

[Description](#)
[Options](#)[Quick start](#)
[Remarks and examples](#)[Menu](#)
[Also see](#)[Syntax](#)

Description

`twoway qfitci` calculates the prediction for *yvar* from a regression of *yvar* on *xvar* and *xvar*² and plots the resulting line along with a confidence interval.

Quick start

Graph of the quadratic prediction of *y* using *x* and *x*² with 95% confidence interval (CI)

```
twoway qfitci y x
```

Add an overlaid scatterplot of the observed data

```
twoway qfitci y x || scatter y x
```

Same as above, but calculate CI using the standard error for an individual forecast

```
twoway qfitci y x, stdf || scatter y x
```

With a 90% confidence interval

```
twoway qfitci y x, level(90) || scatter y x
```

Suppress the legend

```
twoway qfitci y x || scatter y x, legend(off)
```

Same as above, but plot confidence limits as lines

```
twoway qfitci y x, ciplot(rline) || scatter y x, legend(off)
```

Same as above, and reduce the size of the scatterplot markers

```
twoway qfitci y x, ciplot(rline) || scatter y x, legend(off) ///  
msize(.8)
```

Menu

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

Syntax

```
twoway qfitci yvar xvar [if] [in] [weight] [, options]
```

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

Options `range()`, `estopts()`, `predopts()`, `n()`, and `level()` are *rightmost*, and `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.

*aweight*s, *fweight*s, and *pweight*s 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 curve will be predicted; see [R] [regress](#). If this option is specified, 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

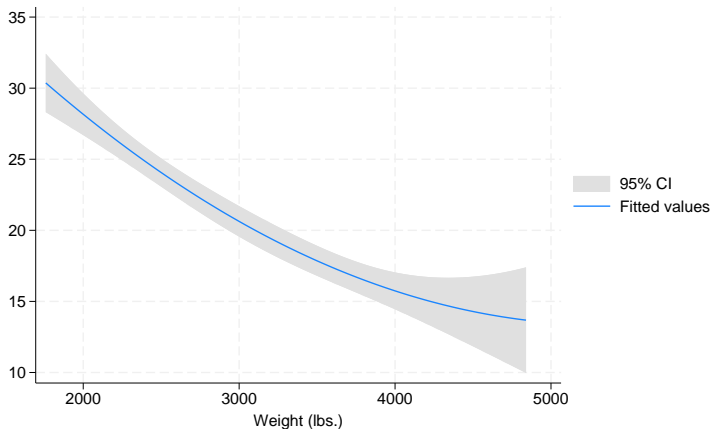
Remarks are presented under the following headings:

Typical use
Advanced use
Cautions
Use with by()

Typical use

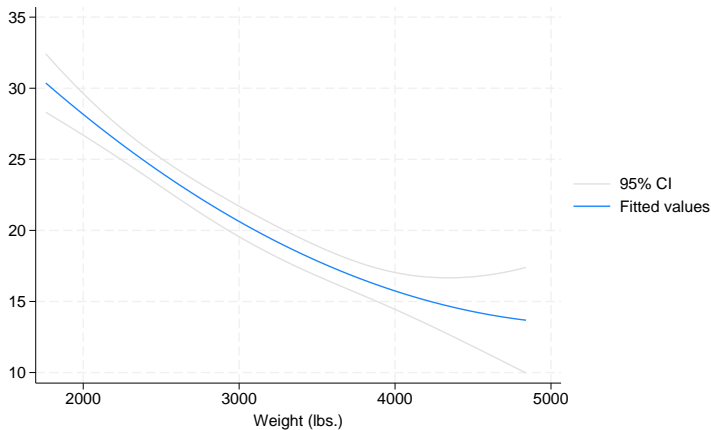
twoway qfitci by default draws the confidence interval of the predicted mean:

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



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

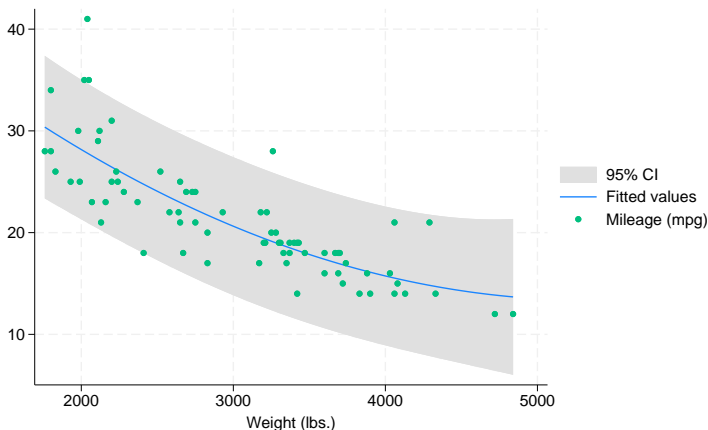
```
. twoway qfitci mpg weight, ciplot(rline)
```



Advanced use

qfitci can be overlaid with other plots:

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



In the above command, 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 qfitci ... || scatter ...
```

and not

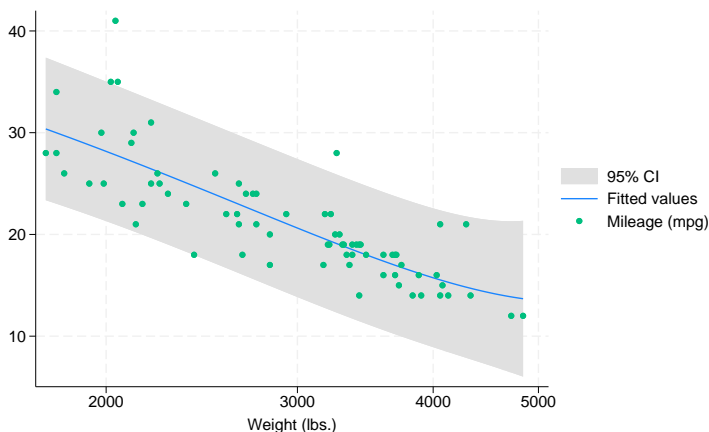
```
. twoway scatter ... || qfitci ...
```

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

Cautions

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

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

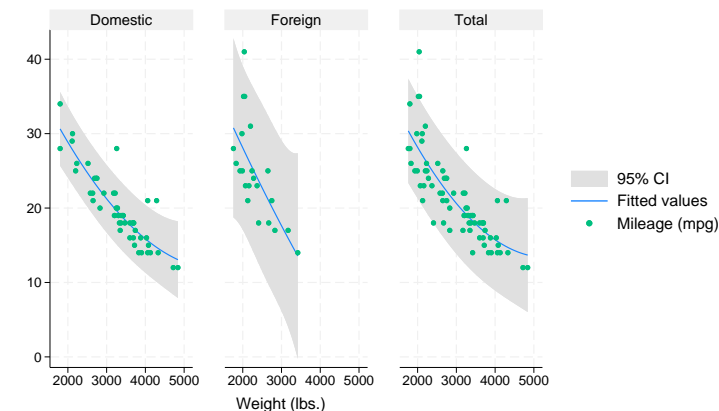


The result may look pretty but, if you think about it, it is not what you want. The prediction line is not a parabola because the regression estimated for the prediction was for `mpg` on `weight` and `weight^2`, not `mpg` on `log(weight)` and `log(weight)^2`.

Use with by()

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

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



Graphs by Car origin

Also see

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

[G-2] [graph twoway lfitei](#) — Two-way linear prediction plots with CIs

[G-2] [graph twoway qfit](#) — Two-way quadratic prediction plots

[R] [regress](#) — Linear regression

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](#).