

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

Description

This entry estimates and interprets the parameters of a simple New Keynesian model using Bayesian methods. We also discuss some postestimation features, diagnostic tests, and how to improve the sampling efficiency of parameters.

Remarks and examples

Remarks are presented under the following headings:

[The model](#)[Parameter estimation](#)[Posterior diagnostics and plots](#)[Improving sampling efficiency](#)[Impulse responses](#)

The model

Equations (1)–(5) specify a canonical New Keynesian model of inflation p_t , the output gap x_t , and the interest rate r_t . The linearized model is

$$p_t = \beta E_t(p_{t+1}) + \kappa x_t \quad (1)$$

$$x_t = E_t(x_{t+1}) - \{r_t - E_t(p_{t+1}) - g_t\} \quad (2)$$

$$r_t = \frac{1}{\psi} p_t + u_t \quad (3)$$

$$u_{t+1} = \rho_u u_t + \epsilon_{t+1} \quad (4)$$

$$g_{t+1} = \rho_g g_t + \xi_{t+1} \quad (5)$$

Equation (1) specifies inflation as a linear combination of expected future inflation and the output gap. Equation (2) specifies the output gap as a linear combination of the expected future output gap, the real interest rate, and a state variable g_t . Equation (3) specifies the interest rate as a linear combination of inflation and a state variable u_t . The state variables are modeled as first-order autoregressive processes. The state variable u_t is the deviation of r_t from its equilibrium value of $(1/\psi)p_t$. The state variable g_t is also the deviation of x_t from its equilibrium value.

Three of the parameters have a structural interpretation. The parameter κ is known as the slope of the Phillips curve and is predicted to be positive. The parameter β is the discount factor that represents the degree to which agents discount the future relative to the current period. The parameter $1/\psi$ measures the degree to which interest rates react to movements in inflation.

The choice of notation is intentional and indicates one way to use priors and parameter transformations. For model stability, the coefficient $1/\psi$ must be greater than one. For theoretical reasons, the region $(1.5, 2)$ is of particular interest. As such, the parameter ψ must lie between 0 and 1. By writing the model in terms of ψ , we use a beta distribution to restrict its range and to put larger weight on the region that is of interest. For example, a beta distribution for ψ that is centered at 0.67 maps on to a prior for $1/\psi$ that places much of its mass around 1.5, and this would be a good mix of logical and theoretical restrictions.

Parameter estimation

We specify priors for the model parameters. The discount rate β must lie between 0 and 1, with common values in the range (0.90, 0.99). The price-adjustment parameter κ is usually thought to be small and positive. The autocorrelation parameters must lie in $(-1, 1)$ but are typically believed to be positive and closer to 1 than 0. The parameter $1/\psi$ must be greater than 1 for model stability; hence, ψ must be between 0 and 1, making the beta distribution a good candidate. The parameters of the beta prior for ψ place its center of mass on 1.5, a value commonly found in the theoretical literature.

Priors were chosen to match the above theoretical considerations. For the discount rate β , a beta distribution with shape parameters (95, 5) is used. These shape parameters place the prior mean at 0.95 and place most of the prior mass in the region between 0.9 and 1. For the price-adjustment parameter κ , a beta distribution with shape parameters (30, 70) is used. These shape parameters place the prior mean at 0.3 and place most of the prior mass in the region between 0.2 and 0.4. For the Taylor rule parameter ψ , a beta distribution with shape parameters (0.67, 0.33) is used. These shape parameters place the prior mean at 0.67, so that its inverse is 1.5, a common value for this parameter in the literature. For the autoregressive parameters, a beta distribution with shape parameters (75, 25) is used. This places the prior mean of each state variable's autoregressive parameter at 0.8, reflecting a prior belief that the state variables show a fairly high degree of persistence.

```
. use https://www.stata-press.com/data/r19/usmacro2
(Federal Reserve Economic Data - St. Louis Fed, 2017-01-15)

. bayes, prior({beta}, beta(95, 5))
> prior({kappa}, beta(30, 70))
> prior({psi}, beta(67, 33))
> prior({rhoul}, beta(75, 20))
> prior({rhog}, beta(75, 20))
> rseed(17) :
> dsge (p = {beta}*F.p + {kappa}*x)
> (x = F.x - (r - F.p - g) , unobserved)
> (r = 1/{psi}*p + u)
> (F.u = {rhoul}*u, state)
> (F.g = {rhog}*g, state)
note: initial parameter vector set to means of priors.

Burn-in ...
Simulation ...

Model summary
```

```
Likelihood:
p r ~ dsgell({beta},{kappa},{psi},{rhoul},{rhog},{sd(e.u)},{sd(e.g)})

Priors:
{beta} ~ beta(95,5)
{kappa} ~ beta(30,70)
{psi} ~ beta(67,33)
{rhoul rhog} ~ beta(75,20)
{sd(e.u) sd(e.g)} ~ igamma(.01,.01)
```

```

Bayesian linear DSGE model          MCMC iterations =      12,500
Random-walk Metropolis-Hastings sampling  Burn-in       =       2,500
Sample: 1955q1 thru 2015q4          MCMC sample size =     10,000
                                     Number of obs    =       244
                                     Acceptance rate  =      .2209
                                     Efficiency: min  =      .01093
                                     avg             =      .02799
                                     max             =      .05246

Log marginal-likelihood = -796.75515

```

	Mean	Std. dev.	MCSE	Median	Equal-tailed [95% cred. interval]	
beta	.9330586	.0273878	.002619	.9354095	.8728009	.975816
kappa	.150112	.0247365	.001796	.1485866	.1038851	.203525
psi	.5905049	.0405852	.002398	.5895752	.5146669	.6752993
rhoul	.6275518	.0256116	.001511	.6283089	.5764364	.6770674
rhog	.9061189	.0131007	.001059	.9069891	.8800901	.9294626
sd(e.u)	2.114667	.1380783	.006028	2.104597	1.869307	2.400718
sd(e.g)	.5579649	.0578754	.002862	.5574834	.4491618	.6778793

Header output repeats the prior specification and reminds us that we are fitting a DSGE model. The MCMC acceptance rate is 0.2209, with efficiencies ranging from 1% to 5%. An acceptance rate of 20% to 25% is typical for these models. Acceptance rates that are too low indicate that a large portion of the proposed MCMC iterations were rejected, so that regions of high posterior probability were not sufficiently explored. Acceptance rates that are too high indicate that the MCMC iterations stayed in a relatively small area of high probability and did not sufficiently explore the parameter region. Efficiency is linked to the autocorrelation of the MCMC draws, with higher efficiency indicating lower autocorrelation.

Turning to parameter estimates, we see the posterior mean for {beta} is 0.93, close to the prior mean of 0.95. The posterior mean for {kappa} is 0.15, about halfway between the prior mean of 0.3 and the maximum likelihood estimate of 0.08 found in [DSGE] Intro 3a. The autoregressive parameters for the state variables are positive, with the state variable g showing autocorrelation {rhog} of 0.91 and the state variable u showing autocorrelation {rhoul} of 0.63.

Posterior diagnostics and plots

We begin by examining the efficiency of the MCMC draws.

```

. bayesstats ess
Efficiency summaries      MCMC sample size =      10,000
                          Efficiency: min  =      .01093
                          avg             =      .02799
                          max             =      .05246

```

	ESS	Corr. time	Efficiency
beta	109.35	91.45	0.0109
kappa	189.60	52.74	0.0190
psi	286.40	34.92	0.0286
rhoul	287.47	34.79	0.0287
rhog	153.01	65.36	0.0153
sd(e.u)	524.63	19.06	0.0525
sd(e.g)	408.84	24.46	0.0409

Some parameters have sampling efficiencies around 1%, indicating poor mixing. Low efficiency implies that it takes longer for the MCMC chain to explore the posterior distribution. We will come back to the issue of improving sampling efficiency in *Improving sampling efficiency* below.

Sometimes, the object of interest is a function of parameters rather than the parameter itself. Such a situation occurs in this model with the Taylor rule inflation-adjustment parameter $1/\psi$. The output is reported in terms of ψ by default, but what we really want to look at is $1/\psi$. We can analyze the posterior distribution of $1/\psi$ using the postestimation commands available after the `bayes` prefix. The `bayesstats summary` command produces posterior summary statistics for functions of model parameters. We investigate the parameter $1/\psi$ in some detail now.

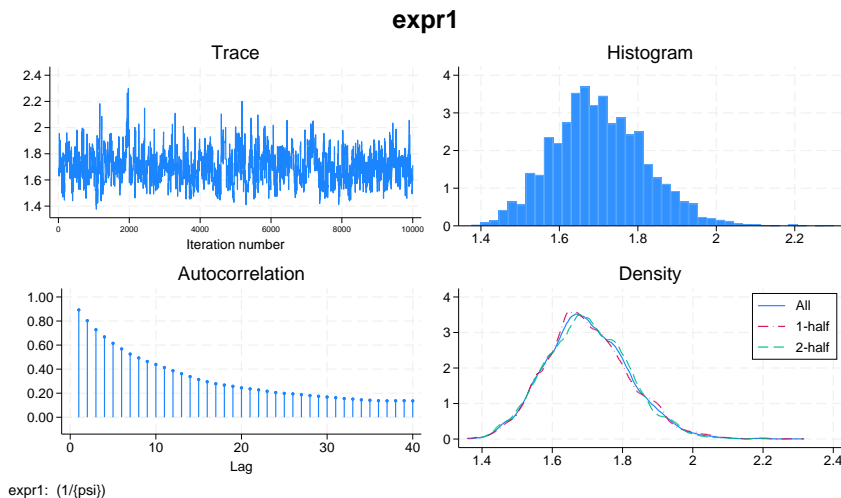
```
. bayesstats summary (1/{psi})
Posterior summary statistics                                MCMC sample size =    10,000
      expr1 : 1/{psi}
```

	Mean	Std. dev.	MCSE	Median	Equal-tailed [95% cred. interval]	
expr1	1.701543	.1181073	.006993	1.696136	1.480825	1.943004

The posterior mean is 1.7, somewhat higher than the prior mean of 1.5.

Next, we investigate the behavior of the MCMC chain using `bayesgraph`. (Actually, this is something that we should have done before obtaining summary statistics to verify that the corresponding MCMC chain converged.) To view the MCMC chain for $1/\psi$, `bayesgraph diagnostics` can be specified with the desired expression directly, bound in parentheses.

```
. bayesgraph diagnostics (1/{psi})
```

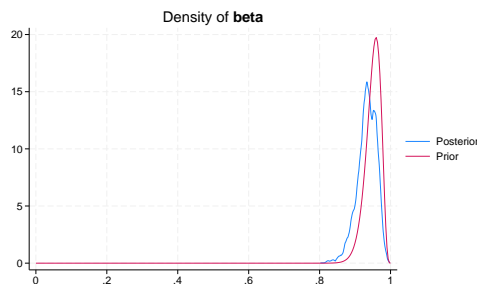
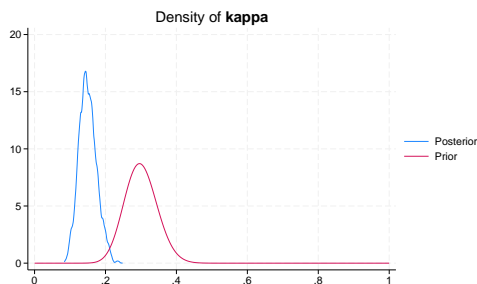


The trace plot shows reasonably good mixing, and the autocorrelations decay at a moderate pace. The density plot shows that the first- and second-half densities are similar to the density of the full MCMC sample. Densities that differ substantially in their first and second half can indicate nonconvergence.

Next, we compare the prior distribution and the posterior distributions for some model parameters directly. The `bayesgraph kdensity` command plots the kernel density of the posterior draws. The `addplot()` option can be used to add the prior to the plot. We plot the prior and posterior of the price-adjustment parameter $\{\kappa\}$ and the discount rate $\{\beta\}$.

```
. bayesgraph kdensity {kappa},
> addplot(function Prior = betaden(30,70, x),
> legend(on label(1 "Posterior"))))

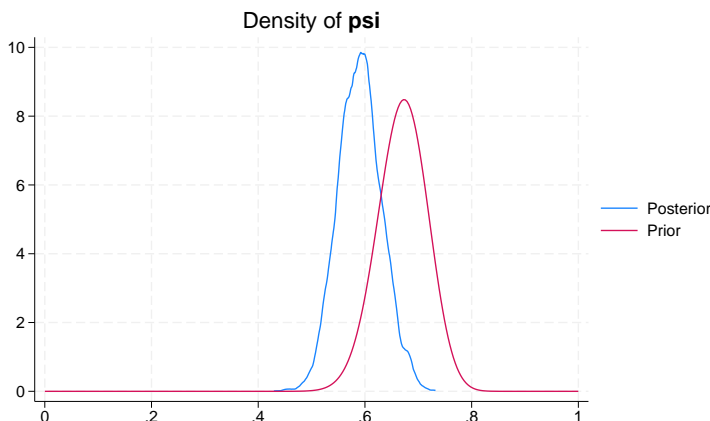
. bayesgraph kdensity {beta},
> addplot(function Prior = betaden(95, 5, x),
> legend(on label(1 "Posterior"))))
```



The price-adjustment parameter $\{\kappa\}$ is now centered on 0.15, rather than the prior mean of 0.3. By contrast, the discount rate parameter $\{\beta\}$ has shown little updating; the data are uninformative on this dimension, and the posterior overlaps with the prior.

Next, we turn to the Taylor rule parameter ψ .

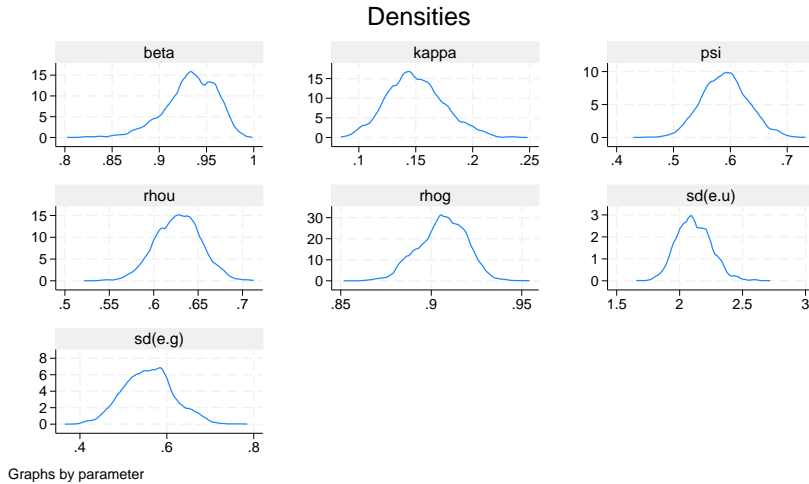
```
. bayesgraph kdensity {psi},
> addplot(function Prior = betaden(67, 33, x),
> legend(on label(1 "Posterior"))))
```



The posterior density for ψ places more weight on smaller values compared with the prior, implying a posterior density for $1/\psi$ that places more weight on larger values than the prior.

Finally, `bayesgraph kdensity _all, byparm` provides posterior kernel densities for all model parameters.

```
. bayesgraph kdensity _all, byparm
```



Such a graph can be useful in summarizing the posterior distribution of all parameters. Roughness observed in these densities can be smoothed by running a longer MCMC chain.

Improving sampling efficiency

The `bayesstats ess` output indicates lower sampling efficiency for several model parameters. Blocking of parameters can improve sampling efficiency. Parameters that are to be sampled independently are specified using the `block()` option of `bayes`. Efficiency can also be enhanced by running a longer MCMC chain, accomplished with the `mcmcsize()` option of `bayes`. A longer burn-in period can help with convergence. To demonstrate these procedures, we block some of the parameters and increase the length of the MCMC chain.

```

. bayes, prior({beta}, beta(95, 5))
> prior({kappa}, beta(30, 70))
> prior({psi}, beta(67, 33))
> prior({rhoul}, beta(70, 20))
> prior({rhog}, beta(70, 20))
> rseed(17)
> block({kappa}) block({rhoul}) block({sd(e.u)})
> mcmcsize(20000) dots :
> dsge (p = {beta}*F.p + {kappa}*x)
> (x = F.x - (r - F.p - g) , unobserved)
> (r = 1/{psi}*p + u)
> (F.u = {rhoul}*u, state)
> (F.g = {rhog}*g, state)
note: initial parameter vector set to means of priors.
Burn-in 2500 aaaaaaaaaa1000aaaaaaaaa2000aaaaaa done
Simulation 20000 .....1000.....2000.....3000.....4000.....5
> 000.....6000.....7000.....8000.....9000.....10000.....
> .11000.....12000.....13000.....14000.....15000.....16000.
> .....17000.....18000.....19000.....20000 done

```

Model summary

Likelihood:

```
p r ~ dsgell({beta},{kappa},{psi},{rhoul},{rhog},{sd(e.u)},{sd(e.g)})
```

Priors:

```

{beta} ~ beta(95,5)
{kappa} ~ beta(30,70)
{psi} ~ beta(67,33)
{rhoul rhog} ~ beta(70,20)
{sd(e.u) sd(e.g)} ~ igamma(.01,.01)

```

Bayesian linear DSGE model	MCMC iterations =	22,500
Random-walk Metropolis-Hastings sampling	Burn-in =	2,500
	MCMC sample size =	20,000
Sample: 1955q1 thru 2015q4	Number of obs =	244
	Acceptance rate =	.4085
	Efficiency: min =	.01979
	avg =	.03764
Log marginal-likelihood = -796.02145	max =	.06486

	Mean	Std. dev.	MCSE	Median	Equal-tailed [95% cred. interval]	
beta	.9369982	.0266526	.001015	.9401569	.8745056	.9795873
kappa	.1543932	.0267139	.001229	.1525291	.1081603	.2125788
psi	.5912198	.0415403	.001646	.5908567	.5109676	.6738902
rhoul	.6201272	.0272482	.001369	.620959	.5658674	.6728684
rhog	.905648	.0145937	.000405	.9062614	.8759466	.9333209
sd(e.u)	2.11736	.1461866	.00511	2.104432	1.85604	2.439449
sd(e.g)	.5544951	.0588497	.0019	.5528083	.4448559	.6758059

The above command placed three parameters into their own blocks: {kappa}, {rhog}, and {sd(e.g)}. (In this example, we specified three separate block() options, but we could have used the shortcut block({kappa rhog sd(e.g)}, split)). Blocking improves efficiency at the cost of longer run time. We see that efficiency has improved overall, with the minimum rising to 2% from 1% compared with the estimation without blocking.

```
. bayesstats ess
```

```
Efficiency summaries      MCMC sample size =    20,000
                          Efficiency:  min =    .01979
                                   avg =    .03764
                                   max =    .06486
```

	ESS	Corr. time	Efficiency
beta	689.22	29.02	0.0345
kappa	472.47	42.33	0.0236
psi	637.19	31.39	0.0319
rhoh	395.87	50.52	0.0198
rhog	1297.26	15.42	0.0649
sd(e.u)	818.49	24.44	0.0409
sd(e.g)	959.44	20.85	0.0480

Effective sample sizes have improved. In particular, the effective sample size for the price-adjustment parameter {kappa} has more than doubled from 189.60 to 472.47.

Impulse responses

To compute Bayesian impulse–response functions, use `bayesirf create` after estimation. We also need to save off our MCMC dataset. To do this, we replay the `bayes` command with the `saving()` option.

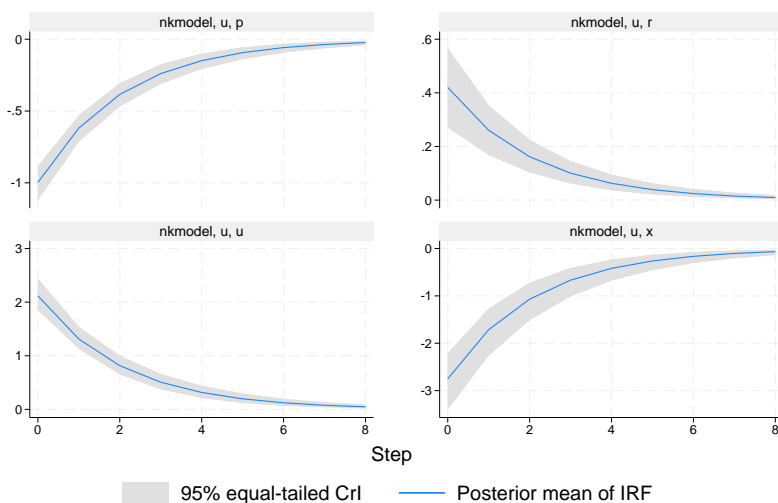
```
. bayes, saving(bdsge_nksim, replace)
```

With the MCMC results saved, we construct impulse–response functions.

```
. bayesirf set bayesirf.irf
. bayesirf create nkmodel
```

Results can be tabulated or graphed with `bayesirf table` and `bayesirf graph`, respectively.

```
. bayesirf graph irf, impulse(g) response(x p r g) byopts(yrescale)
```



Graphs by irfname, impulse variable, and response variable

Plotted are the responses of the output gap x , inflation p , interest rate r , and the state variable u to a shock to u . The steps are in the units of time that were used for estimation, so in this case, one step is one quarter. Eight steps are computed and plotted by default. A rise in the monetary shock causes inflation and interest rates to fall. The output gap becomes negative, indicating that this is a contractionary shock. All three endogenous variables return smoothly to their steady-state values over time, as the effect of the shock dissipates.

Along with point estimates, 95% equal-tailed credible intervals are shown. The point estimates are comparable with those seen in [DSGE] [Intro 1](#). But the credible intervals are somewhat narrower than the confidence intervals because of fairly tight priors used.

Also see

[DSGE] [dsge](#) — Linear dynamic stochastic general equilibrium models

[DSGE] [Intro 1](#) — Introduction to DSGEs

[DSGE] [Intro 3a](#) — New Keynesian model

[DSGE] [Intro 3d](#) — Nonlinear New Keynesian model

[BAYES] [bayes: dsge](#) — Bayesian linear dynamic stochastic general equilibrium models

[BAYES] [bayes: dsge postestimation](#) — Postestimation tools for bayes: dsge and bayes: dsge1

[BAYES] [bayes](#) — Bayesian regression models using the bayes prefix

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](#).