

dsge — Linearized dynamic stochastic general equilibrium models

Description	Menu	Syntax	Options	Remarks
Stored results	Methods and formulas	References	Also see	

Description

`dsge` fits linearized dynamic stochastic general equilibrium (DSGE) models to multiple time series. DSGE models are systems of equations that are motivated by economic theory. In these systems, expectations of future values of variables can affect the current values. The parameters of these models are often directly interpretable in terms of economic theory.

Menu

Statistics > Multivariate time series > Dynamic stochastic general equilibrium (DSGE) models

Syntax

```
dsge eqlist [if] [in] [, options]
```

<i>options</i>	Description
Main	
<code><u>constraints</u>(<i>constraints</i>)</code>	apply specified linear constraints
<code><u>no</u>idencheck</code>	do not check for parameter identification
<code>solve</code>	return model solution at initial values
SE/Robust	
<code>vce(<i>vcetype</i>)</code>	<i>vcetype</i> may be <code>oim</code> or <code>robust</code>
Reporting	
<code><u>level</u>(#)</code>	set confidence level; default is <code>level(95)</code>
<code><u>no</u>cnsreport</code>	do not display constraints
<code><u>display_options</u></code>	control columns and column formats and line width
Maximization	
<code><u>maximize_options</u></code>	control the maximization process
Advanced	
<code><u>no</u>demean</code>	do not demean data prior to estimation
<code>post</code>	force posting of estimation results in the event of errors caused by lack of identification or stability
<code><u>id</u>tolerance(#)</code>	set tolerance used for identification check; seldom used
<code><u>lin</u>tolerance(#)</code>	set tolerance used for linearity check; seldom used
<code><u>coef</u>legend</code>	display legend instead of statistics

You must `tsset` your data before using `dsge`; see [TS] `tsset`.

`coeflegend` does not appear in the dialog box.

See [U] 20 [Estimation and postestimation commands](#) for more capabilities of estimation commands.

Below we present the full specification of `eqlist`. You may prefer to start with the syntax discussion in [DSGE] [intro 2](#).

`eqlist` is

```
eq
eq eqlist
```

`eq` is

```
cntrl_eq
state_eq
```

`cntrl_eq` contains

```
(ocntrl_var = termlist)
(ucntrl_var = termlist, unobserved)
(parm_exp * ocntrl_var = termlist)
(parm_exp * ucntrl_var = termlist, unobserved)
```

`state_eq` contains

```
(F.state_var = , state)
(F.state_var = termlist, state [noshock])
(parm_exp * F.state_var = termlist, state [noshock])
```

`ocntrl_var` is

Stata variable to be treated as an observed control in the system

`ucntrl_var` is

name to be treated as an unobserved control in the system

`state_var` is

name to be treated as an unobserved state in the system

If there happens to be a Stata variable with the same name as `ucntrl_var` or `state_var`, the variable is ignored and plays no role in the estimation.

`termlist` is

```
term
term + termlist
term - termlist
```

`term` is

```
rhs_var
parm_exp * rhs_var
parm_exp * (termlist)
```

`rhs_var` is

```
ocntrl_var
ucntrl_var
state_var
E(F.ocntrl_var)
E(F.ucntrl_var)
```

<i>parm_exp</i> is	scalar substitutable expression; <i>parm_spec</i> elements are allowed and expected <i>rhs_var</i> are not allowed
<i>parm_spec</i> is	{ <i>parm_name</i> } { <i>parm_name</i> = <i>initial_val</i> }
<i>parm_name</i> is	name used to identify model parameter
<i>initial_val</i> is	numeric literal; specifies an initial value

Options

Main

`constraints`(*constraints*); see [R] [estimation options](#).

`noidencheck` skips the check that the parameters are identified at the initial values. Models that are not structurally identified can still converge, thereby producing meaningless results that only appear to have meaning; thus care should be taken in specifying this option. See [DSGE] [intro 6](#) for details.

`solve` puts the model into state-space form at the initial parameter values. No standard errors are produced.

SE/Robust

`vce`(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (`oim`) and that are robust to some kinds of misspecification (`robust`); see [R] [vce_option](#).

Reporting

`level`(#), `nocnsreport`; see [R] [estimation options](#).

`display_options`: `nocl`, `nopvalues`, `cformat`(%*fmt*), `pformat`(%*fmt*), `sformat`(%*fmt*), and `nolstretch`; see [R] [estimation options](#).

Maximization

`maximize_options`: `difficult`, `technique`(*algorithm_spec*), `iterate`(#), `[no]log`, `trace`, `gradient`, `showstep`, `hessian`, `showtolerance`, `tolerance`(#), `ltolerance`(#), `nrtolerance`(#), and `from`(*init_specs*); see [R] [maximize](#).

Advanced

`nodemean` prevents `dsge` from removing the mean of the observed control variables prior to estimation.

`post` causes `dsge` to post the parameter vector into `e()`, even in the event of errors that arise from checking stability conditions or identification.

`idtolerance`(#) specifies the tolerance used in the identification diagnostic. The default is `idtolerance(1e-6)`.

`lntolerance`(#) specifies the tolerance used in the linearity diagnostic. The default is `lntolerance(1e-12)`.

The following option is available with `dsge` but is not shown in the dialog box: `coeflegend`; see [R] [estimation options](#).

Remarks

For an introduction to what `dsge` can do and how to use it, see [DSGE] [intro 1](#). It is highly recommended that you read the introduction first.

For examples of `dsge`, see the examples of classic DSGE models in [DSGE] [intro 3a](#), [DSGE] [intro 3b](#), and [DSGE] [intro 3c](#). Additional examples are presented in [DSGE] [intro 4a](#)–[DSGE] [intro 4g](#). See [DSGE] [intro 4](#) for an overview of these examples.

Stored results

`dsge` stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(k)</code>	number of parameters
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_dv)</code>	number of dependent variables
<code>e(k_state)</code>	number of state equations
<code>e(k_control)</code>	number of control equations
<code>e(k_shock)</code>	number of shocks
<code>e(k_observed)</code>	number of observed control equations
<code>e(k_stable)</code>	number of stable eigenvalues
<code>e(ll)</code>	log likelihood
<code>e(tmin)</code>	minimum time in sample
<code>e(tmax)</code>	maximum time in sample
<code>e(rank)</code>	rank of VCE
<code>e(ic)</code>	number of iterations
<code>e(rc)</code>	return code
<code>e(converged)</code>	1 if converged, 0 otherwise

Macros

<code>e(cmd)</code>	<code>dsge</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	unoperated names of dependent variables
<code>e(state)</code>	unoperated names of state variables
<code>e(control)</code>	unoperated names of control variables
<code>e(observed)</code>	unoperated names of observed control variables
<code>e(title)</code>	title in estimation output
<code>e(tvar)</code>	variable denoting time within groups
<code>e(tmins)</code>	formatted minimum time
<code>e(tmaxs)</code>	formatted maximum time
<code>e(vce)</code>	<i>vctype</i> specified in <code>vce()</code>
<code>e(vctype)</code>	title used to label Std. Err.
<code>e(opt)</code>	type of optimization
<code>e(method)</code>	likelihood method
<code>e(idencheck)</code>	passed, failed, or skipped
<code>e(technique)</code>	maximization technique
<code>e(properties)</code>	<code>b V</code>
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>

Matrices

e(b)	parameter vector
e(Cns)	constraints matrix
e(ilog)	iteration log (up to 20 iterations)
e(eigenvalues)	generalized eigenvalues
e(gradient)	gradient vector
e(shock_coeff)	estimated shock coefficient matrix
e(transition)	estimated state transition matrix
e(policy)	estimated policy matrix
e(V)	variance–covariance matrix of the estimators
e(V_modelbased)	model-based variance

Functions

e(sample)	marks estimation sample
-----------	-------------------------

Methods and formulas

A DSGE model is a system of equations in which the values of the control variables \mathbf{y}_t and of the state variables \mathbf{x}_t depend on their values in previous periods and their one-period-ahead rational expectations; see [DSGE] [intro 1](#) for an introduction to control variables and state variables. The structural form of a DSGE model is

$$\mathbf{A}_0 \mathbf{y}_t = \mathbf{A}_1 E_t(\mathbf{y}_{t+1}) + \mathbf{A}_2 \mathbf{y}_t + \mathbf{A}_3 \mathbf{x}_t \tag{1}$$

$$\mathbf{B}_0 \mathbf{x}_{t+1} = \mathbf{B}_1 E_t(\mathbf{y}_{t+1}) + \mathbf{B}_2 \mathbf{y}_t + \mathbf{B}_3 \mathbf{x}_t + \mathbf{C} \boldsymbol{\epsilon}_{t+1} \tag{2}$$

where the entries in \mathbf{A}_i and \mathbf{B}_i are functions of the parameter vector $\boldsymbol{\theta}$ and \mathbf{C} is a selector matrix of zeros and ones that puts shocks in the state equations that include them. The \mathbf{A}_0 and \mathbf{B}_0 matrices are diagonal, and \mathbf{A}_2 has zeros on its diagonal. See [DSGE] [intro 1](#) for details about how the structural parameters $\boldsymbol{\theta}$ get mapped into entries in \mathbf{A}_i and \mathbf{B}_i matrices.

dsge estimates the structural parameters $\boldsymbol{\theta}$ by maximum likelihood. The remainder of this entry provides some details about this process.

We can estimate only the parameters of structural models like (1) and (2) that we can solve for the state-space form that expresses the control variables as functions of the state variables and expresses the state variables as first-order autoregressive processes, as in (3) and (4),

$$\mathbf{y}_t = \mathbf{G}(\boldsymbol{\theta}) \mathbf{x}_t \tag{3}$$

$$\mathbf{x}_{t+1} = \mathbf{H}(\boldsymbol{\theta}) \mathbf{x}_t + \mathbf{M}(\boldsymbol{\theta}) \boldsymbol{\epsilon}_{t+1} \tag{4}$$

where \mathbf{G} is known as the policy matrix and \mathbf{H} is known as the state transition matrix. The elements in the policy and state transition matrices are functions of the structural parameters that appear in the matrices \mathbf{A} and \mathbf{B} .

dsge implements the solution technique described in Klein (2000), which obtains the state-space form in (3) and (4) from the structural model in (1) and (2).

We can only solve a structural model for its state-space form when i) the structural form has the structure implied by (1) and (2) and when ii) the values of the structural parameters imply that the model neither spirals out of control nor converges to a point, a condition known as saddle-path stability.

Condition (i) seems trivial, but some minor manipulation is frequently required to make it true. Many DSGE models include a problematic term like a shock to a control variable or a lagged state variable. These problematic terms do not fit into the structure required by (1) and (2), but the model is easily rewritten to accommodate these problematic terms; see [DSGE] [intro 4](#) for details.

Condition (ii), the saddle-path stability condition, depends on the values of the parameters. We can only solve models for parameter values that imply a saddle-path-stable model; see [DSGE] [intro 5](#) for details.

Assuming that the shocks ϵ_t are mean-zero, independent, and normally distributed implies that (3) and (4) form a linear state-space model in which the coefficients are nonlinear functions of the structural parameters θ . This structure allows `dsge` to estimate the parameters θ and the standard deviations of the shocks by maximum likelihood.

The Kalman filter is used to form the log-likelihood function. The Kalman filter is a method for recursively obtaining linear, least-squares forecasts conditional on past information. These forecasts are used to construct the log likelihood, assuming normality and stationarity. See *Methods and formulas* in [TS] [sspace](#) for details of the Kalman filter.

When the shocks are independent and identically distributed, but not normally distributed, the maximum likelihood estimator is consistent for θ , but `vce(robust)` standard errors must be used. (The process of using a maximum likelihood estimator under weaker distributional assumptions and correcting the standard errors is known as a quasi-maximum likelihood estimator in the literature.)

One final caveat is that only a subset $\mathbf{y}_{1,t}$ of the controls is treated as observed in some models. This caveat is handled by augmenting the state-space systems in (3) and (4) with the observation equation

$$\mathbf{y}_{1,t} = \mathbf{D}\mathbf{y}_t$$

where \mathbf{D} is a selection matrix that picks out the rows in \mathbf{G} corresponding to elements in the control vector that are observed.

Asymptotic standard errors for postestimation objects such as the entries in the state-space matrices and the impulse responses are obtained using the delta method. See [Serfling \(1980, sec. 3.3\)](#) for a discussion of the delta method.

References

- Klein, P. 2000. Using the generalized Schur form to solve a multivariate linear rational expectations model. *Journal of Economic Dynamics and Control* 24: 1405–1423.
- Serfling, R. J. 1980. *Approximation Theorems of Mathematical Statistics*. New York: Wiley.

Also see

- [DSGE] [dsge postestimation](#) — Postestimation tools for `dsge`
- [DSGE] [intro 2](#) — Learning the syntax
- [TS] [sspace](#) — State-space models
- [TS] [tsset](#) — Declare data to be time-series data
- [TS] [var](#) — Vector autoregressive models
- [U] [20 Estimation and postestimation commands](#)