Title

teoverlap — Overlap plots

Description Quick start Menu Syntax
Options Remarks and examples Stored results References
Also see

Description

One of the assumptions required to use the `teffects` and `stteffects` estimators is the overlap assumption, which states that each individual has a positive probability of receiving each treatment level. `teoverlap`, a postestimation command, plots the estimated densities of the probability of getting each treatment level. These plots can be used to check whether the overlap assumption is violated.

Quick start

Visually check whether the overlap assumption is violated

```
teoverlap
```

As above, but use the Epanechnikov kernel function

```
teoverlap, kernel(epanechnikov)
```

Plot probability of getting treatment 3 for subjects receiving treatments 2 or 3 of a multivalued treatment

```
teoverlap, ptlevel(3) tlevels(2 3)
```

As above, and change legend labels to "Treated 2" and "Treated 3"

```
teoverlap, ptlevel(3) tlevels(2 3)                    ///
        legend(label(1 "Treated 2") label(2 "Treated 3"))
```

Menu

Statistics > Treatment effects > Overlap plots

Syntax

```
teoverlap [ , treat_options kden_options ]
```

treat_options	Description
Main	
`ptlevel(`*treat_level*`)`	calculate predicted probabilities for treatment level *treat_level*; by default, `ptlevel()` corresponds to the first treatment level
`tlevels(`*treatments*`)`	specify conditioning treatment levels; default is all treatment levels
`nolabel`	use treatment level values and not value labels in legend and axis titles

kden_options	Description
Main	
`kernel(`*kernel*`)`	specify kernel function; default is `kernel(triangle)`
`n(#)`	estimate densities using # points; default is `e(N)`, the number of observations in the estimation sample
`bwidth(#)`	half-width of kernel
`at(`*var_x*`)`	estimate densities using the values specified by *var_x*
Kernel plots	
`line#opts(`*cline_options*`)`	affect rendition of density for conditioning treatment #
Add plots	
`addplot(`*plot*`)`	add other plots to the generated graph
Y axis, X axis, Titles, Legend, Overall	
twoway_options	any options other than `by()` documented in [G-3] ***twoway_options***

kernel	Description
`triangle`	triangle kernel function; the default
`epanechnikov`	Epanechnikov kernel function
`epan2`	alternative Epanechnikov kernel function
`biweight`	biweight kernel function
`cosine`	cosine trace kernel function
`gaussian`	Gaussian kernel function
`parzen`	Parzen kernel function
`rectangle`	rectangle kernel function

`collect` is allowed; see [U] **11.1.10 Prefix commands**.

Options

Main

`ptlevel(`*treat_level*`)` specifies that predicted probabilities be calculated for treatment level *treat_level*. The default is `ptlevel(`*first*`)`, where *first* is the first treatment level.

`tlevels(`*treatments*`)` specifies the observations for which to obtain predicted probabilities. By default, all treatment levels are used. Specify *treatments* as a space-delimited list.

For instance,

```
. teoverlap, ptlevel(1) tlevels(1 2)
```

says to predict the probability of getting treatment level 1 for those subjects who actually obtained treatment levels 1 or 2.

`nolabel` specifies that treatment level values and not value labels be used in legend and axis titles.

`kernel(`*kernel*`)` specifies the kernel function for use in calculating the kernel density estimates. The default kernel is the triangle kernel (`triangle`).

`n(#)` specifies the number of points at which the density estimate is to be evaluated. The default is `e(N)`, the estimation sample size.

`bwidth(#)` specifies the half-width of the kernel, the width of the density window around each point. If `bwidth()` is not specified, the "optimal" width is calculated and used; see [R] **kdensity**. The optimal width is the width that would minimize the mean integrated squared error if the data were Gaussian and a Gaussian kernel were used, so it is not optimal in any global sense. In fact, for multimodal and highly skewed densities, this width is usually too wide and oversmooths the density (Silverman 1986).

`at(`*var_x*`)` specifies a variable that contains the values at which the density should be estimated. This option allows you to more easily obtain density estimates for different variables or different subsamples of a variable and then overlay the estimated densities for comparison.

Kernel plots

`line#opts(`*cline_options*`)` affect the rendition of the plotted kernel density estimates. See [G-3] ***cline_options***.

Add plots

`addplot(`*plot*`)` provides a way to add other plots to the generated graph. See [G-3] ***addplot_option***.

Y axis, X axis, Titles, Legend, Overall

twoway_options are any of the options documented in [G-3] ***twoway_options***, excluding `by()`. These include options for titling the graph (see [G-3] ***title_options***) and for saving the graph to disk (see [G-3] ***saving_option***).

Remarks and examples

stata.com

`teoverlap` plots the estimated densities of the probability of getting each treatment level after `teffects`.

These plots can be used to check whether the overlap assumption is violated. The overlap assumption is satisfied when there is a chance of seeing observations in both the control and the treatment groups at each combination of covariate values; see [TE] **teffects intro** or [TE] **teffects intro advanced**.

The overlap assumption is required by the estimators implemented in `teffects`. Intuitively, when the overlap assumption is violated, we cannot predict, or otherwise account for, the unobserved outcomes for some individuals.

There is evidence that the overlap assumption is violated when an estimated density has too much mass around 0 or 1; see Busso, DiNardo, and McCrary (2014). An implication of this point is that when the overlap assumption is violated, the estimated densities will have relatively little mass in the regions in which they overlap.

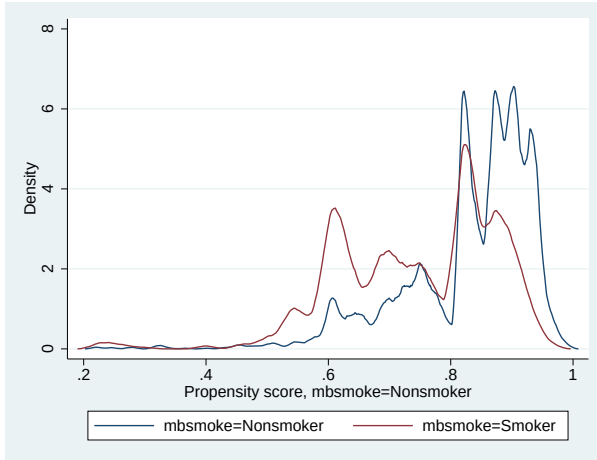
Example 1: Assumption not violated

Continuing with example 1 of [TE] **teffects ipw**, we estimate the average treatment effect of smoking on birthweight and then draw the overlap plot:

```
. use https://www.stata-press.com/data/r17/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects ipw (bweight) (mbsmoke mmarried c.mage##c.mage fbaby medu, probit)
Iteration 0:   EE criterion =  4.621e-21
Iteration 1:   EE criterion =  7.358e-26
Treatment-effects estimation                   Number of obs     =      4,642
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: probit
```

bweight	Coefficient	Robust std. err.	z	P>|z|	[95% conf.	interval]
ATE						
mbsmoke (Smoker vs Nonsmoker)	-230.6886	25.81524	-8.94	0.000	-281.2856	-180.0917
POmean						
mbsmoke Nonsmoker	3403.463	9.571369	355.59	0.000	3384.703	3422.222

```
. teoverlap
```



The graph displays the estimated density of the predicted probabilities that a nonsmoking mother is a nonsmoker and the estimated density of the predicted probabilities that a smoking mother is a nonsmoker.

Neither plot indicates too much probability mass near 0 or 1, and the two estimated densities have most of their respective masses in regions in which they overlap each other. Thus there is no evidence that the overlap assumption is violated.

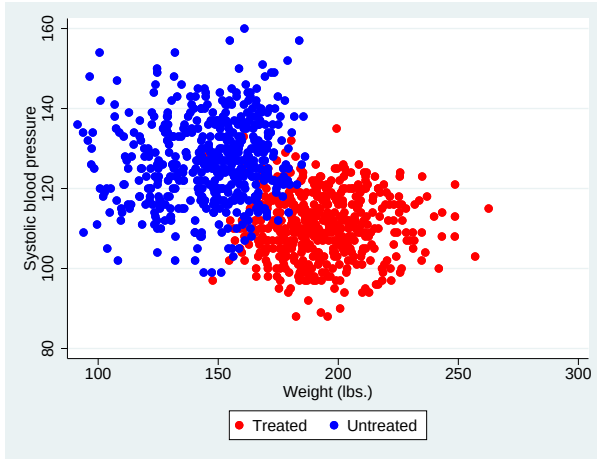
◁

▷ Example 2: Assumption violated

This example produces an overlap plot that indicates a failure of the overlap assumption. We will use simulated data, so we know that the assumption is not true.

In our simulated dataset, some of the 1,000 adult males were given drug XY1 for high blood pressure and others were not. A scatterplot of systolic blood pressure (`systolic`) and weight (`weight`) reveals that heavier men were given the treatment. (The scatterplots corresponding to the treatment group are colored red, while the scatterplots corresponding to the control group are colored blue.)

```
. use https://www.stata-press.com/data/r17/systolic2
(Systolic blood pressure)
. twoway (scatter systolic weight if xy1==1, mcolor(red))
>        (scatter systolic weight if xy1==0, mcolor(blue)),
>         legend(label(1 "Treated") label(2 "Untreated"))
```



There are no observations in the treated group for small weights, and there are no observations in the control group for large weights. There is clear evidence that the overlap assumption is violated.

Drawing an overlaid scatterplot is a straightforward way to check the overlap assumption in this example because there is only one covariate. This method is not available when there is more than one covariate. The predicted probability is a one-dimensional measure that captures the relevant multivariate information.

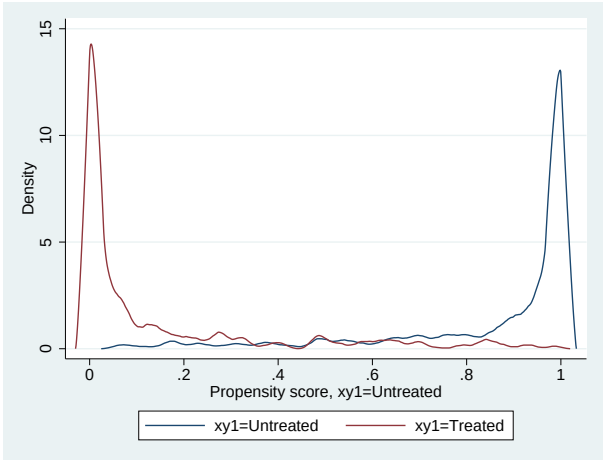
Below we estimate the parameters needed to calculate the predicted probabilities. The `pstolerance(1e-8)` option is specified to ensure that estimation is performed as long as the predicted probabilities are at least as large as 1e–8.

```
. teffects ipw (systolic) (xy1 weight), pstolerance(1e-8)

Iteration 0:   EE criterion =  9.523e-18
Iteration 1:   EE criterion =  3.489e-28

Treatment-effects estimation                    Number of obs     =      1,000
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: logit
```

systolic	Coefficient	Robust std. err.	z	P>|z|	[95% conf.	interval]
ATE						
xy1 (Treated vs Untreated)	-16.23679	2.191703	-7.41	0.000	-20.53245	-11.94114
POmean						
xy1 Untreated	127.9094	.7004533	182.61	0.000	126.5365	129.2822

Now we can obtain the overlap plot.

```
. teoverlap
```



The estimated density of the predicted probabilities that a treated individual is not assigned to XY1 treatment has most of its mass near 0. The estimated density of the predicted probabilities that an untreated individual is not assigned to XY1 treatment has most of its mass near 1. Note that the two have very little mass in the region in which they overlap. There is clear evidence that the overlap assumption is violated.

◁

Stored results

`teoverlap` stores the following in `r()`:

Scalars

`r(bwidth`*j*`)`	kernel bandwidth for treatment level *j*
`r(n`*j*`)`	number of points at which the estimate was evaluated for treatment level *j*
`r(scale`*j*`)`	density bin width for treatment level *j*

Macros

`r(kernel)`	name of kernel

References

Busso, M., J. DiNardo, and J. McCrary. 2014. New evidence on the finite sample properties of propensity score reweighting and matching estimators. *Review of Economics and Statistics* 96: 885–897. https://doi.org/10.1162/REST_a_00431.

Silverman, B. W. 1986. *Density Estimation for Statistics and Data Analysis*. London: Chapman & Hall.

Also see

[TE] **stteffects** — Treatment-effects estimation for observational survival-time data

[TE] **stteffects ipw** — Survival-time inverse-probability weighting

[TE] **stteffects ipwra** — Survival-time inverse-probability-weighted regression adjustment

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

[TE] **teffects aipw** — Augmented inverse-probability weighting

[TE] **teffects ipw** — Inverse-probability weighting

[TE] **teffects ipwra** — Inverse-probability-weighted regression adjustment

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

[TE] **teffects psmatch** — Propensity-score matching

[TE] **teffects ra** — Regression adjustment

[TE] **telasso** — Treatment-effects estimation using lasso