Title

stata.com

nlsur postestimation — Postestimation tools for nlsur

Description

The following postestimation commands are available after `nlsur`:

Command	Description
`estat ic`	Akaike's and Schwarz's Bayesian information criteria (AIC and BIC)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`forecast`	dynamic forecasts and simulations
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`lrtest`	likelihood-ratio test
`margins`[1]	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

[1] You must specify the `variables()` option with `nlsur`.

Syntax for predict

`predict` [*type*] *newvar* [*if*] [*in*] [`, equation(#`*eqno*`) yhat residuals`]

These statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample.

Menu for predict

Statistics > Postestimation > Predictions, residuals, etc.

Options for predict

Main

`equation(#eqno)` specifies to which equation you are referring. `equation(#1)` would mean that the calculation is to be made for the first equation, `equation(#2)` would mean the second, and so on. If you do not specify `equation()`, results are the same as if you had specified `equation(#1)`.

`yhat`, the default, calculates the fitted values for the specified equation.

`residuals` calculates the residuals for the specified equation.

Remarks and examples

stata.com

Example 1

In example 2 of [R] **nlsur**, we fit a four-factor translog cost function to data for the U.S. economy. The own-price elasticity for a factor measures the percentage change in its usage as a result of a 1% increase in the factor's price, assuming that output is held constant. For the translog production function, the own-price factor elasticities are

$$\eta_i = \frac{\delta_{ii} + s_i(s_i - 1)}{s_i}$$

Here we compute the elasticity for capital at the sample mean of capital's factor share. First, we use `summarize` to get the mean of `s_k` and store that value in a scalar:

```
. summarize s_k
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
         s_k |         25    .053488    .0044795     .04602     .06185
. scalar kmean = r(mean)
```

Now we can use `nlcom` to calculate the elasticity:

```
. nlcom (([dkk]_cons + kmean*(kmean-1)) / kmean)
       _nl_1:  ([dkk]_cons + kmean*(kmean-1)) / kmean
------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       _nl_1 |  -.3952986   .1083535    -3.65   0.000    -.6076676   -.1829295
------------------------------------------------------------------------------
```

If the price of capital increases by 1%, its usage will decrease by about 0.4%. To maintain its current level of output, a firm would increase its usage of other inputs to compensate for the lower capital usage. The standard error reported by `nlcom` reflects the sampling variance of the estimated parameter $\widehat{\delta_{kk}}$, but `nlcom` treats the sample mean of `s_k` as a fixed parameter that does not contribute to the sampling variance of the estimated elasticity.

Also see

[R] **nlsur** — Estimation of nonlinear systems of equations

[U] 20 Estimation and postestimation commands