

Intro 9 — Bayesian estimation

[Description](#)[Remarks and examples](#)[Also see](#)

Description

This entry introduces Bayesian estimation of DSGE models. We discuss the setup of such models, the specification of priors, diagnostics tests, and other statistical and computational details important to fit Bayesian models.

We assume that you are already familiar with the elements of a DSGE; see [\[DSGE\] Intro 1](#). For a comprehensive discussion of Bayesian analysis and Bayesian methods using Stata, see [\[BAYES\] Intro](#) and [\[BAYES\] Bayesian commands](#).

Remarks and examples

stata.com

Remarks are presented under the following headings:

[*Introduction*](#)[*Principles of Bayesian DSGE estimation*](#)[*An uninformative prior*](#)[*An informative prior*](#)[*Convergence diagnostics*](#)

Introduction

Bayesian estimation provides an alternative to maximum likelihood estimation. Bayesian analysis combines a likelihood model for the data with a specification of prior knowledge of the distribution of parameters to arrive at a posterior distribution for the parameters. Incorporation of prior information in DSGE models takes several forms.

First, the prior incorporates information that would be difficult or impossible to specify through the likelihood, broadening the amount of information that can be incorporated into the model. The likelihood is typically specified in terms of aggregate equations. However, some information about the parameters can be drawn from microeconomic evidence. This evidence can be incorporated into estimation via the prior. Second, the prior incorporates logical bounds on values that a parameter might take. Some parameters represent shares, like a capital share of income or depreciation rate of capital, so that only the (0,1) interval is an appropriate value for the parameter. A prior that takes on positive value in (0,1) but has a zero probability elsewhere restricts the search to the desired region. Third, the prior can reflect information from theory, for instance, that a parameter should be positive or should lie above a critical threshold. All of this makes Bayesian estimation of DSGE models an appealing alternative to classical frequentist estimation.

Stata provides a full suite of commands for Bayesian estimation. See [\[BAYES\] Intro](#) for an introduction to Bayesian analysis and [\[BAYES\] Bayesian commands](#) for an introduction to Stata's implementation of Bayesian analysis. This entry focuses on Stata's implementation of Bayesian estimation for DSGE models.

Bayesian estimation of DSGE models is accomplished through use of the `bayes:` prefix. If your DSGE model is

```
. dsge (y = z) (F.z = {rho}*z, state)
```

then the corresponding Bayesian estimates can be obtained with

```
. bayes, prior({rho}, uniform(0,1)): dsge (y=z) (F.z={rho}*z, state)
```

The command has two parts: the DSGE model and the prior specification. The DSGE model is specified exactly as it is when estimating using maximum likelihood. Priors are specified with the `prior()` option of `bayes`. Most `bayes` commands come with default priors, intended to be uninformative; see [Remarks and examples](#) in `[BAYES] bayes`. However, in the case of DSGE models, default priors are only provided for standard deviation parameters. All other model parameters must be accompanied by an explicit prior. DSGE parameters are often logically bounded to a certain range by the model structure, such as the (0,1) interval or greater than some critical value, so that uninformative priors over the entire real line are inappropriate. Because DSGE parameters are model dependent, informative priors are readily available and in most cases necessary for estimates to be reasonable.

Principles of Bayesian DSGE estimation

An uninformative prior

We begin with an autoregressive model of order 1, written in its state-space form. This model is used to provide an introduction to the syntax of `bayes: dsge` and the specification of priors. See [\[TS\] sspace](#) for more information on the state-space form of an autoregressive model, and see [\[BAYES\] bayes: var](#) for a more general approach to the estimation of autoregression and vector autoregression models using Bayesian methods. We can estimate the parameters of this model by maximum likelihood using the `dsge` command (see [\[DSGE\] dsge](#)). We use output growth `y` as the dependent variable.

```
. use https://www.stata-press.com/data/r17/usmacro2
(Federal Reserve Economic Data - St. Louis Fed, 2017-01-15)
. dsge (y = z) (F.z = {rho}*z, state), nolog
DSGE model
Sample: 1955q1 thru 2015q4                Number of obs = 244
Log likelihood = -639.18399
```

	y	Coefficient	Std. err.	z	P> z	[95% conf. interval]
/structural						
rho		.3392504	.0608551	5.57	0.000	.2199765 .4585243
sd(e.z)		3.321503	.1503577			3.026808 3.616199

The first-order autocorrelation of output growth is 0.339, and the standard deviation of output shock is 3.32.

To estimate these parameters using Bayesian methods, specify a prior for model parameters, and use the `bayes:` prefix. Priors for most model parameters are required. Default priors are provided for the parameters representing the standard deviation of shocks. For a first example, we specify a flat prior for the autocorrelation parameter ρ in the range $(-1, 1)$. The command is

```
. bayes, prior({rho}, uniform(-1, 1)) : dsge (y = z) (F.z = {rho}*z, state)
```

Because Bayesian estimation is a stochastic procedure, we specify the random-number seed to ensure reproducibility.

```
. bayes, rseed(17) prior({rho}, uniform(-1,1)): dsge (y=z) (F.z={rho}*z, state)
note: initial parameter vector set to means of priors.
Burn-in ...
Simulation ...
Model summary
```

```
Likelihood:
  y ~ dsgell({rho},{sd(e.z)})
Priors:
  {rho} ~ uniform(-1,1)
  {sd(e.z)} ~ igamma(.01,.01)
```

Bayesian linear DSGE model	MCMC iterations =	12,500
Random-walk Metropolis-Hastings sampling	Burn-in =	2,500
	MCMC sample size =	10,000
Sample: 1955q1 thru 2015q4	Number of obs =	244
	Acceptance rate =	.2635
	Efficiency: min =	.07655
	avg =	.1075
	max =	.1385
Log marginal-likelihood = -648.5502		

	Mean	Std. dev.	MCSE	Median	Equal-tailed [95% cred. interval]	
rho	.3385195	.0611807	.001644	.3388822	.2251003	.465511
sd(e.z)	3.33619	.1564792	.005656	3.330808	3.062057	3.666267

The model summary shows the likelihood model and the priors used for model parameters. The likelihood model is `dsgell()`, meaning the state-space form of the DSGE model. The prior for `{rho}` is `uniform(-1, 1)`. The prior for the standard deviation of the state variable, `{sd(e.z)}`, is by default an inverse-gamma prior with parameters (0.01, 0.01).

The estimation header reports the total number of MCMC draws, the length of the burn-in period, the number of MCMC draws used for estimation, the number of observations in the dataset, the acceptance rate, and the efficiency rates of the draws. The acceptance rate specifies the proportion of proposed parameter values accepted by the MCMC algorithm. For the Metropolis–Hastings algorithm used by Bayesian DSGE estimation, this number is rarely above 50% and is often below 30%. A low acceptance rate—say, below 10%—can indicate convergence problems.

The first two columns of the estimation table report the posterior mean and posterior standard deviation. Because the prior was uninformative, the posterior mean of the autoregressive parameter `{rho}`, 0.339, is the same as the maximum likelihood estimate of 0.339. The estimation table also reports Monte Carlo standard errors, medians, and equal-tailed credible intervals (CrIs).

By default, the Bayesian estimates are based on an MCMC sample size of 10,000 after a 2,500-iteration burn-in. It is important to verify that the MCMC simulation has converged; otherwise, the estimates cannot be trusted. We examine convergence diagnostics later in [Convergence diagnostics](#).

An informative prior

Default priors are not provided for user-defined DSGE model parameters. Reasonable priors are model dependent and reflect the role the parameter plays in the model. Share parameters are bounded between 0 and 1; autoregressive parameters are bounded between -1 and 1; some parameters must

be larger than a critical value for model stability. Priors reflect these logical constraints. Additionally, priors reflect information on the probable location of a parameter within that range. Such information can be gleaned through prior studies or through different sources of information, such as microdata or economic theory. Thus priors must be specified for user-defined parameters, and informative priors can affect estimation results.

For standard deviations of shocks, an inverse-gamma prior with the shape and scale parameters of 0.01 is used. You can change the parameters of this prior by specifying the `igammaprior()` option, or you can specify a different prior distribution in the `prior()` option.

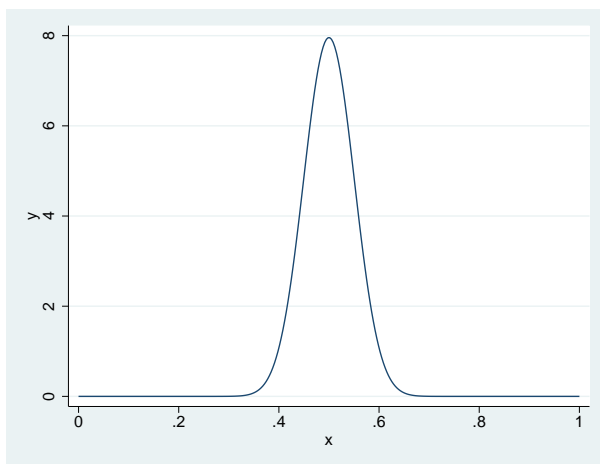
Priors provide several useful functions in DSGE models. Many parameters in DSGE models can be interpreted as share parameters: the capital share, the depreciation rate, the discount rate, etc. Such parameters are naturally bounded by the interval (0,1), and a uniform prior on (0,1) guarantees that the estimated parameter value lies within the economically sensible range. The beta distribution is frequently useful for share parameters, in which greater weight is to be given to a particular region of the (0,1) interval.

In addition, many parameters in DSGE models have microeconomic interpretation. As such, microeconomic information can be brought to bear on the estimation by using informative priors.

In Stata, priors are specified using their statistical parameters as arguments. A uniform prior on (a, b) is specified with `uniform(a, b)`. A normal prior with mean μ and variance σ^2 is specified with `normal(mu, sigma2)`. A beta distribution with shape parameters α and β is specified with `beta(alpha, beta)`. Uniform priors are useful for specifying an uninformative prior on a specified range. Normal priors are useful for parameters that can take on any real value. Beta priors are useful for parameters that are naturally bounded between 0 and 1, a case that is especially salient for the parameters of DSGE models. For a full list of priors, see [\[BAYES\] bayes](#).

We next use an informative prior for $\{\rho\}$ on (0,1), centered at 0.5. An appropriate distribution for this task is the beta distribution, and a fairly tight prior is obtained with the shape parameters (50, 50). A graph of the prior is

```
. twoway function y = betaden(50, 50, x)
```



Estimation with the $\text{beta}(50,50)$ prior is accomplished using the following command:

```
. bayes, rseed(17) prior({rho}, beta(50,50)): dsge (y=z) (F.z={rho}*z, state)
note: initial parameter vector set to means of priors.
Burn-in ...
Simulation ...
Model summary
```

```
Likelihood:
  y ~ dsgell({rho},{sd(e.z)})
Priors:
  {rho} ~ beta(50,50)
  {sd(e.z)} ~ igamma(.01,.01)
```

Bayesian linear DSGE model	MCMC iterations =	12,500
Random-walk Metropolis-Hastings sampling	Burn-in =	2,500
	MCMC sample size =	10,000
Sample: 1955q1 thru 2015q4	Number of obs =	244
	Acceptance rate =	.1459
	Efficiency: min =	.06455
	avg =	.0822
	max =	.09986
Log marginal-likelihood =	-648.3465	

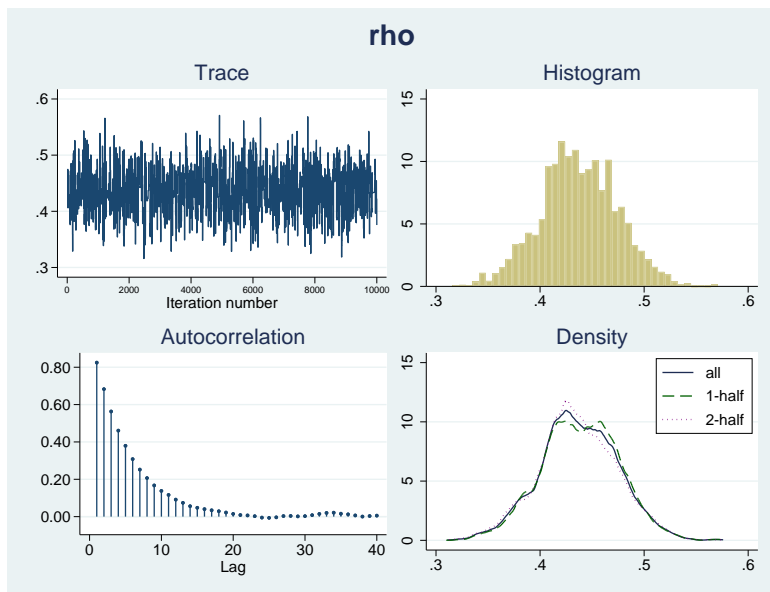
	Mean	Std. dev.	MCSE	Median	Equal-tailed [95% cred. interval]	
rho	.437131	.0373697	.001183	.4355395	.3616302	.5088862
sd(e.z)	3.35469	.1570423	.006181	3.344853	3.057471	3.684154

The acceptance rate is 15%, somewhat lower than the example with an uninformative prior. Sampling efficiencies are little changed at around 10%. The informative prior has pulled the point estimate of $\{\text{rho}\}$ away from its maximum likelihood estimate of 0.34; the posterior mean is now 0.437. The posterior mean for $\{\text{rho}\}$ lies about halfway between the prior mean of 0.5 and the maximum likelihood estimate of 0.34. The posterior distribution reflects a balance between prior information and the data at hand. More informative priors, indicated by a smaller standard deviation around the prior mean, tip the balance toward the prior mean. Larger sample sizes increase the importance of the data at hand, though macroeconomic datasets often face practical limitations on the available sample size.

Convergence diagnostics

Continuing with the [previous example](#), we use `bayesgraph` diagnostics to plot diagnostic charts for the autoregressive parameter $\{\text{rho}\}$. Four plots are provided: the MCMC draws, a histogram of the posterior distribution, the autocorrelation of the draws, and a density plot.

```
. bayesgraph diagnostics {rho}
```



We use these plots to visually assess two aspects of the MCMC chain: convergence and efficiency. There are no single conclusive convergence criteria for the MCMC chain, but some visual diagnostics can be used to explore MCMC convergence. (You can also run multiple chains to compute Gelman–Rubin convergence diagnostics; see [Convergence diagnostics using multiple chains](#).) The trace plot should not exhibit any time trend and should have constant mean and variance. These properties can be inspected in the trace plot in the top left. The density of the chain should not vary over the duration of the MCMC sample. Constancy of the distribution can be assessed in the 1-half and 2-half density plots in the bottom right; if these plots differ dramatically, then the chain has not converged. For further discussion of convergence, see [Convergence of MCMC](#).

Efficiency and mixing are terms used to describe how quickly the MCMC chain traverses the posterior domain. A chain that mixes well has high efficiency, low autocorrelation, and traverses the posterior quickly. A chain that mixes poorly has low efficiency and traverses the posterior slowly, exhibiting high autocorrelation in the MCMC chain. A chain that is less efficient requires more MCMC iterations to obtain the same information as a more efficient chain. Efficiency can also be assessed numerically; see [\[BAYES\] bayesstats ess](#).

For our autoregressive model, the trace plot shows no time trend and appears to have constant mean and variance. The 1-half and 2-half density plots overlap with each other and with the full chain density. These features are indicators that the chain has converged. The autocorrelation of the chain dies out quickly, indicating that the chain has acceptable efficiency. To verify convergence more formally, we could use the `nchains()` option to run multiple chains and to compute Gelman–Rubin convergence diagnostics.

In [\[DSGE\] Intro 9a](#) and [\[DSGE\] Intro 9b](#), we demonstrate Bayesian estimation using some of the models we have already developed in previous sections.

[\[DSGE\] Intro 9a](#) demonstrates Bayesian estimation of a linear New Keynesian model.

[\[DSGE\] Intro 9b](#) demonstrates Bayesian estimation of a nonlinear stochastic growth model.

Also see

[DSGE] **dsge** — Linear dynamic stochastic general equilibrium models

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

[BAYES] **bayes: dsge** — Bayesian linear dynamic stochastic general equilibrium models

[BAYES] **bayes: dsgenl** — Bayesian nonlinear dynamic stochastic general equilibrium models

[BAYES] **bayes** — Bayesian regression models using the bayes prefix

[BAYES] **Intro** — Introduction to Bayesian analysis

[BAYES] **Glossary**