---

help for **psmatch2**

Mahalanobis and Propensity score Matching

psmatch2 *depvar* [*indepvars*] [**if** *exp*] [**in** *range*] [, **outcome**(*varlist*) **pscore**(*varname*) **neighbor**(*integer*) **ai**(*integer*) **radius** **caliper**(*real*) **mahalanobis**(*varlist*) **kernel llr** **kerneltype**(*type*) **bwidth**(*real*) **spline nknots**(*integer*) **common trim**(*real*) **noreplacement descending odds index** **logit ties quietly w**(*matrix*) **ate**]

Description

psmatch2 implements full Mahalanobis matching and a variety of propensity score matching methods to adjust for pre-treatment observable differences between a group of treated and a group of untreated. Treatment status is identified by *depvar*==1 for the treated and *depvar*==0 for the untreated observations.

psmatch2 is being continuously improved and developed. Make sure to keep your version up-to-date as follows

```
. ssc install psmatch2, replace
```

By default **psmatch2** calculates approximate standard errors on the treatment effects assuming independent observations, fixed weights, homoskedasticity of the outcome variable within the treated and within the control groups and that the variance of the outcome does not depend on the propensity score:

```
1/N1*Var(Y | DM=1) + Sum(w_i^2; i in DM=0)/(N1)^2*Var(Y | DM=0)
```

where N1 is the number of matched treated, DM=1 denotes the matched treated, DM=0 the matched controls and w_i is the weight given to control i. **psmatch2** stores the estimate of the standard error of the ATT in *r(seatt)* or with more than one outcome variable, in *r(seatt_varname)*.

With nearest neighbor matching on the x's (**mahal()**), then analytical standard errors as in Abadie and Imbens (2006) are calculated when *M>0* is passed using option **ai(*M*)**. Where *M* is the number of neighbors that are used to calculate the conditional variance (formula (14) in Abadie and Imbens, 2006).

psmatch2 stores the estimate of the treatment effect on the treated in *r(att)*, this allows bootstrapping of the standard error of the estimate (although it is unclear whether the bootstrap is valid in this context). This can be done as follows:

. bootstrap r(att) : psmatch2 training age gender, out(wage)

If the average treatment is requested using option *ate* the estimate is returned in *r(ate)*. The average treatment effect on the untreated is then also returned in *r(atu)*. With more than one outcome variable the effects are returned as r(att_*varname*) etc. for each outcome variable and effect.

See the documentation of bs for more details about bootstrapping in Stata.

If you want to be able to replicate your results you should set seed before calling **psmatch2**.

The propensity score - the conditional treatment probability - is either directly provided by the user or estimated by the program on the *indepvars*. Note that the sort order of your data could affect the results when using nearest-neighbor matching on a propensity score estimated with categorical (non-continuous) variables. Or more in general when there are untreated with identical propensity scores.

Matching methods to choose from are one-to-one (nearest neighbour or within caliper; with or without replacement), *k*-nearest neighbors, radius, kernel, local linear regression, 'spline-smoothing' and Mahalanobis matching. The following list presents the syntax for each method.

You can also click here to pop up a dialog or type db psmatch2.

About sample weights

As far as we know it's not really clear in the literature how to accommodate sample weights in the context of matching. If you are aware how to properly account for sampling weights, please let us know. In the meantime, here are some thoughts you might want to take into consideration when asking yourself the following questions:

1) Should I use weights when estimating the score?

A practical solution would be to try both with and without and see which way the X's are better balanced in a given application. To see the balancing when using pweights in the estimation of the score one would however need to pweight the characteristics in the two samples as well (ie compare the pweighted means in the treated and matched controls). (**pstest** does not allow for that).

2) Should I use weights after having performed matching?

When interested in the effect of treatment on the treated, the sampling weights should refer to the treated alone. So the pweigths should be applied to the observed and to the matched outcome (if need be further

restricted to the treated on the common support) for all the treated:

```
. sum outcome if treated==1 [aw=pweight]

. sum _outcome if treated==1 [aw=pweight]
```

Matching within strata

The following code illustrates how to match within exact cells and then calculate the average effect for the whole population.

```
g att = .
egen g = group(groupvars)
levels g, local(gr)
qui foreach j of local gr {
        psmatch2 treatvar varlist if g==`j', out(outvar)
        replace att = r(att) if  g==`j'
}
sum att
```

Detailed Syntax

One-to-one matching:

psmatch2 *depvar* [*indepvars*] [**if** *exp*] [**in** *range*] , [**outcome**(*varlist*) **pscore**(*varname*) **ai**(*integer k*>1) **mahalanobis**(*varlist*) **caliper**(*real*) **noreplacement** **descending** **common** **trim**(*real*) **odds** **index** **logit** **ties** **nowarnings** **quietly** **ate**]

k-**Nearest neighbors matching:**

psmatch2 *depvar* [*indepvars*] [**if** *exp*] [**in** *range*] , [**outcome**(*varlist*) **pscore**(*varname*) **neighbor**(*integer k*>1) **ai**(*integer*) **caliper**(*real*) **common** **trim**(*real*) **odds** **index** **logit** **ties** **nowarnings** **quietly** **ate**]

Radius matching:

psmatch2 *depvar* [*indepvars*] [**if** *exp*] [**in** *range*] , **radius** **caliper**(*real*) [**outcome**(*varlist*) **pscore**(*varname*) **common** **trim**(*real*) **odds** **index** **logit** **quietly** **ate**]

Kernel matching:

psmatch2 *depvar* [*indepvars*] [**if** *exp*] [**in** *range*] **, kernel**
[**outcome**(*varlist*) **kerneltype**(*kernel_type*)
pscore(*varname*) **bwidth**(*real*) **mahalanobis**(*varlist*) **common**
trim(*real*) **odds index logit quietly ate**]

Local linear regression matching:

psmatch2 *depvar* [*indepvars*] [**if** *exp*] [**in** *range*] **, llr**
outcome(*varlist*) [**kerneltype**(*kernel_type*)
pscore(*varname*) **bwidth**(*real*) **mahalanobis**(*varlist*) **common**
trim(*real*) **odds index logit quietly ate**]

Spline matching:

psmatch2 *depvar* [*indepvars*] [**if** *exp*] [**in** *range*] **, spline**
outcome(*varlist*) [**nknots**(*integer*) **pscore**(*varname*)
neighbor(*integer*) **caliper**(*real*) **common trim**(*real*) **odds**
index logit ties nowarnings quietly ate]

Mahalanobis matching:

psmatch2 *depvar* [**if** *exp*] [**in** *range*] **, mahalanobis**(*varlist*)
outcome(*varlist*) [**kernel**(*kernel_type*) **llr bwidth**(*real*)
caliper(*real*) **w**(*matrix*) **ate**]

psmatch2 creates a number of variables for the convenience of the user:

_treated is a variable that equals 0 for control observations and 1 for treatment observations.

_support is an indicator variable with equals 1 if the observation is on the common support and 0 if the observatio is off the support.

_pscore is the estimated propensity score or a copy of the one provided by **pscore().**

_outcome_variable for every treatment observation stores the value of the matched outcome.

_weight. For nearest neighbor matching, it holds the frequency with which the observation is used as a match; with option **ties** and k-nearest neighbors matching it holds the normalized weight; for kernel matching, and llr matching with a weight other than stata's tricube, it stores the overall weight given to the matched observation. When estimating att only _weight = 1 for the treated.

_id In the case of one-to-one and nearest-neighbors matching, a new identifier created for all observations.

_n*k* In the case of one-to-one and nearest-neighbors matching, for every treatment observation, it stores the observation number of the k-th matched control observation. Do not forget to sort by *_id* if you want to use the observation number (id) of for example the 1st nearest neighbor as in

```
sort _id
g x_of_match = x[_n1]
```

_nn In the case of nearest-neighbors matching, for every treatment observation, it stores the number of matched control observations.

Options

outcome(*varlist***)** the outcome variable(s). When evaluating multiple outcomes psmatch2 reduces to the min common number of observations with non-missing values on ALL outcomes, because otherwise the matching weigths will not sum to the right number. If you have multiple outcomes with widely differing missing values you may wish to run psmatch2 separately for each of the outcomes.

ate with this option the average treatment effect (ate) and average treatment effect on the untreated (atu) are reported in addition to the average treatment effect on the treated (att). The estimates are returned in *r(ate)*, *r(atu)* and *r(att)* respectively, see above.

ai(*integer***)** The number of neighbors that are used to calculate the conditional variance (formula (14) in Abadie and Imbens, 2006).

Options: Estimation of the propensity score

pscore(*varname***)** specifies the variable to be used as propensity score.

Alternatively, *indepvars* need to be specified to allow the program to estimate the propensity score on them. In this case:

logit use logit instead of the default probit to estimate the propensity score.

quietly do not print output of propensity score estimation.

odds match on the odds ratio of the propensity score.

index use the latent variable index instead of the probability.

nowarnings do not test for control observations with duplicate propensity score values.

Options: Imposition of common support

common imposes a common support by dropping treatment observations whose pscore is higher than the maximum or less than the minimum pscore of the controls.

trim(*integer***)** imposes common support by dropping # percent of the treatment observations at which the pscore density of the control observations is the lowest.

Options: Choice of matching estimator

neighbor(*integer***)** number of neighbors used to calculate the matched outcome. Defaults to 1. Default matching method is single nearest-neighbour (without caliper).

noreplacement perform 1-to-1 matching without replacement. Nearest neigbor propensity score matching only.

descending perform 1-to-1 matching without replacement in descending order. Nearest neighbor propensity score matching only.

ties not only match nearest neighbor but also other controls with identical (tied) pscores.

radius perform radius matching within the specified radius given by **caliper.**

caliper(*real***)** value for maximum distance of controls. Use to perform nearest neighbor(s) within caliper, radius matching and Mahalanobis 1-to-1 matching.

kernel perform kernel matching.

kerneltype(*kernel_type***)** specifies the type of kernel:

normal the gaussian kernel.

biweight the biweight kernel.

epan the epanechnikov kernel (Default with kernel matching).

uniform the uniform kernel.

tricube the tricube kernel (Default with llr matching).

llr llr use local linear regression matching instead of kernel matching. Option **kerneltype()** allows changing the kernel from the default tricube one.

bwidth(*real***)** the bandwidth for kernel and local linear regression matching. Default bandwidth is 0.06, except when doing local linear regression with the tri-cube kernel when the default bandwidth is 0.8. In this latter case centered subsets of N*bwidth observations are used.

mahalanobis(*varlist***)** perform Mahalanobis-metric matching on *varlist*.

w(*matrix***)** specify alternative weighting matrix. Mahalanobis-metric matching becomes matching on a quadratic metric with the specified weighting matrix.

spline performs 'spline-smoothing matching' by first fitting a natural cubic spline on pscore (or on the result from estimate) to outcome. The matched values are stored in the new variable, _s_outcomevar. (It requires the **spline** programme, which for stata7 needs to be downloaded by typing: net install snp7_1.)

nknots(*integer***)** specifies the number of interior knots for spline smoothing. Default is the fourth root of the number of comparison units.

Examples

```
. psmatch2 training age gender, kernel k(biweight) out(wage)
. psmatch2 training age gender, n(5) logit
. psmatch2 training age gender, out(wage)
. bs "psmatch2 training age gender, out(wage)" "r(att)"
```

Also see

The commands pstest, psgraph.

Thanks for citing psmatch2 as follows

E. Leuven and B. Sianesi. (2003). "PSMATCH2: Stata module to perform full Mahalanobis and propensity score matching, common support graphing, and covariate imbalance testing".
http://ideas.repec.org/c/boc/bocode/s432001.html. This version

INSERT_VERSION_HERE.

where you can check your version as follows:

. which psmatch2

Disclaimer

THIS SOFTWARE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT WILL THE COPYRIGHT HOLDERS OR THEIR EMPLOYERS, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THIS SOFTWARE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Background Reading

Abadie A. and Imbens, G. (2006), "Large sample properties of matching estimators for average treatment effects", *Econometrica 74*(1), 235-267.

Cochran, W. and Rubin, D.B. (1973), "Controlling Bias in Observational Studies", *Sankyha 35*, 417-446.

Dehejia, R.H and Wahba, S. (1999), "Causal Effects in Non-Experimental Studies: Re-Evaluating the Evaluation of Training Programmes", *Journal of the American Statistical Association 94*, 1053-1062.

Heckman, J.J., Ichimura, H. and Todd, P.E. (1997), "Matching As An Econometric Evaluation Estimator: Evidence from Evaluating a Job Training Programme", *Review of Economic Studies 64*, 605-654.

Heckman, J.J., Ichimura, H. and Todd, P.E. (1998), "Matching as an Econometric Evaluation Estimator", *Review of Economic Studies 65*, 261-294.

Heckman, J.J., Ichimura, H., Smith, J.A. and Todd, P. (1998), "Characterising Selection Bias Using Experimental Data", *Econometrica 66*, 5.

Heckman, J.J., LaLonde, R.J., Smith, J.A. (1998), "The Economics and Econometrics of Active Labour Market Programmes", in Ashenfelter, O. and

Card, D. (eds.), *The Handbook of Labour Economics Vol. 3A*.

Imbens, G. (2000), "The Role of Propensity Score in Estimating Dose-Response Functions", *Biometrika 87(3)*, 706-710.

Lechner, M. (2001), Identification and Estimation of Causal Effects of Multiple Treatments under the Conditional Independence Assumption, in: Lechner, M., Pfeiffer, F. (eds), *Econometric Evaluation of Labour Market Policies*, Heidelberg: Physica/Springer, p. 43-58.

Rosenbaum, P.R. and Rubin, D.B. (1983), "The Central Role of the Propensity Score in Observational Studies for Causal Effects", *Biometrika 70*, 1, 41-55.

Rosenbaum, P.R. and Rubin, D.B. (1985), "Constructing a Control Group Using Multivariate Matched Sampling Methods that Incorporate the Propensity Score", *The American Statistician 39(1)*, 33-38.

Rubin, D.B. (1974), "Estimating Causal Effects of Treatments in Randomised and Non-Randomised Studies", *Journal of Educational Psychology 66*, 688-701.

Rubin, D.B. (1980), "Bias Reduction Using Mahalanobis-Metric Matching", *Biometrics 36*, 293-298.

Author

Edwin Leuven, École Nationale de la Statistique et de l'Administration Économique. If you observe any problems mailto:leuven@ensae.fr.