

[Description](#)[Remarks and examples](#)[Quick start](#)[Methods and formulas](#)[Menu](#)[Reference](#)[Syntax](#)[Also see](#)[Options](#)

## Description

`bayesfcast compute` produces Bayesian dynamic forecasts of the dependent variables in a model previously fit by `bayes: var`. It creates new variables for prediction results and, if necessary, extends the time frame of the dataset to contain the prediction horizon. Prediction results can be posterior means or medians, posterior standard deviations, and credible intervals.

## Quick start

Posterior means stored in `b_y1`, `b_y2`, and `b_y3` as dynamic forecasts after fitting a model with `bayes: var` for dependent variables `y1`, `y2`, and `y3`

```
bayesfcast compute b_
```

Same as above, but begin forecast on the first quarter of 1979 for 10 periods ahead

```
bayesfcast compute b_, dynamic(q(1979q1)) step(10)
```

Same as above, but requesting posterior medians instead of posterior means be saved as forecasts along with 80% equal-tailed credible intervals.

```
bayesfcast compute b_, dynamic(q(1979q1)) step(10) median clevel(80)
```

## Menu

Statistics > Multivariate time series > Bayesian models > VAR forecasts > Compute forecasts (required for graph)

## Syntax

`bayesfcst_compute prefix [ , options ]`

*prefix* is the prefix appended to the names of the dependent variables to create the names of the variables holding the dynamic forecasts.

| <i>options</i>                                | Description  |
|---|--|
| <code>step(#)</code>                          | set # periods to forecast; default is <code>step(1)</code>                         |
| <code>dynamic(time_constant)</code>           | begin dynamic forecasts at <i>time_constant</i>                                    |
| <code>estimate(estname)</code>                | use previously stored results <i>estname</i> ; default is to use active results    |
| <code>replace</code>                          | replace existing forecast variables that have the same prefix                      |
| <code>clevel(#)</code>                        | set credible interval level; default is <code>clevel(95)</code>                    |
| <code>hpd</code>                              | save HPD credible intervals instead of the default equal-tailed credible intervals |
| <code>median</code>                           | save posterior medians instead of default posterior means and standard deviations  |
| <code>mcmcsaving(filename[ , replace])</code> | save simulation results to <i>filename.dta</i>                                     |
| <code>mcmcsaving</code>                       | save simulation results to <i>prefix_mcmc.dta</i>                                  |
| <code>rseed(#)</code>                         | random-number seed   |

`bayesfcst_compute` can be used only after `bayes: var.`

You must `tsset` your data before using `bayesfcst_compute`; see [TS] [tsset](#).

## Options

`step(#)`, `dynamic(time_constant)`, `estimate(estname)`, and `replace`; see [TS] [fcst compute](#).

`clevel(#)` specifies the credible level, as a percentage, for equal-tailed and HPD credible intervals. The default is `clevel(95)` or as set by [BAYES] [set clevel](#).

`hpd` displays the HPD credible intervals instead of the default equal-tailed credible intervals.

`median` calculates and saves posterior medians instead of the posterior means and standard deviations.

`mcmcsaving(filename[ , replace])` saves simulation results in *filename.dta*. The `replace` option specifies to overwrite *filename.dta* if it exists. If the `mcmcsaving()` option is not specified, simulation results are not saved.

The saved dataset has the following structure. Variable `_chain` records chain identifiers. Variable `_index` records iteration numbers. `bayesfcst` saves only states (sets of values) that are different from one iteration to another and the frequency of each state in variable `_frequency`. As such, `_index` may not necessarily contain consecutive integers. Remember to use `_frequency` as a frequency weight if you need to obtain any summaries of this dataset. Values for each forecasted outcome are saved in a separate variable in the dataset. The variable corresponding to outcome *y* and time period *t* is named as *y\_t*.

`mcmcsaving` saves the simulation results in *prefix\_mcmc.dta*.

`rseed(#)` sets the random-number seed. This option can be used to reproduce results. With one chain, `rseed(#)` is equivalent to typing `set seed #` prior to calling the `bayes` prefix; see [R] [set seed](#). With multiple chains, you should use `rseed()` for reproducibility; see [Reproducing results](#) in [BAYES] [bayesmh](#).

## Remarks and examples

Below, we show examples of dynamic forecasts after fitting Bayesian vector autoregression (VAR) models. Also see [example 9](#) in [BAYES] **bayes: var** for another example.

### ► Example 1

We revisit [example 1](#) from [TS] **fcast compute**. It uses `lutkepohl2.dta` of West Germany microeconomic quarterly data for the years between 1960 and 1978. The example studies the relationships between investment (`dln_inv`), income (`dln_inc`), and consumption (`dln_consump`).

```
. use https://www.stata-press.com/data/r19/lutkepohl2
. tsset
```

First, we fit a Bayesian VAR model with two lags on the dependent variables `dln_inv`, `dln_inc`, and `dln_consump` using the `bayes: var` command with default settings. The output of the command is suppressed. The subsequent `bayesfcast` command requires that the simulation results generated by `bayes: var` be saved in a permanent dataset, in our case `bfcastex1.dta`.

```
. bayes, rseed(17) nomodelsummary notable noheader:
> var dln_inv dln_inc dln_consump if qtr<tq(1979q1)
Burn-in ...
Simulation ...
. bayes, saving(bfcastex1, replace)
note: file bfcastex1.dta not found; file saved.
```

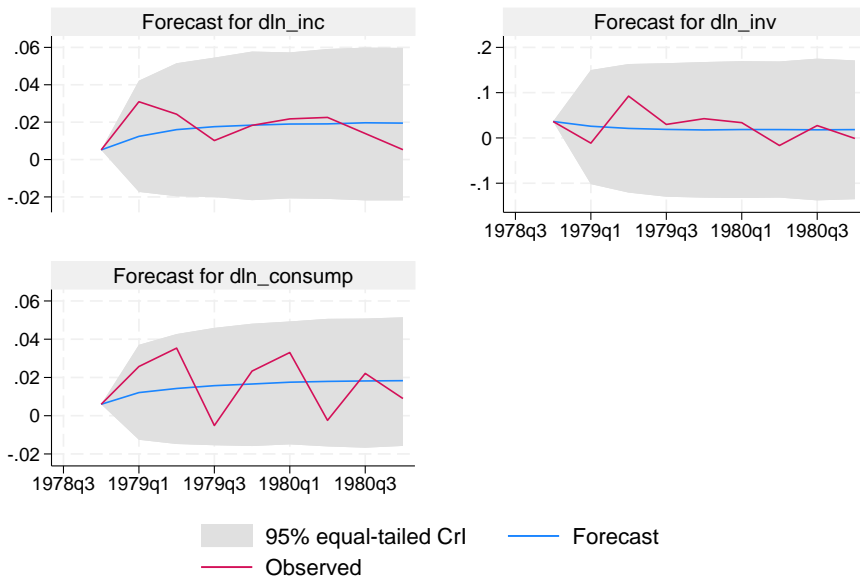
We then compute 8-step dynamic predictions for each of the three dependent variables using `bayesfcast compute`. We specify `b1_` prefix for the newly created variables.

```
. bayesfcast compute b1_, step(8)
```

The prediction results are saved in the current dataset and include posterior means, `b1_*`, posterior standard deviations, `b1_*_sd`, and 95% credible intervals, `b1_*_lb` and `b1_*_ub`. Populated are observations between `qtr = 1978q4` and `qtr = 1980q4`.

Next, using the `bayesfcst graph` command, we show the posterior mean forecasts along with the 95% credible bands.

```
. bayesfcst graph b1_dln_inc b1_dln_inv b1_dln_consump, observed
```



Compared with the original forecasts shown in [example 1](#), the Bayesian posterior means forecasts are much smoother and closer to the stationary state. The variability of the Bayesian forecasts, as measured by the width of the 95% credible bands, tends to increase slightly with time, whereas the width of the confidence bands in the original forecasts stays the same. The Bayesian forecasts thus appear to provide more conservative predictions.

◀

## ► Example 2

Continuing with [example 1](#), we fit a second VAR model in which the Minnesota prior on regression coefficients is more relaxed, thus giving us posterior estimates that are closer to the frequentist ones, as obtained by the `var` command.

We use the same VAR(2) model specification but specify the `selftight(1)` suboption of the `minnconjprior()` option, which controls the Minnesota prior. Again, we suppress the output of the `bayes: var` command and save the simulation results in `bfcstex2.dta`.

```
. bayes, minnconjprior(selftight(1)) rseed(17) nomodelsummary notable noheader:
> var dln_inc dln_consump dln_inv if qtr< tq(1979q1)

Burn-in ...
Simulation ...

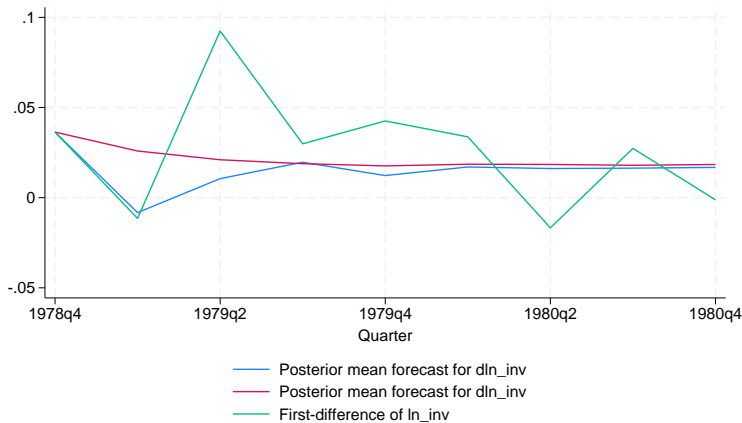
. bayes, saving(bfcstex2, replace)
note: file bfcstex2.dta not found; file saved.
```

Then, we compute Bayesian forecasts based on the second model and save them in the current dataset using the `b2_` prefix for the newly created variables.

```
. bayesfcst compute b2_, step(8)
```

Finally, we plot the forecasts of the two models along with the observed values for the `dln_inv` variable (shown in green).

```
. graph twoway line b2_dln_inv b1_dln_inv dln_inv qtr
> if b2_dln_inv < ., legend(pos(6))
```



The Bayesian forecasts of the second model (shown in blue) are more rugged than those of the first model and are indeed closer to the original forecasts from [example 1](#). Although initially the second model forecasts are closer to the observed, their precision quickly drops afterward, and the more conservative predictions of the first model appear to fare better later on. The forecasts of both models converge with each other after six time steps.



## Methods and formulas

Methods and formulas are presented under the following headings:

*Bayesian dynamic forecasts*  
*Dynamic forecasts after bayes: var*

## Bayesian dynamic forecasts

In the frequentist context, dynamic forecasts are based on one set of point estimates of the model parameter vector  $\theta$ . In a Bayesian framework, instead of point estimates, an entire posterior distribution of  $\theta$  is used to compute forecasts.

Let  $\mathbf{y}_t$  be a vector of outcome variables at time  $t$ . A dynamic forecast with horizon  $h$  is a realization of future observations  $\mathbf{y}_{T+1}, \mathbf{y}_{T+2}, \dots, \mathbf{y}_{T+h}$  based on the observations up to time  $T$ :  $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_T$ . Let  $f(\mathbf{y}_{T+1:T+h}|\mathbf{y}_{1:T}, \boldsymbol{\theta})$  be the distribution of future observations conditional on the observations up to time  $T$ . Bayesian dynamic forecasts are drawn from the posterior predictive distribution

$$p(\mathbf{y}_{T+1:T+h}|\mathbf{y}_{1:T}) = \int f(\mathbf{y}_{T+1:T+h}|\mathbf{y}_{1:T}, \boldsymbol{\theta})p(\boldsymbol{\theta}|D)d\boldsymbol{\theta}$$

where  $p(\boldsymbol{\theta}|D)$  is the posterior distribution of the model with respect to some data  $D$ . For example,  $D$  may include the outcome observations  $\mathbf{y}_{1:T}$  along with observations on exogenous variables  $\mathbf{X}_{1:T}$ .

In practice,  $f(\mathbf{y}_{T+1:T+h}|\mathbf{y}_{1:T}, \boldsymbol{\theta})$  is computed recursively using the factorization

$$f(\mathbf{y}_{T+1:T+h}|\mathbf{y}_{1:T}, \boldsymbol{\theta}) = f(\mathbf{y}_{T+1}|\mathbf{y}_{1:T}, \boldsymbol{\theta})f(\mathbf{y}_{T+2}|\mathbf{y}_{1:T+1}, \boldsymbol{\theta}) \dots f(\mathbf{y}_{T+h}|\mathbf{y}_{1:T+h-1}, \boldsymbol{\theta})$$

After fitting a time-series model using `bayes: var`, we have a Markov chain Monte Carlo (MCMC) sample of realizations of  $\boldsymbol{\theta}$  from its posterior distribution  $p(\boldsymbol{\theta}|D)$ . To simulate the distribution of the dynamic forecast  $p(\mathbf{y}_{T+1:T+h}|\mathbf{y}_{1:T})$ , we recycle this same MCMC sample.

1. For each draw  $\boldsymbol{\theta}^s$  from the MCMC sample  $\{\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \dots, \boldsymbol{\theta}^M\}$ , repeat the following:
  - 2.1. Generate  $\tilde{\mathbf{y}}_{T+1}^s$  from  $f(\mathbf{y}_{T+1}|\mathbf{y}_{1:T}, \boldsymbol{\theta}^s)$ .
  - 2.2. Generate  $\tilde{\mathbf{y}}_{T+2}^s$  from  $f(\mathbf{y}_{T+2}|\mathbf{y}_{1:T}, \tilde{\mathbf{y}}_{T+1}^s, \boldsymbol{\theta}^s)$ .
  - ...
  - 2.h. Generate  $\tilde{\mathbf{y}}_{T+h}^s$  from  $f(\mathbf{y}_{T+h}|\mathbf{y}_{1:T}, \tilde{\mathbf{y}}_{T+1:T+h-1}^s, \boldsymbol{\theta}^s)$ .
3. Save the forecast draw  $(\tilde{\mathbf{y}}_{T+1}^s, \tilde{\mathbf{y}}_{T+2}^s, \dots, \tilde{\mathbf{y}}_{T+h}^s)$ .
4. Using the simulated forecast draws, compute posterior summaries such as means, medians, and credible intervals for each of the  $h$  forecast steps  $\mathbf{y}_{T+1}, \dots, \mathbf{y}_{T+h}$ .

The `bayesfcst compute` command saves the estimated posterior summaries computed by the above algorithm in the current dataset similar to the way the `fcast compute` command computes and saves forecast point estimates.

## Dynamic forecasts after bayes: var

The method of simulating Bayesian dynamic forecasts in the special case of VAR is proposed in [Karls-son \(2013\)](#). It follows the steps of the above general algorithm for simulating an MCMC sample of forecasts from the posterior predictive distribution  $p(\mathbf{y}_{T+1:T+h}|\mathbf{y}_{1:T})$ .

Let's consider a VAR( $p$ ) model using the notation from [Methods and formulas](#) of [\[BAYES\] bayes: var](#):

$$\mathbf{y}_t = \mathbf{A}_1\mathbf{y}_{t-1} + \dots + \mathbf{A}_p\mathbf{y}_{t-p} + \mathbf{C}\mathbf{x}_t + \mathbf{u}_t$$

for  $t = 1, \dots, T$ .

The model parameter vector  $\theta$  includes elements of  $\mathbf{A}_1, \dots, \mathbf{A}_p, \mathbf{C}$ , and error covariance matrix  $\Sigma$  of the error terms  $\mathbf{u}_t$ 's. For each MCMC draw of the parameters  $\theta^s$ , the forecast steps 2.1 to 2.h reduce to the following,

$$\begin{aligned}\tilde{\mathbf{y}}_{T+1}^s &= \mathbf{A}_1^s \mathbf{y}_T + \mathbf{A}_2^s \mathbf{y}_{T-1} \cdots + \mathbf{A}_p^s \mathbf{y}_{T-p+1} + \mathbf{C}^s \mathbf{x}_{T+1} + \tilde{\mathbf{u}}_{T+1}^s \\ \tilde{\mathbf{y}}_{T+2}^s &= \mathbf{A}_1^s \tilde{\mathbf{y}}_{T+1}^s + \mathbf{A}_2^s \mathbf{y}_T + \cdots + \mathbf{A}_p^s \mathbf{y}_{T-p} + \mathbf{C}^s \mathbf{x}_{T+2} + \tilde{\mathbf{u}}_{T+2}^s \\ &\vdots \\ \tilde{\mathbf{y}}_{T+h}^s &= \mathbf{A}_1^s \tilde{\mathbf{y}}_{T+h-1}^s + \mathbf{A}_2^s \tilde{\mathbf{y}}_{T+h-2}^s + \cdots + \mathbf{C}^s \mathbf{x}_{T+h} + \tilde{\mathbf{u}}_{T+h}^s\end{aligned}$$

where  $\tilde{\mathbf{u}}_{T+1}^s, \dots, \tilde{\mathbf{u}}_{T+h}^s$  are independent draws from  $N(\mathbf{0}, \Sigma^s)$ .

## Reference

Karlsson, S. 2013. “Forecasting with Bayesian vector autoregression”. In *Handbook of Economic Forecasting*, edited by G. Elliott and A. Timmermann, vol. 2B: 791–897. Amsterdam: North-Holland. <https://doi.org/10.1016/B978-0-444-62731-5.00015-4>.

## Also see

[TS] **fcst compute** — Compute dynamic forecasts

[BAYES] **bayesfcst graph** — Graphs of Bayesian dynamic forecasts

[BAYES] **bayes: var** — Bayesian vector autoregressive models

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 © 1985–2025 StataCorp LLC, College Station, TX, USA. All rights reserved.

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

