Title

bayestest interval — Interval hypothesis testing

Description

`bayestest interval` performs interval hypothesis tests for model parameters and functions of model parameters using current estimation results from the `bayesmh` command. `bayestest interval` reports mean estimates, standard deviations, and MCMC standard errors of posterior probabilities associated with an interval hypothesis.

Quick start

Posterior probability of the hypothesis that 45 < `{y:_cons}` < 50

```
bayestest interval {y:_cons}, lower(45) upper(50)
```

As above, but skip every 5 observations from the full MCMC sample

```
bayestest interval {y:_cons}, lower(45) upper(50) skip(5)
```

Posterior probability of a hypothesis about a function of model parameter `{y:x1}`

```
bayestest interval (OR:exp({y:x1})), lower(1.1) upper(1.5)
```

Posterior probability of hypotheses 45 < `{y:_cons}` < 50 and 0 < `{var}` < 10 tested independently

```
bayestest interval ({y:_cons}, lower(45) upper(50))  ///
     ({var}, lower(0) upper(10))
```

As above, but tested jointly

```
bayestest interval (({y:_cons}, lower(45) upper(50))  ///
     ({var}, lower(0) upper(10)), joint)
```

Posterior probability of the hypothesis `{mean}` = 2 for discrete parameter `{mean}`

```
bayestest interval ({mean}==2)
```

Posterior probability of the interval hypothesis 0 ≤ `{mean}` ≤ 4

```
bayestest interval {mean}, lower(0, inclusive) upper(4, inclusive)
```

Menu

Statistics > Bayesian analysis > Interval hypothesis testing

Syntax

Test one interval hypothesis about continuous or discrete parameter

`bayestest interval` *exspec* [, *luspec options*]

Test one point hypothesis about discrete parameter

`bayestest interval` *exspec*`==`*#* [, *options*]

Test multiple hypotheses separately

`bayestest interval (`*testspec*`)` [`(`*testspec*`)` ...] [, *options*]

Test multiple hypotheses jointly

`bayestest interval (`*jointspec*`)` [, *options*]

Full syntax

`bayestest interval (`*spec*`)` [`(`*spec*`)` ...] [, *options*]

exspec is optionally labeled expression of model parameters, [*prlabel*`:`]*expr*, where *prlabel* is a valid Stata name (or `prob#` by default), and *expr* is a scalar model parameter or scalar expression (parentheses are optional) containing scalar model parameters. The expression *expr* may not contain variable names.

testspec is *exspec*[, *luspec*] or *exspec*`==`*#* for discrete parameters only.

jointspec is [*prlabel:*] `(`*testspec*`) (`*testspec*`)` ... `, joint`. The labels (if any) of *testspec* are ignored.

spec is one of *testspec* or *jointspec*.

luspec	Null hypothesis
`lower(`*#*`)` [`upper(.)`]	$\theta > \#$
`lower(`*#*`, inclusive)` [`upper(.)`]	$\theta \geq \#$
[`lower(.)`] `upper(`*#*`)`	$\theta < \#$
[`lower(.)`] `upper(`*#*`, inclusive)`	$\theta \leq \#$
`lower(`$\#_l$`) upper(`$\#_u$`)`	$\#_l < \theta < \#_u$
`lower(`$\#_l$`) upper(`$\#_u$`, inclusive)`	$\#_l < \theta \leq \#_u$
`lower(`$\#_l$`, inclusive) upper(`$\#_u$`)`	$\#_l \leq \theta < \#_u$
`lower(`$\#_l$`, inclusive) upper(`$\#_u$`, inclusive)`	$\#_l \leq \theta \leq \#_u$

`lower(`*intspec*`)` and `upper(`*intspec*`)` specify the lower- and upper-interval values, respectively.

intspec is *#* [, `inclusive`]

where *#* is the interval value, and suboption `inclusive` specifies that this value should be included in the interval, meaning a closed interval. Closed intervals make sense only for discrete parameters.

intspec may also contain a dot (`.`), meaning negative infinity for `lower()` and positive infinity for `upper()`. Either option `lower(.)` or option `upper(.)` must be specified.

options	Description
Main	
skip(#)	skip every # observations from the MCMC sample; default is skip(0)
nolegend	suppress table legend
Advanced	
corrlag(#)	specify maximum autocorrelation lag; default varies
corrtol(#)	specify autocorrelation tolerance; default is corrtol(0.01)

Options

Main

skip(#) specifies that every # observations from the MCMC sample not be used for computation. The default is skip(0) or to use all observations in the MCMC sample. Option skip() can be used to subsample or thin the chain. skip(#) is equivalent to a thinning interval of #+1. For example, if you specify skip(1), corresponding to the thinning interval of 2, the command will skip every other observation in the sample and will use only observations 1, 3, 5, and so on in the computation. If you specify skip(2), corresponding to the thinning interval of 3, the command will skip every 2 observations in the sample and will use only observations 1, 4, 7, and so on in the computation. skip() does not thin the chain in the sense of physically removing observations from the sample, as is done by bayesmh's thinning() option. It only discards selected observations from the computation and leaves the original sample unmodified.

nolegend suppresses the display of the table legend. The table legend identifies the rows of the table with the expressions they represent.

Advanced

corrlag(#) specifies the maximum autocorrelation lag used for calculating effective sample sizes. The default is $\min\{500, \texttt{mcmcsize()}/2\}$. The total autocorrelation is computed as the sum of all lag-k autocorrelation values for k from 0 to either corrlag() or the index at which the autocorrelation becomes less than corrtol() if the latter is less than corrlag().

corrtol(#) specifies the autocorrelation tolerance used for calculating effective sample sizes. The default is corrtol(0.01). For a given model parameter, if the absolute value of the lag-k autocorrelation is less than corrtol(), then all autocorrelation lags beyond the kth lag are discarded.

Remarks and examples

stata.com

Remarks are presented under the following headings:

Introduction

In this entry, we describe interval hypothesis testing, the goal of which is to estimate the probability that a model parameter lies in a certain interval. Interval hypothesis testing is inversely related to credible intervals. For example, if we have a 95% credible interval for θ with endpoints U and L, then the probability of a hypothesis H_0: $\theta \in [U, L]$ is 0.95. For hypothesis testing using model posterior probabilities, see [BAYES] **bayestest model**.

In frequentist hypothesis testing, we often consider a point hypothesis such as H_0: $\theta = \theta_0$ versus $H_a: \theta \neq \theta_0$. In Bayesian hypothesis testing, the probability $P(\theta = \theta_0)$ is 0 whenever θ has a continuous posterior distribution. A point hypothesis is relevant only to parameters with discrete posterior distributions. For continuous parameters, all hypotheses should be formulated as intervals. One possibility is to consider an interval hypothesis H_0: $\theta \in (\theta_0 - \epsilon, \theta_0 + \epsilon)$, where ϵ is some small value.

Note that Bayesian hypothesis testing does not really need a distinction between the null and alternative hypotheses, in the sense that they are defined in a frequentist statistic. There is no need to "protect" the null hypothesis: if $P\{H_0\text{: } \theta \in (a, b)\} = p$, then $P\{H_a\text{: } \theta \notin (a, b)\} = 1 - p$. In what follows, when we refer to H_0, we imply a hypothesis of interest H_0: $\theta \in \Theta$, and when we refer to H_a, we imply the complement hypothesis H_a: $\theta \in \Theta^c$, where Θ is a set of points from the domain of θ and Θ^c is its complement.

The `bayestest interval` command estimates the posterior probability of a null interval hypothesis H_0 using the simulated posterior distributions of model parameters produced by `bayesmh`. Essentially, `bayestest interval` reports posterior summaries for a dichotomous expression that represents H_0.

For example, suppose we would like to test the following hypothesis: H_0: $\theta \in (a, b)$. Then,

```
bayestest interval ({theta}, lower(a) upper(b))
```

is equivalent to

```
bayesstats summary ({theta} > a & {theta} < b)
```

`bayestest interval` reports the estimated posterior mean probability for H_0, which is not a p-value—as reported by classical frequentist tests—used to decide whether to reject H_0 in favor of the alternative H_a. The p-value interpretation is based on the dichotomous problem formulation of H_0 versus H_a, assuming that one of these two alternatives is actually true. The answer in the Bayesian context is a probability statement about θ that is free of any deterministic presumptions. For example, if you estimate $P(H_0)$ to be 0.15, you cannot ask whether this value is significant or whether you can reject the null hypothesis. Bayesian interpretation of this probability is that if you draw θ from the specified prior distribution and update your knowledge about θ based on the observed data, then there is a 15% chance that θ will belong to the interval (a, b). So the conclusion of Bayesian hypothesis testing is not an acceptance or rejection of the null hypothesis but an explicit probability statement about the tested hypothesis.

Interval tests for continuous parameters

Let's continue our analysis of auto.dta from example 4 in [BAYES] **bayesmh** using the mean-only normal model for mpg with a noninformative prior.

```
. use http://www.stata-press.com/data/r14/auto
(1978 Automobile Data)
. set seed 14
. bayesmh mpg, likelihood(normal({var}))
> prior({mpg:_cons}, flat) prior({var}, jeffreys)
Burn-in ...
Simulation ...
Model summary
------------------------------------------------------------------------------
Likelihood:
  mpg ~ normal({mpg:_cons},{var})
Priors:
  {mpg:_cons} ~ 1 (flat)
        {var} ~ jeffreys
------------------------------------------------------------------------------

Bayesian normal regression                       MCMC iterations  =     12,500
Random-walk Metropolis-Hastings sampling         Burn-in          =      2,500
                                                 MCMC sample size =     10,000
                                                 Number of obs    =         74
                                                 Acceptance rate  =      .2668
                                                 Efficiency:  min =     .09718
                                                              avg =      .1021
Log marginal likelihood =   -234.645                          max =      .1071
```

	Mean	Std. Dev.	MCSE	Median	Equal-tailed [95% Cred.	Interval]
mpg						
_cons	21.29222	.6828864	.021906	21.27898	19.99152	22.61904
var	34.76572	5.91534	.180754	34.18391	24.9129	47.61286

▷ Example 1: Interval hypothesis and credible intervals

In the introduction, we commented on the inverse relationship that exists between interval hypothesis tests and credible intervals. Let's verify this using bayestest interval. We are interested in a hypothesis H_0: {mpg:_cons} $\in (19.992, 22.619)$, where the specified numbers are the endpoints of the credible interval for {mpg:_cons} from the bayesmh output. To compute the posterior probability for this hypothesis, we specify the parameter following the command line and specify interval endpoints in lower() and upper().

```
. bayestest interval {mpg:_cons}, lower(19.992) upper(22.619)
Interval tests     MCMC sample size =    10,000
       prob1 : 19.992 < {mpg:_cons} < 22.619
```

	Mean	Std. Dev.	MCSE
prob1	.9496	0.21878	.0053652

The estimated posterior probability is close to 0.95, as we expected, because we used the endpoints of the 95% credible intervals for {mpg:_cons}.

By default, `bayestest interval` labels probabilities as `prob#` (`prob1` in our example). You can specify your own label as long as you enclose the parameter in parentheses:

```
. bayestest interval (mean:{mpg:_cons}), lower(19.992) upper(22.619)
Interval tests     MCMC sample size =     10,000
        mean : 19.992 < {mpg:_cons} < 22.619
```

	Mean	Std. Dev.	MCSE
mean	.9496	0.21878	.0053652

◁

▷ Example 2: Testing multiple hypotheses separately

Continuing example 1, we can verify that the probability associated with the credible interval for `{var}` is also close to 0.95.

We can specify multiple hypotheses with `bayestest interval`, but we must enclose them in parentheses.

```
. bayestest interval ({mpg:_cons}, lower(19.992) upper(22.619))
>                    ({var}, lower(24.913) upper(47.613))
Interval tests     MCMC sample size =     10,000
       prob1 : 19.992 < {mpg:_cons} < 22.619
       prob2 : 24.913 < {var} < 47.613
```

	Mean	Std. Dev.	MCSE
prob1	.9496	0.21878	.0053652
prob2	.9502	0.21754	.0053011

The estimated posterior probability `prob2` is also close to 0.95.

◁

▷ Example 3: Testing multiple hypotheses jointly

We can perform joint tests of multiple hypotheses by enclosing hypothesis to be tested jointly in parentheses and by specifying suboption `joint`. Notice that each individual hypothesis must also be enclosed in parentheses.

```
. bayestest interval (({mpg:_cons}, lower(19.992) upper(22.619))
>                     ({var}, lower(24.913) upper(47.613)), joint)
Interval tests     MCMC sample size =     10,000
       prob1 : 19.992 < {mpg:_cons} < 22.619,
               24.913 < {var} < 47.613
```

	Mean	Std. Dev.	MCSE
prob1	.9034	0.29543	.0076789

The joint posterior probability of both `{mpg:_cons}` and `{var}` belonging to their respective intervals is 0.9 with a posterior variance of 0.3 and MCSE of 0.008.

◁

▷ Example 4: Full syntax

We can specify multiple separate hypotheses and hypotheses tested jointly in one call to `bayestest interval`.

```
. bayestest interval (({mpg:_cons}, lower(19.992) upper(22.619))
>                      ({var}, lower(24.913) upper(47.613)), joint)
>                     ({mpg:_cons}, lower(21))
>                     ({var}, upper(40))

Interval tests     MCMC sample size =     10,000

       prob1 : 19.992 < {mpg:_cons} < 22.619,
               24.913 < {var} < 47.613
       prob2 : {mpg:_cons} > 21
       prob3 : {var} < 40
```

	Mean	Std. Dev.	MCSE
prob1	.9034	0.29543	.0076789
prob2	.6505	0.47684	.015786
prob3	.8136	0.38945	.0110613

In addition to the joint hypothesis from the previous example, we specified two new separate interval hypotheses for testing `{mpg:_cons}` $>$ 21 and for testing `{var}` $<$ 40. The estimated posterior probabilities for these hypotheses are 0.65 and 0.81, respectively.

◁

▷ Example 5: Point hypothesis for continuous parameters

As we discussed in *Introduction* above, point hypothesis for continuous parameters do not make sense, because the corresponding probability is 0:

```
. bayestest interval ({mpg:_cons}==21)

Interval tests     MCMC sample size =     10,000

       prob1 : {mpg:_cons}==21
```

	Mean	Std. Dev.	MCSE
prob1	0	0.00000	0

We can consider a small window around the value of interest and test an interval hypothesis instead:

```
. bayestest interval ({mpg:_cons}, lower(20.5) upper(21.5))

Interval tests     MCMC sample size =     10,000

       prob1 : 20.5 < {mpg:_cons} < 21.5
```

	Mean	Std. Dev.	MCSE
prob1	.4932	0.49998	.0138391

The probability that `{mpg:_cons}` is between 20.5 and 21.5 is about 50%.

Note that the probability of a continuous parameter belonging to a closed interval or semiclosed interval is the same as that for the open interval. Below we use suboption `inclusive` within `lower()` and `upper()` to request the closed interval.

```
. bayestest interval ({mpg:_cons}, lower(20.5,inclusive) upper(21.5,inclusive))
Interval tests     MCMC sample size =     10,000
       prob1 : 20.5 <= {mpg:_cons} <= 21.5
```

	Mean	Std. Dev.	MCSE
prob1	.4932	0.49998	.0138391

We obtain the same results as above for the corresponding open interval.

◁

▷ Example 6: Functions of parameters

We can test functions of model parameters. For example, let's compute the probability that the posterior standard deviation is greater than 6.

```
. bayestest interval (sd: sqrt({var}), lower(6))
Interval tests     MCMC sample size =     10,000
          sd : sqrt({var}) > 6
```

	Mean	Std. Dev.	MCSE
sd	.3793	0.48524	.0143883

The estimated probability is 0.38.

◁

Interval tests for discrete parameters

In this section, we demonstrate how to perform hypothesis testing for a discrete parameter.

First, we simulate data from the Poisson distribution with a mean of 2.

```
. clear
. set seed 12345
. set obs 20
number of observations (_N) was 0, now 20
. generate double y = rpoisson(2)
```

We fit a Bayesian Poisson model to the data and specify a discrete prior for the mean $P(\mu = k) = 0.25$ for $k = 1, 2, 3, 4$.

```
. set seed 14

. bayesmh y, likelihood(dpoisson({mu}))
> prior({mu}, index(0.25,0.25,0.25,0.25)) initial({mu} 2)
Burn-in ...
Simulation ...

Model summary
------------------------------------------------------------------------------
Likelihood:
  y ~ poisson({mu})

Prior:
  {mu} ~ index(0.25,0.25,0.25,0.25)
------------------------------------------------------------------------------

Bayesian Poisson model                           MCMC iterations  =     12,500
Random-walk Metropolis-Hastings sampling         Burn-in          =      2,500
                                                 MCMC sample size =     10,000
                                                 Number of obs    =         20
                                                 Acceptance rate  =      .2552
Log marginal likelihood =  -31.58903             Efficiency       =      .4428

------------------------------------------------------------------------------
             |                                                Equal-tailed
           y |      Mean   Std. Dev.     MCSE     Median  [95% Cred. Interval]
-------------+----------------------------------------------------------------
          mu |    2.0014   .1039188   .001562          2          2          2
------------------------------------------------------------------------------
```

▷ Example 7: Point hypotheses for discrete parameters

We can compute probabilities for each of the four discrete values of {mu}.

```
. bayestest interval ({mu}==1) ({mu}==2) ({mu}==3) ({mu}==4)

Interval tests     MCMC sample size =    10,000

       prob1 : {mu}==1
       prob2 : {mu}==2
       prob3 : {mu}==3
       prob4 : {mu}==4

-----------------------------------------------
             |      Mean    Std. Dev.      MCSE
-------------+---------------------------------
       prob1 |     .0047     0.06840   .0013918
       prob2 |     .9892     0.10337   .0027909
       prob3 |     .0061     0.07787   .0017691
       prob4 |         0     0.00000          0
-----------------------------------------------
```

The posterior probability that {mu} equals 2 is 0.99.

◁

Example 8: Interval hypotheses for discrete parameters

As we can with continuous parameters, we can test interval hypotheses for discrete parameters. For example, we can compute the probability of whether {mu} is between 2 and 4.

```
. bayestest interval {mu}, lower(2) upper(4)
Interval tests     MCMC sample size =     10,000
       prob1 : 2 < {mu} < 4
```

	Mean	Std. Dev.	MCSE
prob1	.0061	0.07787	.0017691

The estimated probability is very small.

Note that unlike hypotheses for continuous parameters, hypotheses including open intervals and closed or semiclosed intervals for discrete parameters may have different probabilities.

```
. bayestest interval {mu}, lower(2, inclusive) upper(4, inclusive)
Interval tests     MCMC sample size =     10,000
       prob1 : 2 <= {mu} <= 4
```

	Mean	Std. Dev.	MCSE
prob1	.9953	0.06840	.0013918

The estimated posterior probability that {mu} is between 2 and 4, inclusively, is drastically different compared with the results for the corresponding open interval.

◁

Stored results

bayestest interval stores the following in r():

Scalars
- r(skip) — number of MCMC observations to skip in the computation; every r(skip) observations are skipped
- r(corrlag) — maximum autocorrelation lag
- r(corrtol) — autocorrelation tolerance

Macros
- r(expr_#) — #th probability expression
- r(names) — names of probability expressions

Matrices
- r(summary) — test results for parameters in r(names)

Methods and formulas

Let θ be a model parameter and $\{\theta_t\}_{t=1}^T$ be an MCMC sample of size T drawn from the marginal posterior distribution of θ. It is often of interest to test how likely it is that θ belongs to a particular range of values. Note that testing a point null hypothesis such as H_0: $\theta = \theta_0$ is usually of no interest for parameters with continuous posterior distributions, because the posterior probability $P(H_0)$ is 0.

To perform an open-interval test of the form

$$H_0\colon\ \theta \in (a,b) \text{ versus } H_a\colon\ \theta \notin (a,b)$$

we estimate the posterior probability of H_0 from the given MCMC sample. The `bayestest interval` command calculates the probability $P(H_0)$ based on the simulated marginal posterior distribution of θ. The estimate is given by the frequency of inclusion of θ_ts in the test interval

$$\widehat{P}(H_0) = \frac{1}{T}\sum_{t=1}^{T} 1_{\{\theta_t \in (a,b)\}} \tag{1}$$

where $1_{\{A\}}$ is an indicator function and equals 1 if A is true and 0 otherwise.

When a model parameter θ is discrete, the following closed- and semiclosed-interval tests may be of interest in addition to open-interval tests:

$$H_0\colon\ \theta = a \text{ versus } H_a\colon\ \theta \neq a$$

$$H_0\colon\ \theta \in [a,b] \text{ versus } H_a\colon\ \theta \notin [a,b]$$

$$H_0\colon\ \theta \in [a,b) \text{ versus } H_a\colon\ \theta \notin [a,b)$$

$$H_0\colon\ \theta \in (a,b] \text{ versus } H_a\colon\ \theta \notin (a,b]$$

The corresponding probabilities are calculated as follows:

$$\widehat{P}(H_0) = \frac{1}{T}\sum_{t=1}^{T} 1_{\{\theta_t = a\}}$$

$$\widehat{P}(H_0) = \frac{1}{T}\sum_{t=1}^{T} 1_{\{\theta_t \in [a,b]\}}$$

$$\widehat{P}(H_0) = \frac{1}{T}\sum_{t=1}^{T} 1_{\{\theta_t \in [a,b)\}}$$

$$\widehat{P}(H_0) = \frac{1}{T}\sum_{t=1}^{T} 1_{\{\theta_t \in (a,b]\}}$$

The probability of an alternative hypothesis is always given by $P(H_a) = 1 - P(H_0)$.

The formulas above can be modified to accommodate joint hypotheses tests by multiplying the indicator functions of the individual hypothesis statements. For example, for a joint hypothesis H_0: $\theta_1 > a, \theta_2 < b$, we would replace the indicator function with $1_{\{\theta_{1t} > a\}} \times 1_{\{\theta_{2t} < b\}}$ in (1), where $\{\theta_{1t}\}_{t=1}^{T}$ and $\{\theta_{2t}\}_{t=1}^{T}$ are the corresponding MCMC samples for θ_1 and θ_2.

Also see

[BAYES] **bayesmh** — Bayesian regression using Metropolis–Hastings algorithm

[BAYES] **bayesmh postestimation** — Postestimation tools for bayesmh

[BAYES] **bayesstats summary** — Bayesian summary statistics

[BAYES] **bayestest model** — Hypothesis testing using model posterior probabilities