# Title



**bayesvarstable** — Check the stability condition of Bayesian VAR estimates



# Description

`bayesvarstable` checks the eigenvalue stability condition after fitting Bayesian vector autoregression (VAR) by using `bayes: var`.

# Quick start

Checking eigenvalue stability condition after `bayes: var`

```
bayesvarstable
```

Same as above, but compute 80% highest posterior density (HPD) credible intervals instead of 95% equal-tailed credible intervals

```
bayesvarstable, hpd clevel(80)
```

# Menu

Statistics > Multivariate time series > Bayesian models > Check stability condition of VAR estimates

# Syntax

`bayesvarstable` [ , *options* ]

| *options* | Description |
|---|---|
| `estimates(`*estname*`)` | use previously stored results *estname*; default is to use active results |
| `clevel(`*#*`)` | set credible interval level; default is `clevel(95)` |
| `hpd` | save HPD credible intervals instead of the default equal-tailed credible intervals |
| `mcmcsaving(`*filename* [ , `replace` ]`)` | save simulation results to *filename*`.dta` |

`collect` is allowed; see [U] **11.1.10 Prefix commands**.

# Options

`estimates(`*estname*`)` requests that `bayesvarstable` use the previously obtained set of `bayes: var` estimates stored as *estname*. By default, `bayesvarstable` uses the active estimation results. See [R] **estimates** for information on manipulating estimation results.

`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.

`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. `bayesvarstable` 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 modulus of each eigenvalue are saved in a separate variable in the dataset.

# Remarks and examples

stata.com

Stability is an important condition for VAR model interpretation; see *Remarks and examples* of [TS] **varstable**. If the stability condition of a VAR model is not met, its impulse–response functions (IRFs) and forecast-error variance decompositions do not reach equilibrium and thus do not have clear interpretation.

Lütkepohl (2005) and Hamilton (1994) show that if the modulus of each eigenvalue of the companion matrix $\mathbf{A}$ is strictly less than one, the estimated VAR is stable (see *Methods and formulas* for the definition of the matrix $\mathbf{A}$). In a Bayesian setting, we are concerned with the posterior distribution of $\mathbf{A}$ and its eigenvalues.

Following are two examples illustrating stable and unstable VAR models.

## ▷ Example 1: Stable VAR model

We revisit example 1 from [TS] **varstable**. 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/r18/lutkepohl2
. tsset
```

Using the `bayes: var` command, we fit a Bayesian VAR model with two lags on the dependent variables `dln_inv`, `dln_inc`, and `dln_consump`. Considered are observations between the second quarter of 1961 and the fourth quarter of 1978. We use the default conjugate Minnesota prior for regression coefficients and error covariance matrix.

```
. bayes, rseed(17) nomodelsummary:
> var dln_inv dln_inc dln_consump if qtr>=tq(1961q2) & qtr<=tq(1978q4)
Burn-in ...
Simulation ...

Bayesian vector autoregression                  MCMC iterations  =     12,500
Gibbs sampling                                  Burn-in          =      2,500
                                                MCMC sample size =     10,000
Sample: 1961q2 thru 1978q4                      Number of obs    =         71
                                                Acceptance rate  =          1
                                                Efficiency:  min =      .9556
                                                             avg =      .9962
Log marginal-likelihood =  467.75286                         max =          1

------------------------------------------------------------------------------
             |                                                Equal-tailed
             |      Mean   Std. dev.     MCSE     Median  [95% cred. interval]
-------------+----------------------------------------------------------------
dln_inv      |
     dln_inv |
         L1. |  .4749526   .1046821   .001071   .4762824   .2706787   .6790291
         L2. |  .0062935    .063174   .000632   .0058376  -.1181113    .129959
             |
     dln_inc |
         L1. |  .1150521   .4145854   .004146   .1155755  -.7122031   .9358321
         L2. |  .0096558   .2461088   .002464   .0129206  -.4780951    .490937
             |
 dln_consump |
         L1. | -.0693822   .4910385   .004828  -.0712677  -1.016477   .9050535
         L2. |  .0182113   .2919327   .002919   .0169657  -.5563898   .6010627
             |
       _cons |  .0067839   .0153897   .000154   .0067986  -.0233363   .0367596
-------------+----------------------------------------------------------------
```

```
dln_inc      |
     dln_inv |
         L1. |   .0152113   .0248328   .000248   .0154024  -.0341219   .0635173
         L2. |    .000957   .0149204   .000147   .0010833  -.0285813   .0306545
             |
     dln_inc |
         L1. |    .600281   .0981275   .000981   .5997577   .4077653   .7928394
         L2. |    .011757   .0577031   .000577   .0123101  -.1009659   .1245041
             |
 dln_consump |
         L1. |  -.0331359   .1151265   .001151  -.0318916  -.2594495   .1939938
         L2. |  -.0266197   .0694851   .000695  -.0263958  -.1637059   .1123704
             |
       _cons |   .0084678   .0036265   .000037   .0084371   .0013034   .0155666
-------------+----------------------------------------------------------------
dln_consump  |
     dln_inv |
         L1. |  -.0183312   .0220482    .00022  -.0182937   -.062597   .0243933
         L2. |   .0092806   .0135179   .000135   .0094044  -.0171007    .036166
             |
     dln_inc |
         L1. |  -.0365965   .0875614   .000876  -.0368425  -.2086565   .1364804
         L2. |   .0345945   .0520216   .000514   .0339648  -.0668323    .136918
             |
 dln_consump |
         L1. |   .5444814   .1030406   .001027   .5432019   .3416401   .7489821
         L2. |   .0555939   .0617942   .000618    .055126   -.063175   .1763757
             |
       _cons |   .0078414   .0032597   .000033   .0078245    .001402   .0141132
-------------+----------------------------------------------------------------
   Sigma_1_1 |    .003945   .0006693   6.4e-06   .0038783   .0028446   .0054382
   Sigma_2_1 |  -.0000314   .0001118   1.1e-06  -.0000291  -.0002548   .0001897
   Sigma_3_1 |    .000138   .0001007   1.0e-06   .0001355  -.0000512   .0003478
   Sigma_2_2 |   .0002195   .0000373   3.7e-07   .0002158   .0001579   .0003039
   Sigma_3_2 |   .0000502   .0000238   2.4e-07    .000049   6.46e-06   .0001007
   Sigma_3_3 |   .0001743   .0000294   2.9e-07   .0001714   .0001261   .0002408
------------------------------------------------------------------------------
```

For explanation of the output of `bayes: var`, see *Remarks and examples* of [BAYES] **bayes: var**.

To use the `bayesvarstable` command, we need to save simulation results computed by `bayes: var` in a permanent dataset.

```
. bayes, saving(bvarex1)
note: file bvarex1.dta saved.
```

Now we are ready to check the stability condition for the above Bayesian model.

```
. bayesvarstable
Eigenvalue stability condition                  Companion matrix size =      6
                                                MCMC sample size      =  10000

------------------------------------------------------------------------------
  Eigenvalue |                                                Equal-tailed
     modulus |       Mean   Std. dev.     MCSE     Median  [95% cred. interval]
-------------+----------------------------------------------------------------
           1 |   .7295294   .0952871   .000953   .7272906    .547312   .9209245
           2 |   .6039037   .1045099   .001045   .6094994   .3810883   .7904044
           3 |    .428933   .1272649   .001273   .4239249   .2113325   .6645651
           4 |   .2126552   .0780213    .00078   .1997342   .0900884   .3846134
           5 |   .1378018   .0565196   .000565   .1349177   .0385605   .2577174
           6 |   .0759403     .05052   .000505   .0700686   .0035577   .1847619
------------------------------------------------------------------------------
Pr(eigenvalues lie inside the unit circle) = 0.9966
```

The VAR model has a companion matrix of size 6 (3 response variables times 2 lags). The `bayesvarstable` command thus reports posterior summaries for the moduli of 6 eigenvalues. The maximum one has a posterior mean of 0.73, less than 1. In addition to posterior means, we also see posterior standard deviations, MCMC standard errors, medians, and credible intervals.

The `bayesvarstable` command estimates the probability of unit circle inclusion for all eigenvalues to be 0.9966, or essentially 1. The stability condition is thus satisfied.

We may specify the HPD credible intervals instead of the default equal-tailed ones and change the level of the intervals. This, however, would not change the estimated probability of inclusion and the overall conclusion.

```
. bayesvarstable, hpd clevel(80)
Eigenvalue stability condition                  Companion matrix size =     6
                                                MCMC sample size      = 10000
```

| Eigenvalue modulus | Mean | Std. dev. | MCSE | Median | HPD [80% cred. | interval] |
|---|---|---|---|---|---|---|
| 1 | .7295294 | .0952871 | .000953 | .7272906 | .6066106 | .8490679 |
| 2 | .6039037 | .1045099 | .001045 | .6094994 | .4782224 | .7449145 |
| 3 | .428933 | .1272649 | .001273 | .4239249 | .2656266 | .6001815 |
| 4 | .2126552 | .0780213 | .00078 | .1997342 | .1065876 | .3036596 |
| 5 | .1378018 | .0565196 | .000565 | .1349177 | .0623463 | .2060198 |
| 6 | .0759403 | .05052 | .000505 | .0700686 | .0000169 | .1200219 |

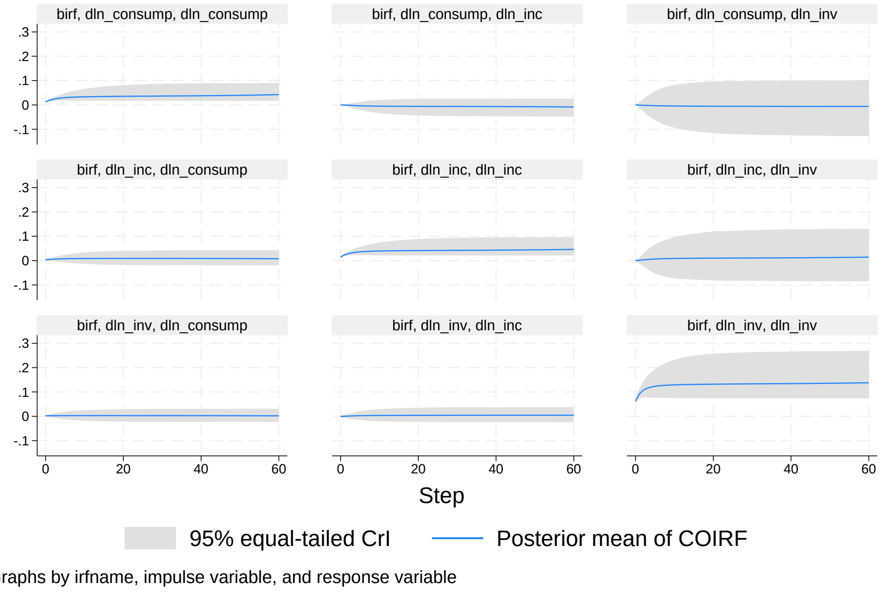
```
Pr(eigenvalues lie inside the unit circle) = 0.9966
```

As we mentioned above, a stable VAR model has IRFs that reach equilibrium in the long run. Let's verify this. We compute IRFs for 60 quarters (15 years) ahead and save them as `birf` estimates in `birfex1.irf`.

```
. bayesirf create birf, step(60) set(birfex1)
(file birfex1.irf created)
(file birfex1.irf now active)
(file birfex1.irf updated)
```

See *Remarks and examples* for details about [BAYES] **bayesirf create**. We check the long-term behavior of the cumulative orthogonalized IRFs using the `bayesirf graph` command.

```
. bayesirf graph coirf
```




In particular, we look at the cumulative shock effects of impulse variables on themselves (the graphs on the diagonal). It is clear that all shocks reach long-term equilibrium after about 2 years (all graphs converge to horizontal asymptotes). These are the types of graphs we expect to see from a stable VAR model.

◁

## ▷ Example 2: Unstable VAR model

In this example, we show how the specification of a strong prior may violate the stability condition of a VAR model.

We consider the same VAR model as in the previous example, but now we reduce the number of lags from 2 to 1 and strengthen the default Minnesota prior. In particular, we change the `selftight()` suboption of `minnconjprior()` from its default value of 0.1 to 0.001. This option determines the prior variance of regression coefficients; see *self-variables tightness parameter*. A value of 0.001 will shrink the regression coefficients to their prior mean values, which are 1 for self-variables first-lag coefficients and 0 otherwise. The shrinkage is thus toward a *random-walk* behavior, which is known to be unstable. Given the modest sample size of 90 observations, we expect the prior to dominate the information available in the data.

```
. bayes, minnconjprior(selftight(0.001)) rseed(17) saving(bvarex2) nomodelsummary:
> var dln_inv dln_inc dln_consump, lags(1)
Burn-in ...
Simulation ...
Bayesian vector autoregression                   MCMC iterations  =     12,500
Gibbs sampling                                   Burn-in          =      2,500
                                                 MCMC sample size =     10,000
Sample: 1960q3 thru 1982q4                       Number of obs    =         90
                                                 Acceptance rate  =          1
                                                 Efficiency:  min =      .9779
                                                              avg =      .9988
Log marginal-likelihood =   590.1324                          max =          1

                                                             Equal-tailed
                     Mean   Std. dev.     MCSE     Median  [95% cred. interval]

dln_inv
     dln_inv
         L1.     .9999075   .0015466   .000015    .9999005   .9968787   1.002993

     dln_inc
         L1.    -.0001024   .0057483   .000056   -.0000905  -.0113532   .0112184

 dln_consump
         L1.     .0000347   .0062553   .000063    .0000484  -.0122773   .0124097

       _cons    -.0000218   .0050112    .00005   -4.42e-06   -.009838   .0097902

dln_inc
     dln_inv
         L1.     5.87e-06   .0003621   3.6e-06    6.24e-06  -.0006947    .000714

     dln_inc
         L1.     .9999134   .0013413   .000013    .9999053   .9973009   1.002532

 dln_consump
         L1.    -.0000275   .0014526   .000015   -.0000321  -.0028922   .0028222

       _cons    -.0001133   .0011346   .000011   -.0001213   -.002378   .0021637

dln_consump
     dln_inv
         L1.    -7.21e-06   .0003546   3.5e-06   -8.11e-06  -.0007066   .0006912

     dln_inc
         L1.    -.0000405    .001341   .000014   -.0000284  -.0027065    .002576

 dln_consump
         L1.     .9998961   .0014424   .000014    .9999152   .9970457   1.002728

       _cons     -.000031   .0011446   .000011   -.0000338  -.0022769   .0022331

   Sigma_1_1      .004672   .0006967   7.0e-06    .0046044   .0034928     .00625
   Sigma_2_1    -.0000808   .0001147   1.1e-06   -.0000799  -.0003115   .0001442
   Sigma_3_1     .0002439   .0001158   1.2e-06      .00024   .0000266   .0004826
   Sigma_2_2     .0002519   .0000382   3.8e-07    .0002482   .0001879    .000336
   Sigma_3_2      .000067   .0000275   2.8e-07    .0000655   .0000169   .0001243
   Sigma_3_3     .0002483   .0000366   3.7e-07    .0002447   .0001869   .0003288

file bvarex2.dta saved.
```

The posterior mean estimates of regression coefficients are very close to their prior mean values.

We use `bayesvarstable` to check the stability condition.

```
. bayesvarstable
Eigenvalue stability condition                    Companion matrix size =     3
                                                  MCMC sample size      = 10000
```

| Eigenvalue modulus | Mean | Std. dev. | MCSE | Median | Equal-tailed [95% cred. | interval] |
|---|---|---|---|---|---|---|
| 1 | 1.001409 | .0012333 | .000012 | 1.001324 | .999263 | 1.004065 |
| 2 | .9998958 | .0011205 | .000011 | .9998891 | .997711 | 1.002059 |
| 3 | .9984138 | .0012513 | .000013 | .998492 | .9957189 | 1.000603 |

```
Pr(eigenvalues lie inside the unit circle) = 0.1194
```

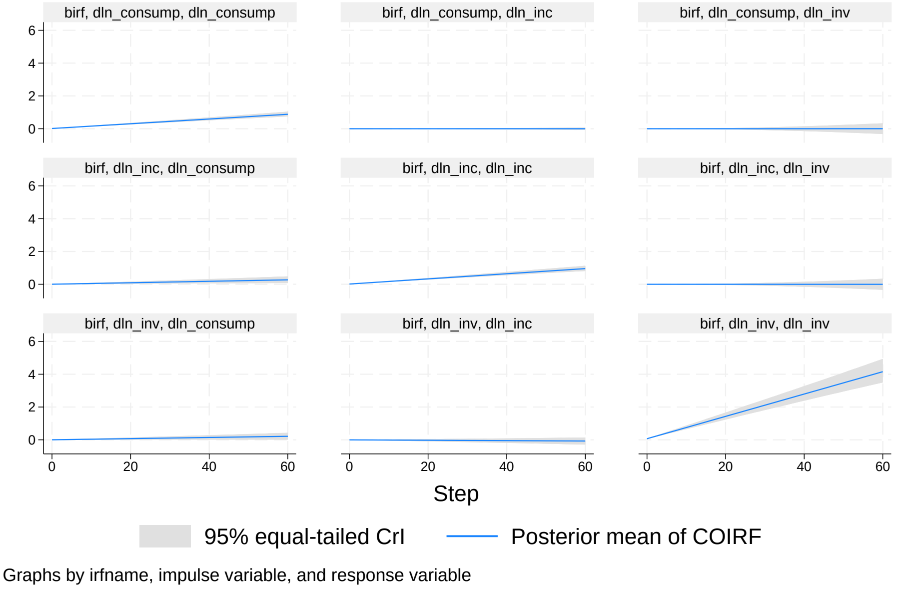
The reported probability that all three eigenvalues lie in the unit circle is only about 12% and is clearly insufficient to claim the stability of the estimates.

We can also look at IRFs for visual confirmation of the instability of the model. We compute IRFs for 60 quarters ahead and save them as `birf` estimates in `birfex2.irf`.

```
. bayesirf create birf, step(60) set(birfex2)
(file birfex2.irf created)
(file birfex2.irf now active)
(file birfex2.irf updated)
```

Then we plot the cumulative orthogonalized IRFs using `bayesirf graph`.

```
. bayesirf graph coirf
```




It is clear that the shocks of impulses on themselves (the graphs on the diagonal) do not reach equilibrium and continue to increase beyond the 60-quarter period. This is a typical behavior of an unstable VAR model.

This particular instability problem arises because the used prior strongly favors an unstable, random-walk model, and there is not enough information in the data to outweigh this prior. For instance, if we specified zero prior means for all coefficients, we would not run into this problem. What constitutes a strong prior depends on the sample size and the amount of information contained in the data about model parameters. The conclusion in this example may not hold for other VAR models and datasets. We thus recommend checking the stability condition after fitting any VAR model before proceeding with postestimation analysis.

◁

## Stored results

`bayesvarstable` stores the following in `r()`:

| | |
|---|---|
| Scalars | |
| `r(prob_incl)` | probability of unit circle inclusion of all eigenvalues |
| `r(mcmcsize)` | MCMC sample size |
| `r(compsize)` | companion matrix size |
| Matrices | |
| `r(summary)` | matrix with posterior summary statistics for eigenvalues |

## Methods and formulas

Consider a companion matrix $\mathbf{A}$ defined in *Methods and formulas* of [TS] **varstable**. In a Bayesian setting, $\mathbf{A}$ is a random matrix with a posterior distribution that depends on the prior distribution of regression coefficients and error covariance matrix. The Bayesian computations use the MCMC sample created by the `bayes: var` command that contains draws from the posterior distribution of the regression coefficients/matrices and error covariance.

For each draw, the eigenvalue moduli of the companion matrix $\mathbf{A}^*$ that corresponds to that draw are computed and saved in an MCMC sample. Finally, the resulting MCMC samples of eigenvalue moduli are summarized, and standard Bayesian statistics such as posterior mean, medians, and credible intervals are reported.

The posterior probability of the unit circle inclusion is estimated as the proportion of MCMC observations for which all eigenvalues of $\mathbf{A}^*$'s are strictly within the unit circle.

## References


Hamilton, J. D. 1994. *Time Series Analysis*. Princeton, NJ: Princeton University Press.

Lütkepohl, H. 2005. *New Introduction to Multiple Time Series Analysis*. New York: Springer.


## Also see

[BAYES] **bayes: var postestimation** — Postestimation tools for bayes: var

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

[TS] **varstable** — Check the stability condition of VAR or SVAR estimates