lateoverlap — Overlap plots+

+This command is part of StataNow.

Description

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

Quick start

Visually check whether the overlap assumption is violated

```
lateoverlap
```

Same as above, but use the Epanechnikov kernel function

```
lateoverlap, kernel(epanechnikov)
```

Menu

Statistics > Postestimation

Syntax

`lateoverlap` [, *treat_options kden_options*]

treat_options	Description
Main	
`ptlevel(`*treat_level*`)`	calculate predicted probabilities for treatment-assignment level *treat_level*; default is the first treatment-assignment level
`tlevels(`*treatments*`)`	specify conditioning treatment-assignment levels; default is all treatment-assignment levels
`nolabel`	use treatment-assignment 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-assignment level *treat_level*. The default is `ptlevel(`*first*`)`, where *first* is the first treatment-assignment level.

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

For instance,

```
. lateoverlap, ptlevel(1) tlevels(0 1)
```

says to predict the probability of getting treatment-assignment level 1 for those subjects who actually obtained treatment-assignment levels 0 or 1.

`nolabel` specifies that treatment-assignment 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

`lateoverlap` plots the estimated densities of the probability of getting each treatment-assignment level after `lateffects`.

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-assignment groups at each combination of covariate values.

The overlap assumption is required by the estimators implemented in `lateffects`. Intuitively, when the overlap assumption is violated, we cannot predict, or otherwise account for, the unobserved outcomes for some units, specifically, for units assigned to treatment with characteristics (that is, covariate values) that are absent in the control group and for units assigned to control with characteristics that cannot be found in the treated group.

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.

For an illustration, see example 2 of [CAUSAL] **lateffects**.

Stored results

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

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

Also see

[CAUSAL] **lateffects** — Local average treatment-effect estimation^{+}

Stata, Stata Press, Mata, NetCourse, and NetCourseNow are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow is a trademark of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2025 StataCorp LLC, College Station, TX, USA. All rights reserved.

For suggested citations, see the FAQ on citing Stata documentation.