graph twoway lfitci — Two-way linear prediction plots with CIs

Description

`twoway lfitci` calculates the prediction for *yvar* from a linear regression of *yvar* on *xvar* and plots the resulting line, along with a confidence interval.

Quick start

A linear fit prediction plot for `y` on `x` with a 95% confidence interval of the prediction

```
twoway lfitci y x
```

Same as above, but with confidence intervals for an individual forecast

```
twoway lfitci y x, stdf
```

Specify a 99% confidence interval

```
twoway lfitci y x, level(99)
```

Overlay a scatterplot showing observed data

```
twoway lfitci y x || scatter y x
```

Display confidence limits using a pair of lines

```
twoway lfitci y x, ciplot(rline) || scatter y x
```

Same as above, but with confidence limits displayed using dashed lines

```
twoway lfitci y x, ciplot(rline) blpattern(dash) || scatter y x
```

Display confidence limits using a range plot with spikes

```
twoway lfitci y x, ciplot(rspike) || scatter y x
```

A separate graph area for each level of `catvar`

```
twoway lfitci y x || scatter y x, by(catvar)
```

Menu

Graphics > Two-way graph (scatter, line, etc.)

Syntax

twoway lfitci *yvar xvar* [*if*] [*in*] [*weight*] [, *options*]

options	Description
stdp	CIs from SE of prediction; the default
stdf	CIs from SE of forecast
stdr	CIs from SE of residual; seldom specified
level(#)	set confidence level; default is level(95)
range(# #)	range over which predictions are calculated
n(#)	number of prediction points
atobs	calculate predictions at *xvar*
estopts(*regress_options*)	options for regress
predopts(*predict_options*)	options for predict
nofit	do not plot the prediction
fitplot(*plottype*)	how to plot fit; default is fitplot(line)
ciplot(*plottype*)	how to plot CIs; default is ciplot(rarea)
fcline_options	change look of predicted line
fitarea_options	change look of CI
axis_choice_options	associate plot with alternative axis
twoway_options	titles, legends, axes, added lines and text, by, regions, name, aspect ratio, etc.

Options range(), estopts(), predopts(), n(), and level() are *rightmost*; atobs, nofit, fitplot(), ciplot(), stdp, stdf, and stdr are *unique*; see [G-4] **Concept: repeated options**.

yvar and *xvar* may contain time-series operators; see **[U] 11.4.4 Time-series varlists**.

aweights, fweights, and pweights are allowed. Weights, if specified, affect estimation but not how the weighted results are plotted. See **[U] 11.1.6 weight**.

Options

stdp, stdf, and stdr determine the basis for the confidence interval. stdp is the default.

stdp specifies that the confidence interval be the confidence interval of the mean.

stdf specifies that the confidence interval be the confidence interval for an individual forecast, which includes both the uncertainty of the mean prediction and the residual.

stdr specifies that the confidence interval be based only on the standard error of the residual.

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

range(# #) specifies the x range over which predictions are calculated. The default is range(. .), meaning the minimum and maximum values of *xvar*. range(0 10) would make the range 0 to 10, range(. 10) would make the range the minimum to 10, and range(0 .) would make the range 0 to the maximum.

n(#) specifies the number of points at which the predictions and the CI over range() are to be calculated. The default is n(100).

`atobs` is an alternative to `n()` and specifies that the predictions be calculated at the *xvar* values. `atobs` is the default if `predopts()` is specified and any statistic other than the `xb` is requested.

`estopts(`*regress_options*`)` specifies options to be passed along to `regress` to estimate the linear regression from which the line will be predicted; see [R] **regress**. If this option is specified, also commonly specified is `estopts(nocons)`.

`predopts(`*predict_options*`)` specifies options to be passed along to `predict` to obtain the predictions after estimation by `regress`; see [R] **regress postestimation**.

`nofit` prevents the prediction from being plotted.

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

`ciplot(`*plottype*`)` specifies how the confidence interval is to be plotted. The default is `ciplot(rarea)`, meaning that the prediction will be plotted by `graph twoway rarea`.

A reasonable alternative is `ciplot(rline)`, which will substitute lines around the prediction for shading. See [G-2] **graph twoway** for a list of *plottype* choices. You may choose any that expect two y variables and one x variable.

fcline_options specify how the prediction line is rendered; see [G-3] ***fcline_options***. If you specify `fitplot()`, then rather than using *fcline_options*, you should select options that affect the specified *plottype* from the options in `scatter`; see [G-2] **graph twoway scatter**.

fitarea_options specify how the confidence interval is rendered; see [G-3] ***fitarea_options***. If you specify `ciplot()`, then rather than using *fitarea_options*, you should specify whatever is appropriate.

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***.

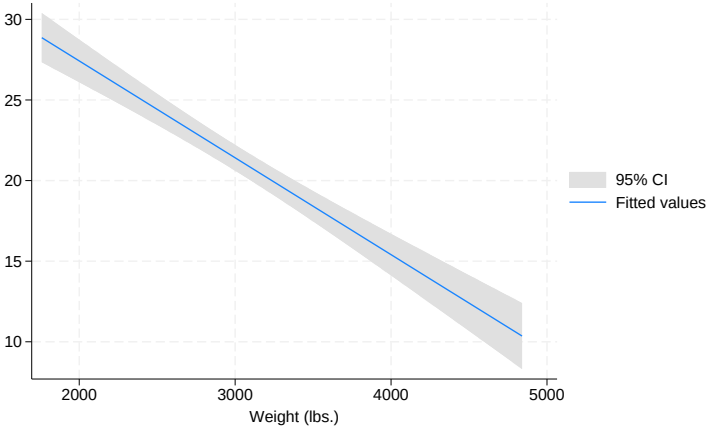
Remarks and examples

Remarks are presented under the following headings:

Typical use

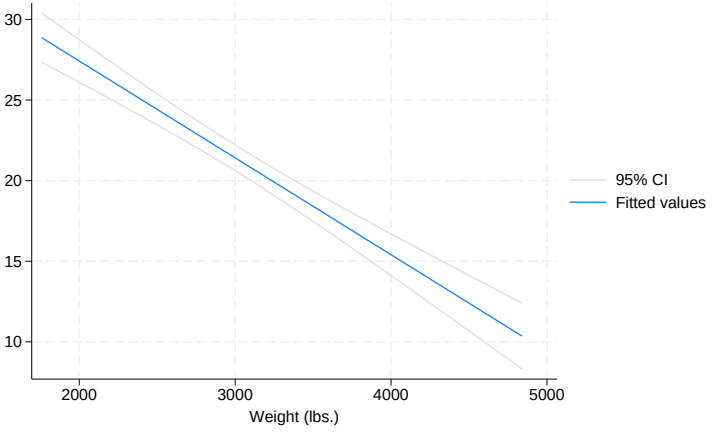
`twoway lfitci` by default draws the confidence interval of the predicted mean:

```
. use https://www.stata-press.com/data/r19/auto
(1978 automobile data)
. twoway lfitci mpg weight
```



If you specify the `ciplot(rline)` option, then rather than being shaded, the confidence interval will be designated by lines:

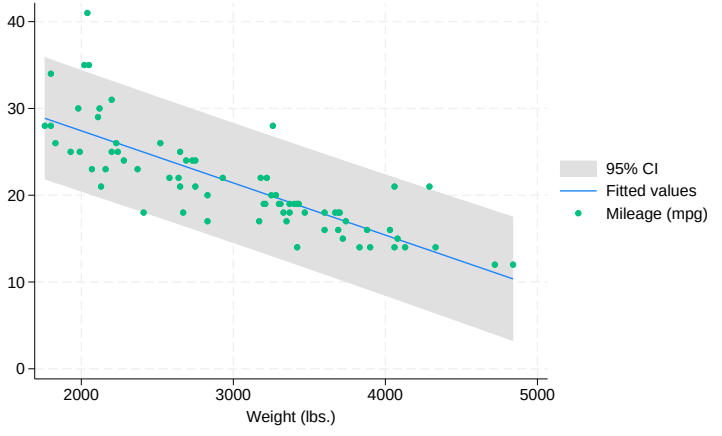
```
. twoway lfitci mpg weight, ciplot(rline)
```



Advanced use

`lfitci` can be usefully overlaid with other plots:

```
. use https://www.stata-press.com/data/r19/auto, clear
(1978 automobile data)
. twoway lfitci mpg weight, stdf || scatter mpg weight
```



In the above example, we specified `stdf` to obtain a confidence interval based on the standard error of the forecast rather than the standard error of the mean. This is more useful for identifying outliers.

We typed

```
. twoway lfitci ... || scatter ...
```

and not

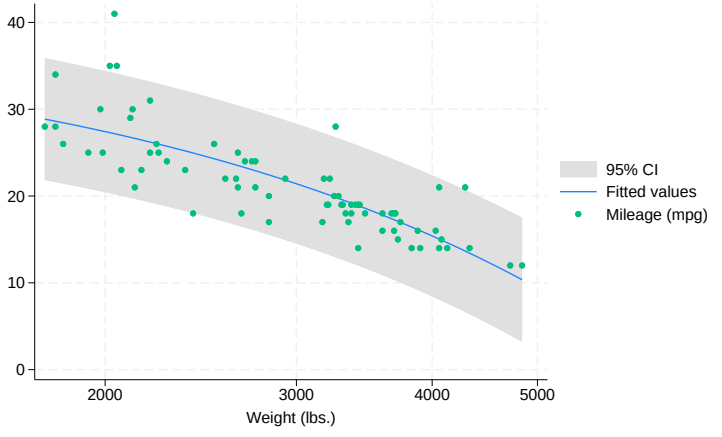
```
. twoway scatter ... || lfitci ...
```

Had we drawn the scatter diagram first, the confidence interval would have covered up most of the points.

Cautions

Do not use `twoway lfitci` when specifying the *axis_scale_options* `yscale(log)` or `xscale(log)` to create log scales. Typing

```
. twoway lfitci mpg weight, stdf || scatter mpg weight ||, xscale(log)
```

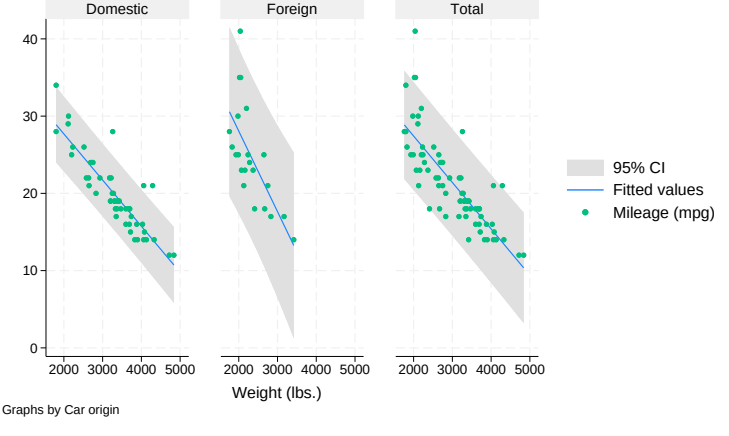


The result may look pretty, but if you think about it, it is not what you want. The prediction line is not straight because the regression estimated for the prediction was for `mpg` on `weight`, not for `mpg` on `log(weight)`.

Use with by()

`lfitci` may be used with `by()` (as can all the `twoway` plot commands):

```
. twoway lfitci  mpg weight, stdf ||
        scatter mpg weight       ||
  , by(foreign, total row(1))
```



Also see

[G-2] **graph twoway fpfitci** — Two-way fractional-polynomial prediction plots with CIs

[G-2] **graph twoway qfitci** — Two-way quadratic prediction plots with CIs

[G-2] **graph twoway lfit** — Two-way linear prediction plots

[R] **regress** — Linear regression