graph twoway lpolyci — Local polynomial smooth plots with CIs

Description

graph twoway lpolyci plots a local polynomial smooth of yvar on xvar by using graph twoway line (see [G-2] graph twoway line), along with a confidence interval by using graph twoway rarea (see [G-2] graph twoway rarea).

Quick start

Local polynomial smooth plot of y versus x with 95% confidence interval and local mean smoothing
   twoway lpolyci y x

As above, but with a polynomial of degree 3
   twoway lpolyci y x, degree(3)

Overlay scatterplot of the observed data on polynomial smooth plot
   twoway lpolyci y x || scatter y x

As above, but with a 99% confidence interval
   twoway lpolyci y x, level(99) || scatter y x

Specify the Gaussian kernel function
   twoway lpolyci y x, kernel(gaussian) || scatter y x

Specify the half-width of the kernel as 85
   twoway lpolyci y x, bwidth(85) || scatter y x

With a separate graph area for each level of catvar
   twoway lpolyci y x || scatter y x, by(catvar)

Add “My Title” as the title of the graph
   twoway lpolyci y x || scatter y x, title("My Title")

Draw the confidence limits as a pair of lines rather than as an area
   twoway lpolyci y x, ciplot(rline) || scatter y x

Menu

Graphics > Twoway graph (scatter, line, etc.)
**Syntax**

```
twoway lpoly ci yvar xvar [if] [in] [weight] [, options]
```

<table>
<thead>
<tr>
<th>options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>kernel(kernel)</code></td>
<td>kernel function; default is <code>kernel(epanechnikov)</code></td>
</tr>
<tr>
<td><code>bwidth(#)</code></td>
<td>kernel bandwidth</td>
</tr>
<tr>
<td><code>degree(#)</code></td>
<td>degree of the polynomial smooth; default is <code>degree(0)</code></td>
</tr>
<tr>
<td><code>n(#)</code></td>
<td>obtain the smooth at # points; default is min(N, 50)</td>
</tr>
<tr>
<td><code>level(#)</code></td>
<td>set confidence level; default is <code>level(95)</code></td>
</tr>
<tr>
<td><code>pwidth(#)</code></td>
<td>pilot bandwidth for standard error calculation</td>
</tr>
<tr>
<td><code>var(#)</code></td>
<td>estimate of the constant conditional variance</td>
</tr>
<tr>
<td><code>nofit</code></td>
<td>do not plot the smooth</td>
</tr>
<tr>
<td><code>fitplot(plottype)</code></td>
<td>how to plot the smooth; default is <code>fitplot(line)</code></td>
</tr>
<tr>
<td><code>ciplot(plottype)</code></td>
<td>how to plot CIs; default is <code>ciplot(rarea)</code></td>
</tr>
<tr>
<td><code>fcl</code></td>
<td>change look of the smoothed line</td>
</tr>
<tr>
<td><code>fitarea</code></td>
<td>change look of CI</td>
</tr>
<tr>
<td><code>axis_choice_options</code></td>
<td>associate plot with alternative axis</td>
</tr>
<tr>
<td><code>twoway_options</code></td>
<td>titles, legends, axes, added lines and text, by, regions, name, aspect ratio, etc.</td>
</tr>
<tr>
<td><code>kernel</code></td>
<td>Description</td>
</tr>
<tr>
<td><code>epanechnikov</code></td>
<td>Epanechnikov kernel function; the default</td>
</tr>
<tr>
<td><code>epan2</code></td>
<td>alternative Epanechnikov kernel function</td>
</tr>
<tr>
<td><code>biweight</code></td>
<td>biweight kernel function</td>
</tr>
<tr>
<td><code>cosine</code></td>
<td>cosine trace kernel function</td>
</tr>
<tr>
<td><code>gaussian</code></td>
<td>Gaussian kernel function</td>
</tr>
<tr>
<td><code>parzen</code></td>
<td>Parzen kernel function</td>
</tr>
<tr>
<td><code>rectangle</code></td>
<td>rectangle kernel function</td>
</tr>
<tr>
<td><code>triangle</code></td>
<td>triangle kernel function</td>
</tr>
</tbody>
</table>

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

**Options**

`kernel(kernel)` specifies the kernel function for use in calculating the weighted local polynomial estimate. The default is `kernel(epanechnikov)`. See [R] kdensity for more information on this option.

`bwidth(#)` specifies the half-width of the kernel, the width of the smoothing window around each point. If `bwidth()` is not specified, a rule-of-thumb bandwidth estimator is calculated and used; see [R] ipoly.

`degree(#)` specifies the degree of the polynomial to be used in the smoothing. The default is `degree(0)`, meaning local mean smoothing.
n(#) specifies the number of points at which the smooth is to be evaluated. The default is $min(N, 50)$, where $N$ is the number of observations.

level(#) specifies the confidence level, as a percentage, for confidence intervals. The default is level(95) or as set by set level; see [U] 20.8 Specifying the width of confidence intervals.

pwidth(#) specifies the pilot bandwidth to be used for standard error computations. The default is chosen to be 1.5 times the value of the rule-of-thumb bandwidth selector.

var(#) specifies an estimate of a constant conditional variance required for standard error computation. By default, the conditional variance at each smoothing point is estimated by the normalized weighted residual sum of squares obtained from locally fitting a polynomial of order $p + 2$, where $p$ is the degree specified in degree().

nofit prevents the smooth from being plotted.

fitplot(plottype) specifies how the prediction is to be plotted. The default is fitplot(line), meaning that the smooth will be plotted by graph twoway line. See [G-2] graph twoway for a list of plottype choices. You may choose any that expects one $y$ and one $x$ variable. fitplot() is seldom used.

ciplot(plottype) specifies how the confidence interval is to be plotted. The default is ciplot(rarea), meaning that the confidence bounds will be plotted by graph twoway rarea. A reasonable alternative is ciplot(rline), which will substitute lines around the smooth for shading. See [G-2] graph twoway for a list of plottype choices. You may choose any that expects two $y$ variables and one $x$ variable.

Fcline_options specify how the lpoly line is rendered and its appearance; see [G-3] fcline_options. Fitarea_options specify how the confidence interval is rendered; see [G-3] fitarea_options. If you specify ciplot(), you should specify whatever is appropriate instead of using fitarea_options.

Axis_choice_options associate the plot with a particular $y$ or $x$ axis on the graph; see [G-3] axis_choice_options.

Twoway_options are a set of common options supported by all twoway graphs. These options allow you to title graphs, name graphs, control axes and legends, add lines and text, set aspect ratios, create graphs over by() groups, and change some advanced settings. See [G-3] twoway_options.

**Remarks and examples**

`graph twoway lpolyci yvar xvar` uses the `lpoly` command—see [R] lpoly—to obtain a local polynomial smooth of `yvar` on `xvar` and confidence intervals and uses `graph twoway line` and `graph twoway rarea` to plot results.

Remarks are presented under the following headings:

Typical use
Use with by()
Typical use

`graph twoway lpolyci` can be used to overlay the confidence bands obtained from different local polynomial smooths. For example, for local mean and local cubic polynomial smooths:

```
use https://www.stata-press.com/data/r16/auto
(1978 Automobile Data)
twoway lpolyci weight length, nofit ||
lpolyci weight length, degree(3) nofit
ciplot(rline) pstyle(ci2) ||
scatter weight length, msymbol(o)
```

The plotted area corresponds to the confidence bands for the local mean smooth and lines correspond to confidence intervals for the local cubic smooth.

When you overlay graphs, you nearly always need to respecify the axis titles by using the `axis_title_options ytitle()` and `xtitle()`; see [G-3] `axis_title_options`. 
Use with `by()`

graph twoway lpolyci may be used with `by()`:

```
. use https://www.stata-press.com/data/r16/auto, clear
   (1978 Automobile Data)
. twoway lpolyci weight length ||
   scatter weight length, msymbol(o) ||
   , by(foreign)
```

Also see

[R] lpoly — Kernel-weighted local polynomial smoothing

[G-2] graph twoway lpoly — Local polynomial smooth plots