. The user can choose among the most common variants of the matching algorithms (nearest neighbour-, stratification- and kernel matching), and the main pre- and post estimation analyses proposed in the literature (reduction of standardised bias, Sianesi test, Ichino-Becker test on the balancing hypotheses, Lechner bounds). To these, two additional diagnostics are proposed in order to better monitor some aspects of the matching process. A validation of the procedure is made using the available STATA tools as a benchmark, with both artificial data and data already used in the literature.

Randomized trials are commonly held to be the “gold standard” in the empirical assessment of causal relationships, due to the complete control the researcher has on the experimental setting. The data available to researchers, however, are only rarely the outcome of experiments. In a seminal work, Rosenbaum and Rubin (1983) proposed propensity score matching as a method to mimic randomization in observational studies which has become increasingly popular in the evaluation of policy interventions and in many other fields (Sekhon 2011). The basic idea is to form groups of treated and controls who are similar with each other not because of ex-ante randomization, but as the ex-post result of matching each treated unit to untreated ones who are observationally similar to them. The same methodology can be applied also for missing data imputation, where a unit is considered “treated” when one or more of its characteristics are not observed (Little and Rubin, 1987). In this case, the aim of the matching is to form pairs of “receivers” and “donors”, so that the missing information for any unit is provided by a donor as similar as possible to it.<sup>1</sup>

---

<sup>1</sup> As Rosenbaum and Rubin (1983) put it, causal evaluation itself can be regarded as a missing-data problem, since one would like to compare what happened to a subject in the factual, observed situation, with what would have happened in a counterfactual, not observed one.

In this paper we describe the %PSMatching macro, an implementation of propensity score matching in a statistical software – the SAS® system – where only limited implementations of it are currently available<sup>2</sup>. The main outcomes of the macro are the “Average effect of Treatment on the Treated” (*ATT*), which is the main quantity of interest for the evaluation of causal relationships, and counterfactual values for the variable of interest at the individual level, which are needed to adopt the methodology in the context of missing data imputation. Besides the most common matching estimators, and the pre- and post-estimation diagnostics proposed in the literature, the macro adds a monitor at the unit level on the quality of the matching and on the number of matches delivered, which we argue are useful both to assess the overall quality of the match, and to support the bandwidth and caliper selection.

In the next section, we briefly describe the methodology and introduce notation. In sections two to four we review the matching estimators, standard error estimators and diagnostics discussed in the literature and implemented in the macro. In section five we present the further diagnostics we implemented and a validation of the procedure both on artificial data and on datasets already used in the literature, using current STATA implementations of PS matching as a benchmark. Concluding remarks are provided, and an Appendix with the user-manual of the macro.

## 1. Introduction to Propensity Score Matching

According to the potential outcome notation, let us call  $Y_{1i}$  the value of an outcome of interest we would observe for individual  $i$  in the case s/he received a given treatment – for instance a welfare programme, or a new drug – and  $Y_{0i}$  what we would observe for the same individual in the absence of the treatment. The “fundamental problem of causal inference” (Holland 1986) is that we either observe  $Y_{1i}$  or  $Y_{0i}$ , so that the effect of the treatment, which is defined as  $(Y_{1i} - Y_{0i})$ , is not observable at the individual level. The basic idea of matching is to estimate it using as a counterfactual for treated units the outcome on non-treated “twins”, *i.e.* subjects who besides the treatment are as similar as possible to them.

The matching process may aim at finding exact matches when the observable characteristics on which similarity is defined are few and categorical. With higher-dimensional vectors of characteristics, however, this is no more feasible – the so called “curse of dimensionality” – and two options are at hand. The first is to define a metric to measure the distance between units and look for minimum-distance matches, as in Mahalanobis distance matching. In their seminal paper, Rosenbaum and Rubin (1983) proposed a different strategy, which consists in collapsing all characteristics into a single variable (the propensity score), and looking for twins as similar as possible in terms of it.

---

<sup>2</sup> See Yang *et al* (2007) and Elizabeth Stuart’s page at <http://www.biostat.jhsph.edu/~estuart/propensityscoresoftware.html> for a review of propensity score implementations in several statistical packages. On the SAS® System, currently available tools offer just basic implementations of the technique, without CI estimates and pre- and post-estimation diagnostics. See however Coca-Perraillon (2007) and Kosanke and Bergstralh (2004) for propensity score matching using “greedy” algorithms, to optimally allocate controls in case of matching without replacement, not covered in our implementation.

In their words, “[t]he propensity score allows to convert the multidimensional setup of matching into a one-dimensional setup”.

The propensity score is defined as the conditional probability of receiving the treatment given pre-treatment characteristics:

$$p(X) \equiv \Pr(D = 1 | X) = E(D | X) \quad (1)$$

where  $D = \{0, 1\}$  is the indicator of exposure to treatment and  $X$  is the vector of pre-treatment characteristics. What Rosenbaum and Rubin showed is that if pre-treatment variables are balanced among treated and controls conditioning on the propensity score:

$$\text{Balancing Hypothesis} \quad D \perp X | p(X) \quad (2)$$

and the exposure to treatment is random conditioning on  $X$ :

$$\text{Unconfoundedness} \quad Y_1, Y_0 \perp D | X \quad (3)$$

than the exposure to treatment can be considered random directly conditioning on the one-dimensional variable  $p(X)$ . As a result, one can mimic a randomized trial comparing individuals which are similar with respect to the propensity score only, and the Average effect of Treatment on the Treated ( $ATT$ ) can be estimated as follows:

$$ATT = E(E(Y_1 | D=1, p(X)) - E(Y_0 | D=0, p(X))) \quad (4)$$

where the outer expectation is computed over the distribution of  $p(X)$  on the treated.

Referring the reader to the literature for a more thorough introduction of the technique, it is useful here to outline the one technical issue which is relevant for applying the method also for imputation purposes.

In a very stylized way, the technique can be implemented in two steps: first, estimate the propensity score as a function of the observables; second, match each treated with one or more controls using the estimated propensity scores as a metric. When one or more matches have been found, one may either (1) select them into a set of matched controls (eventually assigning them a weight, as in kernel matching) or (2) use their average outcome to impute a counterfactual value for the treated, and estimate a unit level treatment effect.

To the purpose of estimating the  $ATT$ , the two options are equivalent. In the first case the  $ATT$  will be computed as the difference in the average outcome among the treated and the matched controls ( $ATT = E(Y | D=1) - E[Y | D=0]$ ), in a randomized trial fashion. In the second one, it will be computed as the average of the unit level treatment effects ( $ATT = E[(Y_1 - Y_0^*) | D=1]$ ), where  $Y_0^*$  is the estimated counterfactual.

For imputation purposes, the second option is to be implemented, since the aim of the technique is exactly to impute to each unit with missing data a counterfactual value, that is, the outcome in the case it was not missing. In this set-up, the propensity score should be estimated modelling the missing data generation process and during the matching procedure counterfactual values should be saved as imputation values.

Hence, although for *ATT* estimation it could suffice to save weights as in option (1), to meet missing data imputation aims we compute also counterfactual values as in option (2). For the sake of comparison, in the STATA implementation proposed by Becker and Ichino (2002) individual level counterfactual values are not available, while in the STATA implementation proposed by Sianesi (2004) they are.

## 2. Propensity Score Dialects

There are many ways in which the propensity score may be used to build counterfactual estimators: as a metric in nearest neighbour matching; as a conditioning variable in stratification (or subclassification) matching; to compute weights in kernel matching. In the following we briefly review the features of the algorithms we implemented in %PSMatching.

### *Nearest neighbour matching*

The following two methods were already summed up in Rosenbaum and Rubin (1985):

*Nearest available matching on the estimated propensity score.* The method consists in selecting at random a treated and searching for the control with the closest propensity score. Both units are then removed from the sample and the procedure is repeated till all treated units have been processed.

*Nearest available Mahalanobis metric matching within calipers defined by the propensity score.* A first treated is selected at random. All controls within a given propensity score distance (a “caliper”) are selected, and Mahalanobis distances are calculated between these units and the treated. The closest control and the treated are then removed from the sample, and the procedure is repeated.

Several variations of the techniques have been developed, for example relaxing the non-replacement option; using the caliper-option also in the first variant; considering the *N* nearest controls instead of only the first one (*1-N Matching*). To achieve a flexible choice of the estimators we implemented the two main matching algorithms (nearest neighbour in terms of the propensity score or in terms of Mahalanobis distance) and left the user the possibility of specifying a caliper and/or a replacement option.

### *Stratification based on the propensity score*

This is a variant of sub-classification matching (Cochran 1968; Cochran and Rubin 1973), where the strata are formed conditioning on the propensity score instead than on

multiple covariates (Rosenbaum and Rubin, 1983). The sample of treated and controls is split in  $Q$  propensity score intervals, choosing a partitioning such that in each block the covariates are balanced. For each block then an average treatment effect is computed:

$$ATT_q = \frac{\sum_{i \in I(q)} Y_i^T}{N_q^T} - \frac{\sum_{j \in I(q)} Y_j^C}{N_q^C}$$

where  $I(q)$  is the set of units in the block, and  $N_q^T$  and  $N_q^C$  are the numbers of treated and controls in the block. The overall  $ATT$  estimate is then computed as:

$$ATT = \sum_{q=1}^Q ATT_q \frac{\sum_{i \in I(q)} D_i}{\sum_{\forall i} D_i}$$

that is, as a weighted average of the treatment effects estimated in the blocks, where the weights are the shares of treated units in a block on the total.

### *Kernel matching*

A further variant of propensity score based estimators, called kernel matching, has been proposed by Heckman, Ichimura and Todd (1997, 1998) and Heckman *et al* (1998). Instead of comparing each treated with one or few controls they are matched with all of them, weighting their outcome with a kernel function which is decreasing in the distance between the treated's and controls' propensity score:

$$ATT = \frac{1}{N^T} \sum_{i \in T} \left\{ Y_i^T - \frac{\sum_{j \in C} Y_j^C G\left(\frac{p_j - p_i}{h}\right)}{\sum_{k \in C} G\left(\frac{p_k - p_i}{h}\right)} \right\}$$

where  $G(\cdot)$  is a kernel function and  $h$  is a bandwidth parameter. The available options about the kernel function are in Appendix A4.

Dehejia and Wahba (2002) suggested a further variant, called *radius matching*, which consists in matching all the controls within a given distance (the radius). A benefit of this approach is that it uses as many comparison units as are available within the radius and therefore allows for usage of extra (fewer) units when good matches are (not) available. Hence, it shares the attractive feature of oversampling, but avoids the risk of bad matches. Note, however, that the method is equivalent to kernel matching using the uniform density as a kernel and a bandwidth equal to the radius.

### 3. The Computation of *ATT* Confidence Intervals

In the case of stratification matching, an analytical approximation of the standard error of the *ATT* estimator has been proposed by Becker and Ichino (2002):

$$\hat{\sigma}^2 = \frac{1}{N^T} \left[ \text{Var}(Y^T) + \sum_{q=1}^Q \frac{N_q^T}{N^T} \frac{N_q^T}{N^C} \text{Var}(Y^C) \right]$$

The formula does not take into account the estimator variability due to the number of strata and their thresholds, which too are defined within the algorithm. The latter variability could be grasped by bootstrapping the entire estimator, defining at each iteration a new partition of the data and estimating the standard error. An issue with this is that within each iteration the balancing hypothesis should be tested, too, but in the case it is rejected the researcher can not try a different specification for the propensity score estimator<sup>3</sup>. On the other side, the above formula shows a good performance in the simulations we propose below.

Bootstrapped standard errors are the most common estimators employed in the literature in the case of kernel matching. While this was common also in the case of one-to-one matching, Abadie and Imbens (2008, 2006a, 2006b) proved the failure of the bootstrap in this setting. The authors recommend to use, as an alternative, either subsampling bootstrap (Politis and Romano, 1994), or one of the variance estimator approximations they proposed in Abadie and Imbens (2006a). In %PSMatching we considered one of them, which writes:

$$\hat{\sigma}^2 = \frac{1}{N_T^2} \sum_{i=1}^N (D_i - K_i)^2 \hat{\sigma}_i^2(X_i, D_i)$$

where  $K_i$  is the weighted number of times unit  $i$  is used as a match and  $\hat{\sigma}_i^2(X_i, D_i)$  is an estimator based on matching of the (unit level) conditional variance of  $Y_i$  given  $D_i$  and  $X_i$ . Calling  $l(i)$  each of the  $J$  closest matches to unit  $i$ , among the units with the same value for the treatment (that is, treated are matched to treated, and controls are matched to controls), the estimator is defined as follows:

$$\hat{\sigma}_i^2(X_i, D_i) = \frac{J}{J+1} \left( Y_i - \frac{1}{J} \sum_{j=1}^J Y_{l_j(i)} \right)^2$$

which, assuming a constant treatment effect and homoskedasticity, simplifies to:

---

<sup>3</sup> Actually, in Becker and Ichino (2002) the number of strata and their thresholds are kept constraint across bootstrap iterations. In %PSMatching bootstrap estimates are available in which we iterate the entire process, but our suggestion is for the analytical approximation.

$$\hat{\sigma}^2 = \frac{1}{2N_T} \sum_{i:D_i=1} \left\{ \frac{1}{J_M(i)} \sum_{l \in J_M(i)} (Y_i - Y_l - ATT)^2 \right\}$$

where  $J_M(i)$  denote the set of indices for the matches for unit  $i$  which are at least as close as the  $M$ -th match (if there are no ties, the number of elements is  $M$ ).

#### 4. Pre- and post-estimation diagnostics

The main hypothesis on which PS-based estimators rely on is unconfoundedness, but there are no automatic tests of it: it is upon the researcher to argue that the selection into treatment depends (at most) on observable characteristics. Besides this, a good performance of the estimators requires both a sufficient overlap of the the propensity score distributions among the treated and controls, and a good performance of matching in achieving the balance in treated and controls' observable characteristics. In the following, we briefly present which are the tests we implemented among those proposed in the literature to address these latter two points.

##### *Common Support*

The common support, or overlap condition, serves the purpose of ruling out the perfect predictability of  $D$  given  $X$ :

$$0 < P(D = 1 | X) < 1$$

It ensures that units with the same  $X$  values have a positive probability of being both participants and non-participants (Heckman, LaLonde, and Smith, 1999). From a practical point of view, we rarely know the true value of the propensity score, and moreover, at low (high) values of the probability of being treated, we may empirically find no treated (controls) in the sample. In %PSMatching, similarly to what is done in other implementations, the issue is addressed in two ways. The first is to produce the empirical density functions of the propensity score conditional on the treatment, for an eyeball evaluation of their overlap. The second is to allow the user to restrict the analysis to a “common support” region, which is the estimated propensity score interval where both treated and controls are observed<sup>4</sup>.

##### *Lechner Bounds*

The drawback of restricting the estimation of the  $ATT$  to only those units for which a match is available is the loss of external validity. Lechner (2000) describes an approach to check the robustness of estimated treatment effects due to a non complete match. He incorporates information from those treated individuals who failed to be matched, to

---

<sup>4</sup> Note that also imposing a caliper is a way to restrain the propensity score distance at which treated and controls may be matched. We allow the user to ask for both a caliper and a common support region: the latter, in this case, is extended by the length of the caliper. See Appendix A2 for details.

calculate non-parametric bounds for the  $ATT$  if all individuals from the sample at hand would have been included.

The  $ATT$  bounds are built around corresponding bounds defined for the potential outcome  $Y_0$ . For example, when the outcome variable is a binary indicator, a lower bound  $\underline{Y}_0$  can be set at zero and an upper bound  $\overline{Y}_0$  can be set at one; or, for continuous variables, one could use the minimum and the maximum outcome observed within the control group. The idea is to obtain a lower (upper) bound for the  $ATT$  using the upper (lower) bound of the potential outcome as a counterfactual for non-matched units:

$$\overline{ATT} = E(Y_1 - Y_0^* | M = 1, D = 1)Pr(M = 1 | D = 1) + E(Y_1 - \underline{Y}_0 | M = 0, D = 1)Pr(M = 0 | D = 1)$$

$$\underline{ATT} = E(Y_1 - Y_0^* | M = 1, D = 1)Pr(M = 1 | D = 1) + E(Y_1 - \overline{Y}_0 | M = 0, D = 1)Pr(M = 0 | D = 1)$$

where  $M$  indicates the units for which a match have been found.

### *Reduction in the Standardized Differences*

The original indicator suggested by Rosenbaum and Rubin (1985) to assess the balancing achieved between treated and controls is the standardised bias, or standardized differences (SD). For each covariate  $X$  it is defined as the difference of sample means in the treated and matched control subsamples as a percentage of the square root of the average of sample variances in both groups. The standardised bias before matching is given by:

$$SD_{before} = 100 \frac{(\bar{X}^T - \bar{X}^C)}{\sqrt{0.5(V^T(X) + V^C(X))}}$$

The standardised bias after matching is given by:

$$SD_{after} = 100 \frac{(\bar{X}^{TM} - \bar{X}^{CM})}{\sqrt{0.5(V^{TM}(X) + V^{CM}(X))}}$$

where  $\bar{X}^T$  ( $V^T$ ) is the mean (variance) in the treatment group before matching and  $\bar{X}^C$  ( $V^C$ ) the analogue for the control group.  $\bar{X}^{TM}$  ( $V^{TM}$ ) and  $\bar{X}^{CM}$  ( $V^{CM}$ ) are the corresponding values for the matched samples.

As regards the value of a standardized difference that should be judged as a signal of bad balancing, Austin (2009) showed that a value of 10% is equivalent to a correlation coefficient of 0.05 (indicating negligible correlation) between the treatment variable and the covariate. Caliendo *et al* (2014) cite also 3% and 5% as commonly used thresholds.

### *Ichino-Becker Test of the Balancing Hypothesis*

Another indicator of the level of similarity achieved between treated and controls has been proposed by Becker and Ichino (2002), who suggested to test the Balancing Hypothesis using the following algorithm:

1. Estimate the propensity score.
2. Split the sample into  $Q$  equally spaced intervals of the propensity score.
3. Within each interval, test that the average propensity score of treated and controls does not differ.
4. If the test fails in one interval, split it in half and test again.
5. Continue until the propensity score is balanced in all intervals.
6. Within each interval, test that the means of each characteristic are also balanced between treated and controls.

If the means of one or more variables differ, the authors suggest that a better propensity score model should be specified and estimated, and the test delivered again. In their implementation, the t-test on the differences in the averages is computed assuming that the propensity score and the observed variables have an equal variance among the treated and the controls. Since this is a somewhat stringent assumption, in %PSMatching we implemented also the Satterthwaite approximation for the variance and the degrees of freedom of the test (see for example Khattree and Naik, 1999).

### *Sianesi Quality Test*

Sianesi (2004) suggested a further test to assess how well the matching process was able to balance treated and controls. The suggestion is to re-estimate the propensity score on the matched sample, *i.e.* only on participants and matched non-participants, and compare the pseudo- $R^2$  obtained before and after matching. The idea is that since the pseudo- $R^2$  indicates how well the regressors  $X$  explain the participation probability, after matching there should be no systematic differences in the distribution of covariates between groups and therefore the pseudo- $R^2$  should be fairly low. One can also perform an F-test on the joint significance of all regressors. The test should not be rejected before, and should be rejected after matching.

## **5. Further implementation details and validation**

The %PSMatching macro offers a flexible implementation of all the propensity score based estimators, the standard error estimators and of the suite of pre- and post-estimation diagnostics illustrated in previous sections. The user may obtain either estimated counterfactuals at the individual level, which is needed for imputation purposes, or call the macro asking just for *ATT* estimation. All outputs are saved also as temporary datasets, which is useful in the design of Monte Carlo studies. A full user-manual of the macro is in the Appendix.

To the post-estimation diagnostics proposed in the literature, %PSMatching computes two more indicators, which, to our knowledge, are not available in most implementations of the technique, and which may be useful in monitoring and fine-tuning the functioning of the matching process:

- In the case of nearest neighbour estimators, the propensity score (or mahalanobis) distance at which a nearest neighbour was found is saved. The average value of it is presented among the outputs, but also the individual level values are saved, so that the distribution of it may be studied. High average distances may alert the user about a potential bias in the *ATT* estimates; fat tails in the distribution may suggest the introduction of a caliper.
- In the case of kernel matching estimators, the number of matched controls (*i.e.*, those falling within the bandwidth) is saved. Again, both the average value and the individual level values are saved. A high average number of controls matched to the treated may suggest the user a reduction of the bandwidth, to achieve a better mix along the bias/precision trade-off; a high density of the distribution near a low number of matched controls may suggest an issue of common support.  
The variable is created also in the case of nearest neighbour matching, in which case it monitors how many ties are found in terms of the propensity score.

Another option which has been included is the possibility to choose, in the case of bootstrapped standard errors, whether the propensity score should be re-estimated at each bootstrap cycle. What is the best option is questionable: on the one side, also the propensity score estimation is a source of variability which should be taken into account; on the other side, the user has always the option to provide his/her own propensity score estimates, and the ones obtained in the original sample are the best available. We did not find any discussion in the literature on this point, and the simulations we did revealed negligible differences in the standard error estimates with the option on or off. Since it can achieve a huge improvement in computation time, we left to the user the choice on this option.

The validation of the macro has been made using as a benchmark the STATA programs provided by Becker and Ichino (BI 2002: PSCORE, ATTND, ATTNW, ATTR, ATTS and ATTK) and by Leuven and Sianesi (LS: PSMATCH2):

- The BI programs implement all the estimators considered here, with the exception of Mahalanobis matching within caliper, and of all estimators without replacement. A limit of the implementation is that it does not save counterfactual values at the individual level, so that it can not be used for imputation purposes. The standard errors are computed both by bootstrap, and with an analytic formula different from the ones proposed by Abadie and Imbens.
- The PSMATCH2 program is a very comprehensive package filling some gaps in the BI ones: it can be used also for data imputation, implements Mahalanobis matching and allows the choice between replacement and non-replacement. On the other side,

it does not implement strata matching. There exists different releases of it: we tested the last available (release 4.0.4 10nov2010) and an older one (release 3.1.2 20dec2006). An issue with both versions emerged about the Mahalanobis matching within calipers. In the 4.0.4 version only calipers in terms of Mahalanobis distance are available; while in the 3.1.2 one the calipers may be expressed in term of the propensity score, but when we tested it we found some matched control falling outside the caliper. For the sake of our tests, we used the 4.0.4 version, setting calipers sufficiently high so that they are never binding.

No standard errors are computed, which is a limit in its application to one-to-one estimators. Bootstrapped ones on the contrary may be obtained using a STATA native command (bs).

To validate the algorithms we implemented, we first compared the *ATT* estimators as computed by %PSMatching, BI and LS implementations on real data, using the publicly available data from Dehejia and Wahba (1999), which is based on LaLonde's (1986) seminal study of the comparison between experimental and non-experimental methods for the evaluation of causal effects<sup>5</sup>. The data combine treated units from a randomized evaluation of the National Supported Work (NSW) demonstration with non-experimental comparison units drawn from survey data. For the purpose of our tests, we restricted our analysis to the so-called NSW-PSID-1 subsample, consisting of the male NSW treatment units and the largest of the three PSID subsamples (see Dehejia and Wahba 1999 for more details). For the sake of comparison we excluded duplicated controls in terms of the propensity score – *i.e.* in case of ties we kept only one control – since the ties handling is different between BI and LS. We tested two nearest neighbour estimators (nearest propensity score neighbour and nearest Mahalanobis distance neighbour within caliper), two kernel estimators (Gaussian and Epanechnikov), and stratification matching. The three implementations produced estimates which were either perfectly equal or showing negligible differences (see Table 1).

**Table 1: *ATT* matching estimates under different algorithms and implementations**

| Estimation's algorithm                                  | algorithm's implementation |          |             |
|---------------------------------------------------------|----------------------------|----------|-------------|
|                                                         | IB                         | LS       | %PSMatching |
| NN in terms of PS                                       | 2.365,25                   | 2.365,25 | 2.365,25    |
| NN in terms of Mahalanobis distance                     | -                          | 1.544,34 | 1.544,34    |
| NN in terms of Mahalanobis distance within a PS caliper | -                          | 1.396,07 | 1.396,07    |
| Kernel matching (Epanechnikov)                          | 1.162,15                   | 1.162,15 | 1.162,11    |
| Kernel matching (Gaussian)                              | 811,56                     | 811,56   | 811,54      |
| Stratification method - Pooled variance                 | 1.969,12                   | -        | 1.969,12    |
| Stratification method - Satterthwaite variance          | -                          | -        | 1.221,37    |

Notes: data from Dehejia and Wahba (1999), excluding duplicated controls in terms of the propensity score.

The estimates in columns 2 to 4 are obtained using the implementations of Becker and Ichino (2002, column "IB"), Leuven and Sianesi (column "LS") and our own %PSMatching macro.

<sup>5</sup> The data are available on Rajeev Dehejia's web site under the following address: <http://www.columbia.edu/~rd247/nswdata.html>

To validate the estimators of standard errors we implemented a Monte Carlo simulation to average out the differences emerging in the bootstrap estimates due to sub-sampling variability. For each of the 50 Monte Carlo cycles we sampled 1.000 observations from a different data generating process, simulating one outcome variable, one treatment indicator and 5 covariates. The latter were generated random drawing a triangular Cholesky decomposition matrix and extracting the 5 covariates from a multivariate normal distribution having the corresponding variance–covariance matrix. We then simulated the treatment assignment and the outcome as a function of the 5 covariates and random components independent from each other. For each simulated sample we then computed the *ATT* and the bootstrapped standard errors, while in the case of NN matching we estimated also the analytical standard errors available in the implementations.

**Table 2: Standard errors of the *ATT* estimated under different algorithms and implementations**

|                              |                        | BI   | %PSMatching |
|------------------------------|------------------------|------|-------------|
| <i>ATT</i>                   | Nearest Neighbour      | 4,16 | 4,17        |
|                              | Kernel-Epanechnikov    | 4,11 | 4,11        |
|                              | Kernel-Gaussian        | 4,45 | 4,45        |
|                              | Strata - Pooled        | 4,20 | 4,20        |
|                              | Strata - Satterthwaite | -    | 4,21        |
| Analytical Standard Errors   | Nearest Neighbour      | 1,61 | 1,56        |
|                              | Strata - Pooled        | 1,15 | 1,17        |
|                              | Strata - Satterthwaite | -    | 1,17        |
| Bootstrapped Standard errors | Nearest Neighbour      | 1,81 | 1,81        |
|                              | Kernel- Epanechnikov   | 1,19 | 1,18        |
|                              | Kernel-Gaussian        | 1,14 | 1,14        |
|                              | Strata - Pooled        | 1,18 | 1,20        |
|                              | Strata - Satterthwaite | -    | 1,20        |

Notes: artificial data with sample size = 1.000, five covariates sampled from a multivariate normal distribution with random variance-covariance matrix, share of treated = 20% and treatment effect = 4. The estimates in columns 3 and 4 are the average estimates in 50 Monte Carlo cycles obtained using the implementations of Becker and Ichino (2002, column “IB”) and our own %PSMatching macro. Bootstrapped standard errors are computed with 1.000 cycles.

The estimates of the *ATT* are again aligned across implementations (see Table 2). In the case of NN, a very small difference emerges due to ties, which in IB are handled choosing at random the control that will be matched<sup>6</sup>. Also the bootstrapped standard errors estimates are very similar, the small differences being justified by the random re-sampling used by the bootstrap procedure. The observed differences in the analytical standard errors, on the contrary, are due to the different formula used, with the Becker and Ichino (2002) formula producing estimates slightly higher with respect to what is obtained using the Abadie-Imbens formula.

<sup>6</sup> In %PSMatching one may opt, instead of choosing at random a control, to use as a counterfactual the average outcome among the tied controls.

A second Monte Carlo simulation was designed to check the validity of the confidence intervals implemented in %PSMatching. For each of 1,000 Monte Carlo cycles we sampled 1,000 observations from the same data generating process described above. We then computed analytical and bootstrapped 95% confidence interval for various estimators (including an OLS one as a reference, using the true data generating process as the model specification). We then compared the average CI widths estimated with the empirical ones observed in the Monte Carlo, and checked whether they included the true value of the treatment effect. As expected, in the case of nearest neighbour matching the bootstrapped standard errors are sensibly biased, implying a CI width of 8.83, 6.5% higher than the empirical one. This in turn leads to an empirical coverage rate of 96.7%. Bootstrapped standard errors, on the other side, show a better, although not faultless performance in the case of the kernel matching estimator. The Abadie and Imbens approximate formula shows a better performance in our setting, the empirical coverage rate being 95.6%. In general, the CI widths produced by all analytical estimators have a difference with the empirical one in the order of 1%.

**Table 3: Empirical coverage rates of 95% confidence intervals estimated under different algorithms and implementations**

| Estimation's algorithm                | Att  | CI coverage       |                  | CI width         |                   |                  |
|---------------------------------------|------|-------------------|------------------|------------------|-------------------|------------------|
|                                       |      | <i>analytical</i> | <i>bootstrap</i> | <i>empirical</i> | <i>analytical</i> | <i>bootstrap</i> |
| OLS                                   | 3,95 | 0,950             | -                | 6,00             | 6,13              | -                |
| NN in terms of the propensity score   | 4,01 | 0,956             | 0,967            | 8,29             | 8,36              | 8,83             |
| Kernel matching (Epanechnikov)        | 3,93 | -                 | 0,954            | 6,35             | -                 | 6,61             |
| Stratification method - Pooled        | 4,22 | 0,948             | -                | 6,27             | 6,34              | -                |
| Stratification method - Satterthwaite | 4,22 | 0,949             | -                | 6,27             | 6,34              | -                |

Notes: CI width in case of OLS is computed using analytical CI, in the other case we use bootstrapped CI

Notes on MC parameters: True TE = 4; Sample size = 1.000; %Treated = 30; n° Xs = 5; Bottstrap Cycles = 250; MC Cycles = 1.000; CI = 95%

## 6. Conclusions

In the present paper we introduced %PSMatching, a SAS® macro implementing the most common propensity score based estimators for imputation and causal inference purposes, most of the pre- and post-estimation diagnostics proposed in the literature plus two original ones useful in the fine-tuning of the algorithms parameters.

The algorithms implemented have been validated using as a term of reference the widely used STATA implementations of the estimators, both on real and artificial data. The formulas and algorithms for the computation of standard errors have been further validated by means of a Monte Carlo simulation, showing a good performance for all the estimators implemented.

## 7. References

- Abadie A., and G. Imbens, (2008): “Notes and Comments on the Failure of the Bootstrap for Matching Estimators”, *Econometrica*, Vol. 76(6), 1537–1557.
- Abadie, A., and G. Imbens, (2006a): “Large Sample Properties of Matching Estimators for Average Treatment Effects”, *Econometrica*, vol. 74(1), 235-267.
- Abadie, A., and G. Imbens, (2006b): “On the Failure of the Bootstrap for Matching Estimators”, Working Paper.
- Austin, P.C. (2009): “Using the standardized difference to compare the prevalence of a binary variable between two groups in observational research”, *Communications in Statistics-Simulation and Computation*, 38(6), 1228-1234.
- Becker, S. O. and A. Ichino (2002): “Estimation of Average Treatment Effects Based on Propensity Scores”, *The Stata Journal*, 2(4), 358-377.
- Caliendo M., R. Mahlstedt and O. Mitnik (2014): “Unobservable, but Unimportant? The Influence of Personality Traits (and Other Usually Unobserved Variables) for the Evaluation of Labor Market Policies”, IZA Discussion Paper No. 8337
- Coca-Perraillon, M. (2007). Local and global optimal propensity score matching. In SAS Global Forum 2007. Paper 185-2007
- Cochran, W.G. (1968), “The effectiveness of adjustment by subclassification in removing bias in observational studies”, *Biometrics*, 24(2), 295-313
- Cochran, W.G. and D.B. Rubin (1973), “Controlling Bias in Observational Studies. A review”, *Sankhyā: The Indian Journal of Statistics, Series A*, 35(4), 417-446
- Dehejia, R. H., and S. Wahba (1999): “Causal Effects in Non-Experimental Studies: Reevaluating the Evaluation of Training Programs”, *Journal of the American Statistical Association*, 94(448), 1053-1062.
- Dehejia, R. H., and S. Wahba (2002): “Propensity Score Matching Methods for Nonexperimental Causal Studies”, *The Review of Economics and Statistics*, 84(1), 151 {161.
- Heckman, J. J., Ichimura, H. and Todd, P. (1997) Matching as an econometric evaluation estimator: evidence from evaluating a job training programme. *Rev. Econ. Stud.*, 64, 605–654.
- Heckman, J. J., Ichimura, H. and Todd, P. (1998) Matching as an econometric evaluation estimator. *Rev. Econ. Stud.*, 65, 261–294.
- Heckman, J. J., Ichimura, H., Smith, J. and Todd, P. (1998) Characterizing selection bias using experimental data. *Econometrica*, 66, 1017–1098.
- Heckman, J., R. LaLonde, and J. Smith (1999): “The Economics and Econometrics of Active Labor Market Programs”, in *Handbook of Labor Economics Vol.III*, ed. by O. Ashenfelter, and D. Card, pp. 1865-2097. Elsevier, Amsterdam.

- Holland, P.W. (1986): “Statistics and Causal Inference”, *Journal of the American Statistical Association*, 81(396): 945–960
- Khattree, R., D.N. Naik (1999), *Applied Multivariate Statistics with SAS Software*, Second Edition, Cary, NC: SAS Institute Inc.
- Kosanke, J., and Bergstralh, E. (2004). gmatch: Match 1 or more controls to cases using the GREEDY algorithm, <http://www.mayo.edu/research/departments-divisions/department-health-sciences-research/division-biomedical-statistics-informatics/software/locally-written-sas-macros>, consulted June 30<sup>th</sup>, 2015
- LaLonde, R. (1986): “Evaluating the econometric evaluations of training programs with experimental data”, *American Economic Review*, 76, 604-620
- Lechner, M. (2000): “A Note on the Common Support Problem in Applied Evaluation Studies”, Discussion Paper, SIAW.
- Little, R.J.A. and Rubin, D.B. (1987). *Statistical Analysis with Missing Data*. New York: John Wiley.
- Politis, D.N. and J. Romano, (1994): “Large Sample Confidence Regions Based on Subsamples Under Minimal Assumptions”, *The Annals of Statistics*, vol. 22(4), 2031-2050.
- Rosenbaum, P., and D. Rubin (1983): “The Central Role of the Propensity Score in Observational Studies for Causal Effects”, *Biometrika*, 70, 41-50.
- Rosenbaum, P., and D. Rubin (1985): “Constructing a Control Group Using Multivariate Matched Sampling Methods that Incorporate the Propensity Score”, *The American Statistician*, 39, 33-38.
- Sekhon, J.S. (2011), “Multivariate and Propensity Score Matching Software with Automated Balance Optimization: The Matching Package for R”, *Journal of Statistical Software*, Volume 42, Issue 7
- Sianesi, B. (2004): “An Evaluation of the Active Labour Market Programmes in Sweden”, *The Review of Economics and Statistics*, 86(1), 133-155.
- Yang G., Stemkowski S. and Saunders W. (2007), “A Review of Propensity Score Application in Healthcare Outcome and Epidemiology”, R02 Pharma SUG Conference, Denver, CO.

## 8. Appendix

### *A1 – User manual*

In its most basic usage, the %PSMatching macro asks for four compulsory parameters: a dataset containing the observations; a treatment indicator variable; a target variable; either a variable containing the propensity score, or a list of variables to be included as explicatives in the model with which the propensity score should be estimated:

```
%PSMatching(
```

```

    data, treatment_flag, target_var,
    pscore_var = my_pscore);

%PSMatching(
    data, treatment_flag, target_var,
    pmodel_vars = X1-X10);

```

A first output dataset is generated (adding the suffix “\_matched” to *data*), containing for each treated unit the estimated counterfactual of the target variable, plus the other outputs and diagnostics listed in Appendix A3. As a default, the estimates are computed using nearest neighbour matching in terms of the propensity score with replacement and no calipers. In the second syntax, where the user did not provide the propensity score, the latter is computed using a logistic regression.

Also as a default, the macro interprets the target variable as the outcome in a causal investigation, and generates a second dataset (called “ATT\_ds”), containing the estimate of the *ATT*, the total number of treated units, the number of matched treated units and all the optional outputs requested.

Referring the reader to Appendix A2 for a description of all the parameters and their default values, we present here some syntax examples in typical settings. Further reference may be obtained at runtime, calling `%PSMatching(help)` or `%PSMatching(help parameter_name)`.

In the following example, we request the estimation of the propensity score with the `pmodel_vars` parameter, opting for a probit model with stepwise selection of the covariates. We also check the achieved balance in the observables with the pre-estimation Ichino-Becker test and the post-estimation Sianesi Quality test:

```

%PSMatching(
    data, treatment_flag, target_variable,
    pmodel_vars = X1-X10,
    link = probit, stepwise = Y,
    BH_test = Y, Sianesi_test = Y
);

```

The choice of the matching algorithm is driven by the `match_type` parameter. In the following example, we ask for kernel matching, using the gaussian function and setting the bandwidth to 0.02. We also request bootstrapped confidence intervals with 500 bootstrap cycles:

```

%PSMatching(
    data, treatment_flag, target_variable,
    pmodel_vars = X1-X3,
    match_type = kernel, kernel_function = gaus, bwidth = 0.02,
    bs = Y, bs_cycles = 500
);

```

Here an example for Mahalanobis distance matching within a propensity score caliper, where we opt for the non-reimmission of controls and we specify which are the variables on which the Mahalanobis distance is to be computed:

```
%PSMatching(  
    data, treatment_flag, target_variable,  
    pmodel_vars = X1-X3,  
    match_type = nnmd, mahalanobis_vars = X1-X3 Z1-Z4 pscore,  
    replacement = N,  
    caliper = 0.03  
);
```

Note that, if `mahalanobis_vars` is not specified, `pmodel_vars` will be used, while if the user is passing its own estimate of the `pscore` `mahalanobis_vars` is compulsory. Note also that the name of the propensity score computed by the macro is “`pscore`”, and if one wants to perform Mahalanobis distance matching including the propensity score, as in Rosembaum and Rubin (1985), s/he can add it into the `mahalanobis_vars` list, as in the syntax above.

For imputation purposes, the target variable can be interpreted as the one in which there are missing values. The syntax is the same as in all preceding examples: it is upon to the user to pour the estimates of the counterfactual values into the original variable. Note also that the target variable for the treated will be always missing, so that the *ATT* will end up being missing, too. For computational efficiency, its calculation and the creation of the *ATT\_ds* output dataset may be switched-off setting the parameter “`ATT`” to N.

## *A2 – Parameters (and defaults)*

**data:** name of the dataset containing the treated and controls unit to be matched.

**treatment\_var:** name of the dummy identifying the treated (1 if treated, 0 if control).

**outcome\_var:** outcome variable

**pscore\_var:** variable containing the pre-computed propensity score.

**pmodel\_vars:** covariates to be used in the estimation of the propensity score.

**link** (logit): switches between logit/probit model for the propensity score estimation.

**stepwise** (N): to use the stepwise procedure for the propensity score estimation.

**BH\_test** (N): to perform the Ichino-Becker (2002) test of the Balancing Hypothesis. It is needed in the case of stratification matching: in that case the value of the parameter is forced to “Y”. The output of the test is saved on a dataset named **BH\_test\_output**, reporting for each `pscore` stratum the t-test results on the differences in the variables included in the propensity score model.

**nquantiles** (5): the initial number of propensity score quantiles in which to split the sample for the `BH_test`.

**BH\_test\_conf** (0.05): the confidence level to be used in the t-tests during the `BH_test`.

**ttest\_variance** (satter): to choose how the variance of the t-tests is computed among one of the following: (pooled) it assumes that the pscore and the observables have an equal variance among the treated and the controls; (satter) it drops the constant variance hypothesis and uses the Satterthwaite approximation formula (see for example Khattree and Naik, 1999).

**match\_type** (nnps): to choose the type of matching among one of the following: nnps (Nearest Neighbour in terms of pscore), nnmd (Nearest Neighbour in terms of Mahalanobis Distance), strata (Stratification Matching), kernel (Kernel Matching).

**replacement** (Y): to use or not the replacement option in nearest neighbour matching estimators.

**ties\_handling** (rd): to choose how to handle cases of multiple nearest neighbours among one of the following: rd (random draw a control); avg (compute an average target variable among the tied controls).

**caliper** (0.02): to set a caliper (in term of the propensity score distance) in nearest neighbour matching estimators.

**mahalanobis\_vars**: list of the variables on which the Mahalanobis distance is computed. (If not specified, pmodel\_vars will be used.)

**kernel\_function** (epa): to choose the kernel function to be used in kernel matching among one of the following: gaus (gaussian), epa (Epanechnikov), tri (tricube), biw (biweight) and uni (uniform). See appendix A4 for details.

**bwidth** (0.06): bandwidth parameter for kernel matching.

**ATT** (Y): to compute the *ATT*.

**ATT\_ds** (ATT\_ds): name of the dataset where to save the *ATT* estimates.

**ATT\_alpha** (95): to set the confidence level for the *ATT* confidence intervals' estimation among one of the following: 90, 95, 98, 99.

**CS\_graph** (N): to request a graphic representation of common support.

**CS\_restriction** (N): to impose a common support restriction, by processing only treated units whose pscore is within the interval defined by the minimum and the maximum pscore among the controls, extended by the caliper when specified.

**LBounds** (N); to compute the Lechner Bounds. To compute them the best and worst potential outcome in case of non participation must be specified (parameters Yup and Ylw). The computed bounds are saved into ATT\_ds.

**LBounds\_Yup**: upper bound for the potential outcome in case of non participation (needed to compute the Lechner Bounds).

**LBounds\_Ylow**: lower bound for the potential outcome in case of non participation (needed to compute the Lechner Bounds).

**reduction\_SD** (N): to test the reduction in the Standardized Differences before-after matching (Rosenbaum and Rubin, 1985). The output is saved in a dataset called **standardized\_differences**.

**Sianesi\_test** (N): to compute the Sianesi (2004) pseudo R2 quality test.

**seed** (-1): to select a seed for all pseudo-random number generators.

**matched\_ds** (Y): saves a copy of the input dataset, called “&data.\_matched”, with the individual counterfactual values plus the other outputs and diagnostics listed in Appendix A3.

**bs** (N): to compute the ATT bootstrapped confidence intervals at the ATT\_alpha confidence level. The CIs are saved into ATT\_ds.

**bs\_pct\_style** (Y): use percentiles of the bootstrapped ATT and center them around the estimated ATT.

**bs\_sd\_style** (N): use the bootstrapped standard error of the ATT to build confidence intervals assuming a Normal distribution of the ATT estimator.

**bs\_cycles** (250): the number of bootstrap cycles.

**bs\_pscore** (Y): to bootstrap also the pscore estimation.

**analytical\_std** (N): to compute the ATT analytical standard error using the Abadie and Imbens (2006) formula for one-to-one matching estimators.

**verbose** (Y): to post the narratives about the macro-execution into the log window.

**GC** (Y): to remove from the work library the temporary datasets created by the macro.

**preliminaryChecks** (Y): to perform some checks on the coherence of macro-call parameters.

### ***A3 – Output datasets***

The main output dataset is a copy of the original one with a “\_matched” suffix. The following variables are added to those already present in the original dataset.

For all treated units:

- *counterfactual*: the estimated value of the target variable; missing in case of non-matched units.
- *n\_matches*: the number of controls that have been matched to the unit. In the case of nearest neighbour matching the variable is either equal to 1, or to the number of tied controls.
- *control\_position*, the record number of the control that has been matched to the unit. (Available only in the case of nearest neighbour matching.)
- *best\_distance*: the propensity score or mahalanobis distance at which the nearest neighbour has been found. (Available only in the case of nearest neighbour matching.)

For all control units:

- *weight*: the weight with which the control units enter into the matched sample (in the case of kernel matching, it is the sum of the weights with which it has been matched to different treated units). A missing value indicates that the control has not been

processed, either because of missing values in one or more variables or due to a common support restriction.

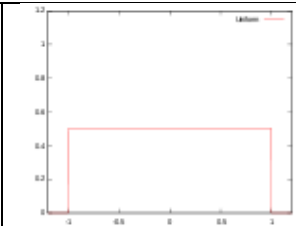
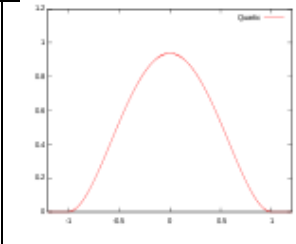
The **ATT\_ds** dataset contains the *ATT* estimate (*att*), the total number of treated units (*treated*), the number of matched treated units (*treated\_matched*) and, depending on the options selected, the following variables:

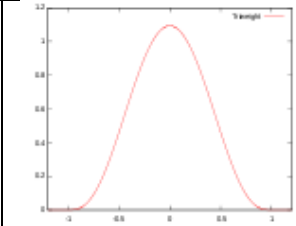
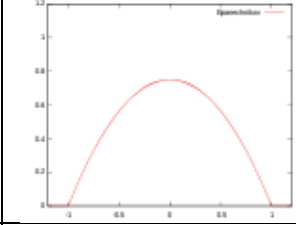
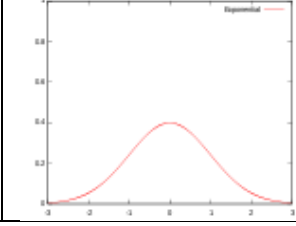
- *std*, *LB\_CIα* and *UB\_CIα*: the analytical estimators of the standard deviation and CI bounds, where  $\alpha$  is the value set with *ATT\_alpha*.
- *avg\_bs\_att*: the average value of the *ATT* computed in the bootstrap cycles.
- *bs\_std*, *LB\_pct\_CIα*, *UB\_pct\_CIα*, *LB\_sd\_CIα*, and *UB\_sd\_CIα*: the bootstrap estimators of the standard deviation and CI bounds, where  $\alpha$  is the value set with *ATT\_alpha*.
- *lechner\_lower\_bound* and *lechner\_upper\_bound*, the Lechner Bounds.

The **BH\_test\_output** is the output dataset produced performing the IB test of the balancing hypothesis. It reports, for each pscore strata (*scoreclass*), the result of the t-test between treated and controls computed on all covariates of the probability model. If the test is passed the flag *mean\_test* is 1 (0 otherwise). For each strata also the t-statistic (*t\_value*), the satterthwaite formula degrees of freedom (*DF*) and the corresponding probability (*probt*) are reported.

The **Standardized\_differences** is the output dataset produced performing the Rosenbaum and Rubin test on the Standardized Differences. It holds, for each variable, the standardized difference pre- and post-matching (*std\_diff\_pre\_matching* and *std\_diff\_post\_matching*). The post-matching differences should be lower then those pre-matching, and as a rule of thumb, they should be lower than 10 in absolute value.

#### ***A4 – Different types of kernel functions in kernel matching***

|                       |                                                       |                                                                                      |
|-----------------------|-------------------------------------------------------|--------------------------------------------------------------------------------------|
| Uniform               | $K(u) = \frac{1}{2} 1_{\{ u  \leq 1\}}$               |  |
| Biweight<br>(Quartic) | $K(u) = \frac{15}{16} (1 - u^2)^2 1_{\{ u  \leq 1\}}$ |  |

|                        |                                                             |                                                                                    |
|------------------------|-------------------------------------------------------------|------------------------------------------------------------------------------------|
| Tricube<br>(Triweight) | $K(u) = \frac{35}{32}(1-u^2)^3 \mathbb{1}_{\{ u  \leq 1\}}$ |  |
| Epanechnikov           | $K(u) = \frac{3}{4}(1-u^2) \mathbb{1}_{\{ u  \leq 1\}}$     |  |
| Gaussian               | $K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2}$          |  |

Note:  $u$  is the difference between the pscore of the treated unit and the pscore of control units, divided by the bandwidth parameter.