

dsgenl postestimation — Postestimation tools for dsgenl

[Postestimation commands](#)
[predict](#)
[Remarks and examples](#)
[Methods and formulas](#)
[Also see](#)

Postestimation commands

The following postestimation commands are of special interest after `dsgenl`:

Command	Description
<code>estat covariance</code>	display model-implied covariance matrix of model variables
<code>estat policy</code>	display policy matrix of estimated model
<code>estat stable</code>	assess stability of the system
<code>estat steady</code>	display steady state of the system
<code>estat transition</code>	display transition matrix of estimated model
<code>irf</code>	create and analyze IRFs and FEVDs

The following standard postestimation commands are also available:

Command	Description
<code>estat ic</code>	Akaike's, consistent Akaike's, corrected Akaike's, and Schwarz's Bayesian information criteria (AIC, CAIC, AICc, and BIC)
<code>estat summarize</code>	summary statistics for the estimation sample
<code>estat vce</code>	variance–covariance matrix of the estimators (VCE)
<code>estimates</code>	cataloging estimation results
<code>etable</code>	table of estimation results
<code>forecast</code>	dynamic forecasts and simulations
<code>lincom</code>	point estimates, standard errors, testing, and inference for linear combinations of coefficients
<code>lrtest</code>	likelihood-ratio test
<code>nlcom</code>	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
<code>predict</code>	one-step-ahead predictions, prediction errors, and other diagnostic measures
<code>predictnl</code>	point estimates, standard errors, testing, and inference for generalized predictions
<code>test</code>	Wald tests of simple and composite linear hypotheses
<code>testnl</code>	Wald tests of nonlinear hypotheses

predict

Description for predict

`predict` creates new variables containing predictions such as expected values. Predictions are available as static one-step-ahead predictions or as dynamic multistep predictions, and you can control when dynamic predictions begin.

Menu for predict

Statistics > Postestimation

Syntax for predict

```
predict [type] {stub* | newvarlist} [if] [in] [, statistic options]
```

<i>statistic</i>	Description
------------------	-------------

Main

<code>xb</code>	linear prediction for observed variables
<code>states</code>	linear prediction for latent state variables

<i>options</i>	Description
----------------	-------------

Options

<code>rmse(<i>stub*</i> <i>newvarlist</i>)</code>	put estimated root mean squared errors of predicted statistics in new variables
<code>dynamic(<i>time_constant</i>)</code>	begin dynamic forecast at specified time

Advanced

<code>smethod(<i>method</i>)</code>	method for predicting unobserved states
-------------------------------------	---

<i>method</i>	Description
---------------	-------------

<code>onestep</code>	predict using past information
<code>filter</code>	predict using past and contemporaneous information

Options for predict

Main

`xb`, the default, calculates the linear predictions of the observed variables.

`states` calculates the linear predictions of the latent state variables.

Options

`rmse(stub* | newvarlist)` puts the root mean squared errors of the predicted statistics into the specified new variables. The root mean squared errors measure the variances due to the disturbances but do not account for estimation error.

`dynamic(time_constant)` specifies when `predict` starts producing dynamic forecasts. The specified *time_constant* must be in the scale of the time variable specified in `tsset`, and the *time_constant* must be inside a sample for which observations on the dependent variables are available. For example, `dynamic(tq(2008q4))` causes dynamic predictions to begin in the fourth quarter of 2008, assuming that your time variable is quarterly; see [D] [Datetime](#). If the model contains exogenous variables, they must be present for the whole predicted sample.

Advanced

`smethod(method)` specifies the method for predicting the unobserved states, `smethod(onestep)` and `smethod(filter)`, and causes different amounts of information on the dependent variables to be used in predicting the states at each time period.

`smethod(onestep)`, the default, causes `predict` to estimate the states at each time period using previous information on the dependent variables. The Kalman filter is performed on previous periods, but only the one-step predictions are made for the current period.

`smethod(filter)` causes `predict` to estimate the states at each time period using previous and contemporaneous data by the Kalman filter. The Kalman filter is performed on previous periods and the current period. `smethod(filter)` may be specified only with `states`.

Remarks and examples

[stata.com](http://www.stata.com)

For examples of `estat covariance`, see [DSGE] [Intro 3e](#) and [DSGE] [estat covariance](#).

For examples of `estat policy`, see [DSGE] [Intro 1](#), [DSGE] [Intro 3d](#), and [DSGE] [Intro 3e](#).

For examples of `estat transition`, see [DSGE] [Intro 1](#), [DSGE] [Intro 3d](#), and [DSGE] [Intro 3e](#).

For examples of `irf` after `dsgenl`, see [DSGE] [Intro 3d](#) and [DSGE] [Intro 3e](#).

For examples of `predict` and `forecast` after fitting a DSGE model, see [DSGE] [Intro 1](#). Examples are shown with estimation results from `dsge` but apply to results from `dsgenl` as well.

Methods and formulas

The most important thing to keep in mind is that postestimation predictions after `dsgenl` are made on the basis of the linearized form of the model given to `dsgenl`.

Estimating the unobserved states is the key to predicting the observed variables.

By default and with the `smethod(onestep)` option, `predict` estimates the state in each period by applying the Kalman filter to all previous periods and only making the one-step prediction to the current period.

With the `smethod(filter)` option, `predict` estimates the states in each period by applying the Kalman filter on all previous periods and the current period. The computational difference between `smethod(onestep)` and `smethod(filter)` is that `smethod(filter)` performs the update step on the current period while `smethod(onestep)` does not. The statistical difference between `smethod(onestep)` and `smethod(filter)` is that `smethod(filter)` uses contemporaneous information on the observed variables while `smethod(onestep)` does not.

The observed control variables are predicted by plugging in the estimated states.

Also see

[DSGE] **dsgenl** — Nonlinear dynamic stochastic general equilibrium models

[DSGE] **estat covariance** — Display estimated covariances of model variables

[DSGE] **estat policy** — Display policy matrix

[DSGE] **estat stable** — Check stability of system

[DSGE] **estat steady** — Display steady state of nonlinear DSGE model

[DSGE] **estat transition** — Display state transition matrix

[TS] **forecast** — Econometric model forecasting

[TS] **irf** — Create and analyze IRFs, dynamic-multiplier functions, and FEVDs

[U] **20 Estimation and postestimation commands**

Stata, Stata Press, and Mata are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow and NetCourseNow are trademarks of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2023 StataCorp LLC, College Station, TX, USA. All rights reserved.



For suggested citations, see the FAQ on [citing Stata documentation](#).