Title

stata.com

teffects psmatch — Propensity-score matching

Description

`teffects psmatch` estimates the average treatment effect (ATE) and average treatment effect on the treated (ATET) from observational data by propensity-score matching (PSM). PSM estimators impute the missing potential outcome for each subject by using an average of the outcomes of similar subjects that receive the other treatment level. Similarity between subjects is based on estimated treatment probabilities, known as propensity scores. The treatment effect is computed by taking the average of the difference between the observed and potential outcomes for each subject. `teffects psmatch` accepts a continuous, binary, count, fractional, or nonnegative outcome.

See [CAUSAL] **teffects intro** or [CAUSAL] **teffects intro advanced** for more information about estimating treatment effects from observational data.

Quick start

ATE of `treat` on `y` estimated by PSM using a logistic model for `treat` on `x` and indicators for levels of categorical variable `a`

```
teffects psmatch (y) (treat x i.a)
```

Same as above, but estimate the ATET

```
teffects psmatch (y) (treat x i.a), atet
```

ATE of `treat` using a heteroskedastic probit model for treatment

```
teffects psmatch (y) (treat x i.a, hetprobit(x i.a))
```

With 4 matches per observation

```
teffects psmatch (y) (treat x i.a), nneighbor(4)
```

Menu

Statistics > Causal inference/treatment effects > Continuous outcomes > Propensity-score matching

Statistics > Causal inference/treatment effects > Binary outcomes > Propensity-score matching

Statistics > Causal inference/treatment effects > Count outcomes > Propensity-score matching

Statistics > Causal inference/treatment effects > Fractional outcomes > Propensity-score matching

Statistics > Causal inference/treatment effects > Nonnegative outcomes > Propensity-score matching

Syntax

```
teffects psmatch (ovar) (tvar tmvarlist [, tmodel]) [if] [in] [weight]
    [, stat options]
```

ovar is a binary, count, continuous, fractional, or nonnegative outcome of interest.

tvar must contain integer values representing the treatment levels.

tmvarlist specifies the variables that predict treatment assignment in the treatment model. Only two treatment levels are allowed.

tmodel	Description
Model	
`logit`	logistic treatment model; the default
`probit`	probit treatment model
`hetprobit(`*varlist*`)`	heteroskedastic probit treatment model

tmodel specifies the model for the treatment variable.

stat	Description
Stat	
`ate`	estimate average treatment effect in population; the default
`atet`	estimate average treatment effect on the treated

options	Description
Model	
nneighbor(#)	specify number of matches per observation; default is nneighbor(1)
SE/Robust	
vce(*vcetype*)	*vcetype* may be vce(robust [, nn(#)]); use robust Abadie–Imbens standard errors with # matches vce(iid); use independent and identically distributed Abadie–Imbens standard errors
Reporting	
level(#)	set confidence level; default is level(95)
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Advanced	
caliper(#)	specify the maximum distance for which two observations are potential neighbors
pstolerance(#)	set tolerance for overlap assumption
osample(*newvar*)	*newvar* identifies observations that violate the overlap assumption
control(# | *label*)	specify the level of *tvar* that is the control
tlevel(# | *label*)	specify the level of *tvar* that is the treatment
generate(*stub*)	generate variables containing the observation numbers of the nearest neighbors
coeflegend	display legend instead of statistics

tmvarlist may contain factor variables; see **[U] 11.4.3 Factor variables**.

by, collect, and statsby are allowed; see **[U] 11.1.10 Prefix commands**.

fweights are allowed; see **[U] 11.1.6 weight**.

coeflegend does not appear in the dialog box.

See **[U] 20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options

Model

nneighbor(#) specifies the number of matches per observation. The default is nneighbor(1). Each individual is matched with at least the specified number of individuals from the other treatment level. nneighbor() must specify an integer greater than or equal to 1 but no larger than the number of observations in the smallest group.

Stat

stat is one of two statistics: ate or atet. ate is the default.

ate specifies that the average treatment effect be estimated.

atet specifies that the average treatment effect on the treated be estimated.

SE/Robust

vce(*vcetype*) specifies the standard errors that are reported. By default, teffects psmatch uses two matches in estimating the robust standard errors.

vce(robust [, nn(#)]) specifies that robust standard errors be reported and that the requested number of matches be used optionally.

vce(iid) specifies that standard errors for independent and identically distributed data be reported.

The standard derivative-based standard-error estimators cannot be used by teffects psmatch, because these matching estimators are not differentiable. The implemented methods were derived by Abadie and Imbens (2006, 2011, 2012); see *Methods and formulas*.

As discussed in Abadie and Imbens (2008), bootstrap estimators do not provide reliable standard errors for the estimator implemented by teffects psmatch.

Reporting

level(#); see [R] **Estimation options**.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

Advanced

caliper(#) specifies the maximum distance at which two observations are a potential match. By default, all observations are potential matches regardless of how dissimilar they are.

In teffects psmatch, the distance is measured by the estimated propensity score. If an observation has no matches, teffects psmatch exits with an error.

pstolerance(#) specifies the tolerance used to check the overlap assumption. The default value is pstolerance(1e-5). teffects will exit with an error if an observation has an estimated propensity score smaller than that specified by pstolerance().

osample(*newvar*) specifies that indicator variable *newvar* be created to identify observations that violate the overlap assumption. Two checks are made to verify the assumption. The first ensures that the propensity scores are greater than pstolerance(#) and less than 1 − pstolerance(#). The second ensures that each observation has at least nneighbor(#) matches in the opposite treatment group within the distance specified by caliper(#).

The vce(robust, nn(#)) option also requires at least # matches in the same treatment group within the distance specified by caliper(#).

The average treatment effect on the treated, option atet, using vce(iid) requires only nneighbor(#) control group matches for the treated group.

control(# | *label*) specifies the level of *tvar* that is the control. The default is the first treatment level. You may specify the numeric level # (a nonnegative integer) or the label associated with the numeric level. control() and tlevel() may not specify the same treatment level.

tlevel(# | *label*) specifies the level of *tvar* that is the treatment for the statistic atet. The default is the second treatment level. You may specify the numeric level # (a nonnegative integer) or the label associated with the numeric level. tlevel() may only be specified with statistic atet. tlevel() and control() may not specify the same treatment level.

generate(*stub*) specifies that the observation numbers of the nearest neighbors be stored in the new variables *stub*1, *stub*2, ... This option is required if you wish to perform postestimation based

on the matching results. The number of variables generated may be more than `nneighbor(#)` because of tied distances. These variables may not already exist.

The following option is available with `teffects psmatch` but is not shown in the dialog box:

`coeflegend`; see [R] **Estimation options**.

Remarks and examples

stata.com

Propensity-score matching uses an average of the outcomes of similar subjects who get the other treatment level to impute the missing potential outcome for each subject. The ATE is computed by taking the average of the difference between the observed and potential outcomes for each subject. `teffects psmatch` determines how near subjects are to each other by using estimated treatment probabilities, known as propensity scores. This type of matching is known as propensity-score matching (PSM).

PSM does not need bias correction, because PSM matches on a single continuous covariate. In contrast, the nearest-neighbor matching estimator implemented in `teffects nnmatch` uses a bias-correction term when matching on more than one continuous covariate. In effect, the PSM estimator parameterizes the bias-correction term in the treatment probability model. See [CAUSAL] **teffects intro** or [CAUSAL] **teffects intro advanced** for more information about this estimator.

We will illustrate the use of `teffects psmatch` by using data from a study of the effect of a mother's smoking status during pregnancy (`mbsmoke`) on infant birthweight (`bweight`) as reported by Cattaneo (2010). This dataset also contains information about each mother's age (`mage`), education level (`medu`), marital status (`mmarried`), whether the first prenatal exam occurred in the first trimester (`prenatal1`), whether this baby was the mother's first birth (`fbaby`), and the father's age (`fage`).

▷ Example 1: Estimating the ATE

We begin by using `teffects psmatch` to estimate the ATE of `mbsmoke` on `bweight`. We use a logistic model (the default) to predict each subject's propensity score, using covariates `mage`, `medu`, `mmarried`, and `fbaby`. Because the performance of PSM hinges upon how well we can predict the propensity scores, we will use factor-variable notation to include both linear and quadratic terms for `mage`, the only continuous variable in our model:

```
. use https://www.stata-press.com/data/r18/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects psmatch (bweight) (mbsmoke mmarried c.mage##c.mage fbaby medu)
Treatment-effects estimation                   Number of obs      =      4,642
Estimator      : propensity-score matching     Matches: requested =          1
Outcome model  : matching                                  min =          1
Treatment model: logit                                     max =         74
```

bweight	Coefficient	AI robust std. err.	z	P>|z|	[95% conf.	interval]
ATE						
mbsmoke (Smoker vs Nonsmoker)	-210.9683	32.021	-6.59	0.000	-273.7284	-148.2083

The average birthweight if all mothers were to smoke would be 211 grams less than the average that would occur if none of the mothers had smoked.

◁

By default, `teffects psmatch` estimates the ATE by matching each subject to a single subject with the opposite treatment whose propensity score is closest. Sometimes, however, we may want to ensure that matching occurs only when the propensity scores of a subject and a match differ by less than a specified amount. To do that, we use the `caliper()` option. If a match within the distance specified in the `caliper()` option cannot be found, `teffects psmatch` exits.

▷ Example 2: Specifying the caliper

Here we reconsider the previous example, first specifying that we only want to consider a pair of observations a match if the absolute difference in the propensity scores is less than 0.03:

```
. teffects psmatch (bweight) (mbsmoke mmarried c.mage##c.mage fbaby medu),
> caliper(0.03)
no propensity-score matches for observation 4504 within caliper 0.03; use option
osample() to identify all observations with deficient matches
r(459);
```

The error arose because there is not a smoking mother whose propensity score is within 0.03 of the propensity score of the nonsmoking mother in observation 4504. If we instead raise the caliper to 0.10, we have matches for all subjects and therefore obtain the same results as in example 1:

```
. teffects psmatch (bweight) (mbsmoke mmarried c.mage##c.mage fbaby medu),
> caliper(0.1)

Treatment-effects estimation                   Number of obs      =      4,642
Estimator      : propensity-score matching     Matches: requested =          1
Outcome model  : matching                                 min =          1
Treatment model: logit                                    max =         74
```

bweight	Coefficient	AI robust std. err.	z	P>|z|	[95% conf.	interval]
ATE						
mbsmoke (Smoker vs Nonsmoker)	-210.9683	32.021	-6.59	0.000	-273.7284	-148.2083

◁

❑ Technical note

Example 2 highlights that estimating the ATE requires finding matches for both the treated and control subjects. In contrast, estimating the ATET only requires finding matches for the treated subjects. Because subject 4504 is a control subject, we can estimate the ATET using `caliper(0.03)`. We must also specify `vce(iid)` because the default robust standard errors for the estimated ATET require viable matches for both treated subjects and control subjects. (This requirement comes from the nonparametric method derived by Abadie and Imbens [2012].)

```
. teffects psmatch (bweight) (mbsmoke mmarried c.mage##c.mage fbaby medu),
> atet vce(iid) caliper(0.03)
Treatment-effects estimation                   Number of obs      =      4,642
Estimator      : propensity-score matching     Matches: requested =          1
Outcome model  : matching                                  min =          1
Treatment model: logit                                     max =         74
```

bweight	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
ATET						
mbsmoke (Smoker vs Nonsmoker)	-236.7848	26.11698	-9.07	0.000	-287.9731	-185.5964

❑

In the previous examples, each subject was matched to at least one other subject, which is the default behavior for `teffects psmatch`. However, we can request that `teffects psmatch` match each subject to multiple subjects with the opposite treatment level by specifying the `nneighbor()` option. Matching on more distant neighbors can reduce the variance of the estimator at a cost of an increase in bias.

▷ Example 3

Now we request that `teffects psmatch` match a mother to four mothers in the opposite treatment group:

```
. teffects psmatch (bweight) (mbsmoke mmarried c.mage##c.mage fbaby medu),
> nneighbor(4)
Treatment-effects estimation                   Number of obs      =      4,642
Estimator      : propensity-score matching     Matches: requested =          4
Outcome model  : matching                                  min =          4
Treatment model: logit                                     max =         74
```

bweight	Coefficient	AI robust std. err.	z	P>|z|	[95% conf.	interval]
ATE						
mbsmoke (Smoker vs Nonsmoker)	-224.006	29.88627	-7.50	0.000	-282.582	-165.43

These results are similar to those reported in example 1.

◁

Video example

Treatment effects in Stata: Propensity-score matching

Stored results

teffects psmatch stores the following in e():

Scalars

e(N)	number of observations
e(n*j*)	number of observations for treatment level *j*
e(k_levels)	number of levels in treatment variable
e(caliper)	maximum distance between matches
e(treated)	level of treatment variable defined as treated
e(control)	level of treatment variable defined as control
e(k_nneighbor)	requested number of matches
e(k_nnmin)	minimum number of matches
e(k_nnmax)	maximum number of matches
e(k_robust)	matches for robust VCE

Macros

e(cmd)	teffects
e(cmdline)	command as typed
e(depvar)	name of outcome variable
e(tvar)	name of treatment variable
e(subcmd)	psmatch
e(tmodel)	logit, probit, or hetprobit
e(stat)	statistic estimated, ate or atet
e(wtype)	weight type
e(wexp)	weight expression
e(title)	title in estimation output
e(tlevels)	levels of treatment variable
e(psvarlist)	variables in propensity-score model
e(hvarlist)	variables for variance, only if hetprobit
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
e(datasignature)	the checksum
e(datasignaturevars)	variables used in calculation of checksum
e(properties)	b V
e(estat_cmd)	program used to implement estat
e(predict)	program used to implement predict
e(marginsnotok)	predictions disallowed by margins
e(asbalanced)	factor variables fvset as asbalanced
e(asobserved)	factor variables fvset as asobserved

Matrices

e(b)	coefficient vector
e(V)	variance–covariance matrix of the estimators
e(bps)	coefficient vector from propensity-score model
e(Vps)	variance–covariance matrix of the estimators from propensity-score model

Functions

e(sample)	marks estimation sample

In addition to the above, the following is stored in r():

Matrices

r(table)	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in r() are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

The methods and formulas used by teffects psmatch are documented in the *Methods and formulas* of [CAUSAL] **teffects nnmatch**.

References

Abadie, A., and G. W. Imbens. 2006. Large sample properties of matching estimators for average treatment effects. *Econometrica* 74: 235–267. https://doi.org/10.1111/j.1468-0262.2006.00655.x.

——. 2008. On the failure of the bootstrap for matching estimators. *Econometrica* 76: 1537–1557. https://doi.org/10.3982/ECTA6474.

——. 2011. Bias-corrected matching estimators for average treatment effects. *Journal of Business and Economic Statistics* 29: 1–11. https://doi.org/10.1198/jbes.2009.07333.

——. 2012. Matching on the estimated propensity score. Harvard University and National Bureau of Economic Research. http://www.nber.org/papers/w15301.

Alejo, J., A. F. Galvao, and G. Montes-Rojas. 2020. A practical generalized propensity-score estimator for quantile continuous treatment effects. *Stata Journal* 20: 276–296.

Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. https://doi.org/10.1016/j.jeconom.2009.09.023.

Díaz, J. D., I. Gutiérrez, and J. Rivera. 2021. Implementing blopmatching in Stata. *Stata Journal* 21: 180–194.

Huber, C. 2015. Introduction to treatment effects in Stata: Part 2. *The Stata Blog: Not Elsewhere Classified.* http://blog.stata.com/2015/08/24/introduction-to-treatment-effects-in-stata-part-2/.

Tazare, J., L. Smeeth, S. J. W. Evans, I. J. Douglas, and E. J. Williamson. 2023. hdps: A suite of commands for applying high-dimensional propensity-score approaches. *Stata Journal* 23: 683–708.

Also see

[CAUSAL] **teffects postestimation** — Postestimation tools for teffects

[CAUSAL] **teffects** — Treatment-effects estimation for observational data

[CAUSAL] **teffects nnmatch** — Nearest-neighbor matching

[U] 20 Estimation and postestimation commands