bmaregress — Bayesian model averaging for linear regression

Description

bmaregress performs Bayesian model averaging (BMA) for linear regression, which accounts for the uncertainty of which predictors should be included in the regression model. It can be used for inference, prediction, or model selection. Inference can be made about models based on posterior model probabilities (PMPs), importance of predictors based on posterior inclusion probabilities (PIPs), and regression coefficients based on their posterior distributions. bmaregress allows you to include predictors as groups and provides several ways of dealing with interaction terms. It supports a variety of priors for models and regression coefficients. Also see *Brief motivation* in *Remarks and examples* of [BMA] **Intro** for a quick overview of BMA.

Quick start

Perform BMA for linear regression of y on x1, x2, and x3 using the default model enumeration and the default priors

    bmaregress y x1 x2 x3

Same as above, but specify binomial model prior distribution with a 0.4 probability of inclusion for all predictors

    bmaregress y x1 x2 x3, mprior(binomial 0.4)

Same as above, but with prior probability of inclusion of 0.2 for x1 and 0.6 for x2

    bmaregress y x1 x2 x3, mprior(binomial x1 0.2 x2 0.6)

Specify beta-binomial model prior with shape parameters 2 and 4

    bmaregress y x1 x2 x3, mprior(betabinomial 2 4)

Specify a fixed value of 0.5 for the g parameter of a Zellner's g-prior for regression coefficients

    bmaregress y x1 x2 x3, gprior(fixed 0.5)

Same as above, but use the Markov chain Monte Carlo (MCMC) model composition (MC3) sampling algorithm with 5,000 burn-in iterations and MCMC sample size of 50,000

    bmaregress y x1 x2 x3, gprior(fixed 0.5) sampling burnin(5000) ///
      mcmcsize(50000)

Specify a random g parameter with a hyper-g prior distribution with hyperparameter 3

    bmaregress y x1 x2 x3, gprior(hyperg 3)

Specify that predictors x2 and x3 be considered as a group, and include predictor x4 in all models

    bmaregress y (x4, always) x1 (x2 x3)

Specify factor variables a and b and their interaction, and request that no heredity rules be applied

    bmaregress y x1 i.a##i.b, heredity(none)

Redisplay results in variable input order, and report only predictors with PIP above 0.5

        bmaregress, inputorder pipcutoff(0.5)

Save model results to bmadata.dta on replay, and store estimates in memory as bmaest

        bmaregress, saving(bmadata.dta)
        estimates store bmaest

Menu

Statistics > Bayesian model averaging > Linear regression

Syntax

BMA linear regression with in–out predictors

   bmaregress *depvar* [*inoutvars*] [*if*] [*in*] [*weight*]

      [, mprior(*mspec*) gprior(*gspec*) *options*]

BMA linear regression with always-included predictors

   bmaregress *depvar* (*alwaysvars*, always) [*inoutvars*] [*if*] [*in*] [*weight*]

      [, mprior(*mspec*) gprior(*gspec*) *options*]

BMA linear regression with groups of predictors

   bmaregress *depvar* [(*alwaysvars*, always)] [*inoutspec*] [*if*] [*in*] [*weight*]

      [, mprior(*mspec*) gprior(*gspec*) *options*]

inoutvars and *alwaysvars* are *varlist*.

inoutspec is a combination of in–out terms *inoutterm*, where *inoutterm* is one of in–out predictors, *var-name*, or one of in–out groups of predictors, (*varlist*). *varname* is always its own group; that is, (*var-name*) is implied. Also see *Groups of predictors* in *Remarks and examples*.

inoutvars, *alwaysvars*, and *inoutspec* may not contain duplicate terms. See [D] **vl** for managing large variable lists.

options	Description
Model	
<u>mprior</u>(*mspec*)	prior for model space; default is mprior(betabinomial)
gprior(*gspec*)	prior for Zellner's *g* parameter; default is gprior(bench)
<u>hered</u>ity(*heredspec*)	heredity rule for interactions of predictors; default is heredity(strong)
groupfv	group all factor variables and their interactions
Simulation	
<u>enum</u>eration	model enumeration; default with [fixed *g*] and no more than 12 predictors or groups; not allowed with [random *g*]
sampling	model sampling; default with [random *g*] or with more than 12 predictors or groups
mcmcsize(*#*)	MCMC sample size; default is mcmcsize(10000)
<u>burnin</u>(*#*)	burn-in period; default is burnin(2500)
<u>thin</u>ning(*#*)	thinning interval; default is thinning(1)
<u>minit</u>ial(*minitspec*)	initial model for MC3 sampling; default varies
<u>ginit</u>ial(*ginitspec*)	initial value for random *g* for MC3 sampling; default varies
rseed(*#*)	random-number seed
Reporting	
<u>saving</u>(*filename*[, replace])	save simulation results to *filename*.dta
<u>clev</u>el(*#*)	set credible interval (CrI) level for random *g*; default is clevel(95)
hpd	display highest posterior density (HPD) CrIs instead of the default equal-tailed CrIs for random *g*
[no]dots	suppress or display dots every 100 iterations and iteration numbers every 1,000 iterations; default is dots
dots(*#*[, every(*#*)])	display dots as simulation is performed
<u>inputo</u>rder	display results in input order; default is descending order of PIP
<u>recen</u>tered	show coefficients for recentered predictors
pipcutoff(*#*)	lowest PIP value for displaying results; default is pipcutoff(.01)
<u>notab</u>le	suppress estimation table
<u>nohead</u>er	suppress output header
title(*string*)	display *string* as title above the table of parameter estimates
display_options	control spacing, line width, and base and empty cells
allcoef	show all coefficients; equivalent to pipcutoff(0)

inoutvars, *alwaysvars*, and *inoutspec* may contain factor variables; see [U] 11.4.3 Factor variables.

depvar, *inoutvars*, *alwaysvars*, and *inoutspec* may contain time-series operators; see [U] 11.4.4 Time-series varlists.

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

Only fweights are allowed; see [U] 11.1.6 weight.

allcoef does not appear in the dialog box.

See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

mspec	Description
<u>betab</u>inomial	same as betabinomial 1 1; the default
<u>betab</u>inomial $\#_1$ $\#_2$	beta-binomial prior using shape parameters of beta distribution
<u>betab</u>inomial $\#$	beta-binomial prior using expected model size $\#$
<u>unif</u>orm	uniform prior
<u>binom</u>ial $\#$	binomial prior with the same inclusion probability $\#$ for all in–out terms
<u>binom</u>ial	same as binomial 0.5
<u>binom</u>ial *inoutterm* $\#$ $[\ldots]$	binomial prior with specified inclusion probabilities

gspec	Description
Fixed priors:	
bench	set $g = \max(n, p^2)$; the default
uip	set $g = n$ (n is sample size); unit-information prior
ric	set $g = p^2$ (p is the number of predictors); risk inflation criterion
sqrtn	set $g = \sqrt{n}$
fixed $\#$	set g to fixed value $\#$
ebl	empirical Bayes local estimate
Random priors:	
betashrink $\#_1$ $\#_2$	beta prior with shapes $\#_1$ and $\#_2$ on shrinkage $\delta = g/(1 + g)$
betabench $\#$	benchmark beta-shrinkage prior with shapes $\# \times \max(n, p^2)$ and $\#$
hyperg $\#$	hyper-g prior with hyperparameter $\#$
hypergn $\#$	hyper-g/n prior with hyperparameter $\#$
zsiow	Zellner–Siow prior
robust	robust prior

heredspec	Description
strong	strong heredity; the default
weak	weak heredity
none	no heredity

Options

    ┌─ Model ┐

mprior(*mspec*) specifies model prior, a prior distribution on the model space. In a regression setting, this is equivalent to specifying prior probabilities for the inclusion of predictors in a model. A group of predictors is viewed as one term in the model prior specification. That is, the prior inclusion probability is specified for the entire group and not the individual predictors.

    mspec is one of betabinomial, betabinomial $\#_1$ $\#_2$, betabinomial $\#$, uniform, binomial $\#$, binomial, and binomial *inoutterm* $\#$ $[\ldots]$.

    betabinomial, the default, and betabinomial $\#_1$ $\#_2$ specify a binomial model prior with an inclusion probability (IP) and a beta prior on the IP. The two arguments specify the shape parameters of the beta distribution. betabinomial is a synonym for betabinomial 1 1.

betabinomial # specifies a binomial model prior and a beta prior for the IP. This specification includes one argument #, which is the expected number of *inoutterm*s to be included (Ley and Steel 2009). # must be greater than 0 and less than the total number of *inoutterm*s specified in the regression model. If there are no groups specified, # is the expected model size.

uniform specifies a uniform model prior, which assumes an equal probability for each model.

binomial # specifies a binomial model prior, in which all *inoutterm*s have the same IP # between 0 and 1.

binomial is a synonym for binomial 0.5.

binomial *inoutterm* # [*inoutterm* # [. . .]] specifies a binomial model prior with different values of IP for *inoutterm*s. IPs are specified as pairs consisting of an *inoutterm* and the corresponding IP #, $0 < \# < 1$. *inoutterm*s for which no IP is specified are assumed to have probability of 0.5. The specified *inoutterm*s must be one of the *inoutterm*s from the regression model specification. For grouped terms, it is sufficient to list one of the predictors from the group to specify the IP, and this probability will be used for the entire group. But you may not specify different IPs for multiple predictors from the same group. You can specify the same IP # for multiple predictors from different groups by using binomial *inoutterm* [*inoutterm* [. . .]] #.

gprior(*gspec*) specifies a fixed value or a prior distribution (hyperprior) for parameter g of a Zellner's g-prior assumed for regression coefficients. Let p be the number of predictors and n be the sample size.

gspec is one of the fixed priors—bench, uip, ric, sqrtn, fixed #, or ebl—or one of the random priors—betashrink $\#_1$ $\#_2$, betabench #, hyperg #, hypergn #, zsiow, or robust.

The following are the fixed priors:

bench specifies the benchmark prior with $g = \max(n, p^2)$, which is a combination of the unit-information and risk inflation criterion priors. This is the default. This prior was suggested by Fernández, Ley, and Steel (2001a), who found it to perform well in a variety of cases with respect to a model's predictive performance.

uip specifies the unit-information prior with $g = n$ (Kass and Wasserman 1995), which assigns the same amount of information to the prior for regression coefficients as is contained in one observation. Under this prior, the Bayes factors behave like the Bayesian information criterion (BIC).

ric specifies the prior based on the risk inflation criterion with $g = p^2$ (Foster and George 1994). This prior has good small-sample properties.

sqrtn specifies the square-root n prior with $g = \sqrt{n}$ (Porwal and Raftery 2022b).

fixed # specifies a fixed value # for g.

ebl use the empirical Bayes local estimate for g as suggested by Liang et al. (2008). Instead of using an a priori value for g, this prior estimates g from the data and does it separately for each model.

The following are the random priors:

betashrink $\#_1$ $\#_2$ specifies a general beta-shrinkage prior, which uses a beta prior with shape parameters $\#_1$ and $\#_2$ for the shrinkage factor $\delta = g/(g + 1)$.

betabench # specifies the benchmark beta-shrinkage prior suggested by Ley and Steel (2012), which uses a beta prior with shape parameters $\# \times \max(n, p^2)$ and # on the shrinkage factor $\delta = g/(g+1)$.

hyperg # specifies the hyper-g prior suggested by Liang et al. (2008), which is a beta prior with shape parameters 1 and $\#/2 - 1$ on shrinkage factor $\delta = g/(g+1)$. The hyperparameter # must satisfy $2 < \# \leq 4$.

hypergn # specifies the hyper-g/n prior suggested by Liang et al. (2008), which is a beta prior with shape parameters 1 and $\#/2 - 1$ on $g/(g+n)$. The hyperparameter # must satisfy $2 < \# \leq 4$.

zsiow specifies a prior suggested by Zellner and Siow (1980), which is an inverse-gamma prior on g with shape 0.5 and rate $n/2$.

robust specifies the robust prior as defined in Bayarri et al. (2012).

heredity(*heredspec*) specifies a heredity rule for interaction terms. This option is relevant only when interaction terms, terms that include # or ##, are present; see [U] **11.4.3 Factor variables** and *Handling factor variables and interactions* in *Remarks and examples*.

heredspec is one of strong, weak, or none.

strong specifies that whenever an interaction is included in a model, all main effects are also included, which aids interpretation. This is the default. This rule is the most restrictive for the model space. If your goal is prediction, you may decide to choose a less restrictive rule.

weak specifies that whenever an interaction is included in a model, at least one main effect is also included. This rule is less restrictive for the model space than heredity(strong).

none specifies that interactions be included independently of the main effects. This rule does not make any additional restrictions for the model space and may be more beneficial when prediction is of interest.

groupfv specifies that all factor-variable terms be grouped and is relevant only in the presence of factor variables; see [U] **11.4.3 Factor variables** and *Handling factor variables and interactions* in *Remarks and examples*.

    Simulation

enumeration specifies that model enumeration be used to explore the model space. This option considers all 2^p models, where p is the number of predictors, and is thus not feasible with many predictors. It is the default for a fixed g when the number of predictors or groups is less than or equal to 12. And it may not be specified with more than 24 predictors or groups of predictors. Only one of enumeration or sampling may be specified.

enumeration is not allowed with a random g, meaning when one of the random g priors is specified in option gprior(): betashrink, betabench, hyperg, hypergn, zsiow, or robust.

Model enumeration is a deterministic procedure, so the following sampling options are not available with it: mcmcsize(), burnin(), thinning(), minitial(), ginitial(), rseed(), clevel(), and hpd. Also, options nodots, dots, and dots() are not relevant and thus ignored.

sampling specifies that sampling be used to explore the model space. This is the default with a random g or when the number of predictors or groups is greater than 12. With a fixed g, the MC3 algorithm (Madigan and York 1995) is used to sample from the model space (Fernández, Ley, and Steel 2001a and Ley and Steel 2012). With a random g, a modified MC3 is used—MC3 is used to sample models, and an adaptive Metropolis–Hastings method is used to sample g. Only one of sampling or enumeration may be specified.

mcmcsize(#) specifies the target MCMC sample size for the model space. The default is mcmcsize(10000). The total number of iterations for the MC3 algorithm equals the sum of the burn-in iterations and the MCMC sample size (in the absence of thinning). If thinning is present, the total number of MCMC iterations is computed as burnin() + (mcmcsize() − 1) × thinning() + 1. Computation time is proportional to the total number of iterations. The MCMC sample size determines the precision of posterior summaries. Also see *Convergence of BMA* in *Remarks and examples* and *Burn-in period and MCMC sample size* in [BAYES] **bayesmh**. This option is not allowed with model enumeration.

burnin(#) specifies the number of iterations for the burn-in period of MCMC. The MCMC states simulated during burn-in are not used for estimation. The default is burnin(2500). Also see *Convergence of BMA* in *Remarks and examples* and *Burn-in period and MCMC sample size* in [BAYES] **bayesmh**. This option is not allowed with model enumeration.

thinning(#) specifies the thinning interval. Only simulated values from every $(1 + k \times \#)$th iteration for $k = 0, 1, 2, \ldots$ are saved in the final MCMC sample; all other simulated values are discarded. The default is thinning(1); that is, all simulation values are saved. Thinning greater than one is typically used for decreasing the autocorrelation of the simulated MCMC sample. This option is not allowed with model enumeration.

minitial(*minitspec*) specifies an initial model for the MCMC sample of models. This option is not allowed with model enumeration.

    minitspec is one of random, *varlist*, null, or full.

    random specifies that the initial model for the MCMC be generated randomly from the model prior distribution. This is the default with fewer than 1,000 predictors.

    varlist specifies that the initial model include a given set of predictors *varlist*. The initial model must be compatible with the specified regression model for the in–out predictors. For example, if you specified the following command,

```
. bmaregress x1 (x2 x3) (x4, always)
```

    then the valid initial model may 1) include or exclude both x2 and x3 and 2) include or exclude x1.

    null specifies that the null model, the model with only a constant term, be used as initial. This is the default with 1,000 predictors or more.

    full specifies that the full model, the model that includes all predictors, be used as initial.

ginitial(*ginitspec*) specifies the initial g value for the MCMC for random g. This option is allowed only with random g priors specified in option gprior(): betashrink, betabench, hyperg, hypergn, zsiow, or robust. This option is not allowed with model enumeration.

    ginitspec is one of ebl, random, or #.

    ebl specifies that the empirical Bayes local estimate for the initial model be used as the initial value for g. This is the default.

    random specifies that the initial value for g be generated randomly from its prior distribution.

    # specifies the initial value for g.

rseed(#) sets the random-number seed. This option can be used to reproduce results. rseed(#) is equivalent to typing set seed # prior to calling bmaregress; see [R] **set seed**. This option is not allowed with model enumeration.

saving(*filename*[, replace]) saves simulation results in *filename*.dta. The replace option speci-
fies to overwrite *filename*.dta if it exists. If the saving() option is not specified, bmaregress saves
simulation results in a temporary file for later access by postestimation commands. This temporary
file will be overridden every time bmaregress is run and will also be erased if the current estimation
results are cleared. saving() may be specified during estimation or on replay.

The saved dataset has the following structure. Variable _chain is the chain identifier; it is al-
ways 1. Variable _index records iteration numbers. Variables _loglikelihood, _logposterior,
and _logmprior record the log-likelihood, log-posterior, and log model-prior, respectively.
_logposterior records the joint posterior of models and g. Variable g records the used fixed value
of Zellner's g parameter or a sample from its posterior distribution. Variable logs2delta contains
a sample from the posterior distribution of parameter $\ln(s_{j,\delta}^2)$, defined in *Conditional posterior dis-
tribution of model parameters* of *Methods and formulas*. This parameter corresponds to the log of
two times the inverse-gamma scale parameter of the posterior distribution of error variance for a
given model. Model states are encoded as binary vectors and saved in variables state_eq1_p1
through state_eq1_pp for each of the p predictors. Each of these variables can be viewed as a
sample from a posterior distribution of a random indicator for inclusion of the respective predictor
in a model. The model-specific posterior mean estimates of coefficients are saved in mean_eq1_p1
through mean_eq1_pp. The model-specific posterior variance estimates of coefficients are saved in
var_eq1_p1 through var_eq1_pp. Variable _hash records information used by bmaregress to
identify models. bmaregress saves only model states and parameter values that are different from
one iteration to another and the frequency of each state in variable _frequency. Thus, _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.

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

hpd displays the HPD CrIs instead of the default equal-tailed CrIs for random g.

Options clevel() and hpd are not allowed with model enumeration. These options are allowed only
with random g priors specified in option gprior(): betashrink, betabench, hyperg, hypergn,
zsiow, or robust.

nodots, dots, dots(#), and dots(#, every(#)) specify to suppress or display dots during simula-
tion. nodots, the default, suppresses the display of dots. dots displays dots every 100 iterations and
iteration numbers every 1,000 iterations; it is a synonym for dots(100, every(1000)). dots(#)
displays a dot every #iterations. If dots(..., every(#)) is specified, then an iteration number is dis-
played every #th iteration instead of a dot. dots(, every(#)) is equivalent to dots(1, every(#)).
During the adaptation period, a symbol a is displayed instead of a dot. The dots options are ignored
with model enumeration.

inputorder specifies that the results be displayed in the order in which predictors are specified in the
model. By default, the results are displayed in the descending order of PIP.

recentered specifies that coefficients for the recentered predictors be shown in the output table. By
default, the coefficients for the untransformed predictors are shown. The two representations are
related by a linear transformation that affects only the constant term in the model.

pipcutoff(#) specifies the lowest PIP limit for displaying the results. The results for predictors with
PIP lower than this limit are not shown in the output table. The default is pipcutoff(0.01). For
models with many predictors, you can limit the displayed results by using a higher pipcutoff().

`notable` suppresses the estimation table from the output.

`noheader` suppresses the output header either at estimation or upon replay.

`title(`*string*`)` specifies an optional title for the command that is displayed above the table of the parameter estimates.

display_options: `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(`*style*`)`, and `nolstretch`; see [R] **Estimation options**.

The following option is available with `bmaregress` but is not shown in the dialog box:

`allcoef` shows all regression coefficients. This is equivalent to `pipcutoff(0)`. This option may produce a lengthy output with many predictors.

Remarks and examples

Remarks are presented under the following headings:

Examples are presented under the following headings:

Introduction to BMA for linear regression

A widely used technique in data analysis is model selection, which comprises two stages: choosing an optimal model for the data based on specific criteria and conducting inference based on the selected model. The inference is conditional on the assumption that the data are actually generated from this model. This approach does not account for the uncertainty of model selection and thus may render the

results more precise than they actually are. This may lead to overly confident inferential conclusions and predictions. And if the selected model is substantively different from the true generating model, the results may even be biased.

Instead of selecting just one model, BMA (Leamer 1978) considers a set of plausible models and bases its inference on averaging across those models, weighing by model probabilities given the observed data. BMA is a principled statistical method that addresses the problem of underestimating or rather ignoring model uncertainty. You may not model averaging for every data analysis, but even if you need to choose just one model, you may find BMA helpful for selecting the best one available for your data.

See [BMA] **Intro** for a general introduction to BMA. In this entry, we focus on BMA for linear regression. In what follows, we present some of the main concepts of BMA following the discussions in Steel (2020), Moral-Benito (2015), and Hoeting et al. (1999).

In a regression setting with p potential predictors, model uncertainty is confined to which predictors should be included in the model (for example, Steel [2020] and Moral-Benito [2015]). There are 2^p possible regression models that include or exclude each of the p predictors in a model. Variable-selection methods such as stepwise regression and information critera focus on selecting one "best" model for the data according to some criteria. Depending on the application, results based on a single model may be overly optimistic (for example, Steel [2020]).

Unlike variable-selection methods, BMA incorporates (potentially) all models from the considered model space by averaging over it to estimate model parameters and obtain predictions. That is, in the context of a linear regression, BMA would potentially account for all 2^p models in the analysis. When we say "account for all 2^p models", we mean that BMA will consider these models and adjust parameter estimates for the likeliness of each model. Thus, BMA results incorporate the uncertainty about the model-selection process.

Suppose we have an outcome Y and p predictors $X_1, X_2, \ldots, X_p$. We want to model Y as a linear function of these predictors, but we do not know which ones to include in the model. We can use a BMA linear regression to explore all 2^p models.

Model setup. Consider an $n \times 1$ vector of outcome values $\mathbf{y} = (y_1, y_2, \ldots, y_n)'$, an $n \times p$ design matrix $\mathbf{X}$ containing values of p predictors, and an $n \times 1$ vector of ones $\mathbf{1}_n$. Let M_j be a regression model that contains a subset of p_j predictors stored in an $n \times p_j$ design submatrix $\mathbf{X}_j$. Then, for each $j \in \{1, 2, \ldots, 2^p\}$, consider a linear regression M_j,

$$\mathbf{y} = \alpha \mathbf{1}_n + \mathbf{X}_j \boldsymbol{\beta}_j + \boldsymbol{\epsilon}_j$$

where α is an unknown intercept, $\boldsymbol{\beta}_j$ is a $p_j \times 1$ vector of unknown (M_j-specific) regression coefficients, and $\boldsymbol{\epsilon}_j = (\epsilon_{1,j}, \epsilon_{2,j}, \ldots, \epsilon_{n,j})'$ is an $n \times 1$ vector of model-specific error terms, which are independently normally distributed with mean zero and variance σ^2.

Priors. In a Bayesian linear regression (see [BAYES] **bayes: regress**), we assume a prior $p(\alpha, \boldsymbol{\beta}_j, \sigma^2 | M_j)$ for model parameters α, $\boldsymbol{\beta}_j$, and σ^2, which is conditional on model M_j. In a BMA framework, model M_j itself is considered random and thus assumed to have a discrete prior $P(M_j)$ for $j \in \{1, 2, \ldots, 2^p\}$.

A BMA linear regression assumes noninformative priors for α and σ^2 and a Zellner's g-prior for $\boldsymbol{\beta}_j$. Thus, the prior for regression coefficients is controlled by the g parameter of a Zellner's g-prior; g can be fixed or random. For a random g, a prior distribution (hyperprior) is specified for g. The g parameter controls the amount of shrinkage of regression coefficients toward zero. Smaller g leads to more shrinkage. There are many choices for a g-prior; see *Priors for parameter g* in *Methods and formulas*.

Fernández, Ley, and Steel (2001a) suggest that the so-called benchmark prior, where $g = \max(n, p^2)$, tends to perform well in a variety of cases with respect to a model's predictive performance. Other priors suggested in the literature are $g = \sqrt{n}$, local empirical Bayes, and hyper-g priors (Porwal and Raftery 2022a). The use of hyperpriors is often suggested to deal with data robustness, when small differences in the data can lead to substantive changes in the results. See Steel (2020) and references therein for more discussion.

There are also a number of choices for a model prior $P(M_j)$; see *Priors on the model space* in *Methods and formulas*. A uniform prior, which assumes that each model is equally likely, and a beta-binomial prior with shape parameters of 1, which assumes that models of all sizes are equally likely, are some of the common choices. See Moral-Benito (2015), Steel (2020), and Fragoso, Bertoli, and Louzada (2018) for a discussion of various model priors and Ley and Steel (2009), Ley and Steel (2012), and Porwal and Raftery (2022b) for a comparison of performance of different priors.

As with any Bayesian results, BMA results (in particular, PMPs) can be sensitive to the choice of priors. See Steel (2020) for a discussion of the role of priors on both PMPs and posterior inference. Regardless of the chosen prior, you should perform sensitivity analysis using a variety of relevant priors to evaluate their impact on final results.

Estimation. The estimation of PMPs, model probabilities conditional on the observed data, is central to BMA estimation. PMPs are used as weights in the averaging of the model-specific results that produces the final BMA results. How PMPs are estimated depends on how the model space is explored and whether the g parameter is fixed or random.

For a moderate number of predictors, say, $p \leq 24$, model enumeration can be used to explore the entire space of 2^p models. With more predictors, model enumeration becomes less feasible, and an MCMC sampling algorithm, for example, MC3 (Madigan and York 1995), is used to explore the model space.

When g is fixed, analytical formulas are available to compute PMPs. We refer to these PMPs as analytical PMPs, $P_a(M_j|\mathbf{y})$. In the case of sampling, analytical PMPs are conditional on the model space visited by the sampling algorithm. When g is random, what we call a modified MC3 algorithm—MC3 for model space and adaptive Metropolis–Hastings (MH) for g—is used to jointly sample the model space and g. In this case, analytical formulas for PMPs are not available, and PMPs are estimated from the MCMC sample of models by using frequencies. We refer to these PMPs as frequency PMPs, $P_f(M_j|\mathbf{y})$.

Convergence. See *Convergence of BMA*.

Inference. In addition to standard Bayesian posterior inference for model parameters, BMA provides inference for models via PMP and formal assessment of the importance of considered predictors in relation to the outcome via PIP, which is a probability that a predictor is included in a model given the observed data. Models with high PMPs are of interest. And predictors with high PIPs are of interest. See [BMA] **bmastats models** and [BMA] **bmastats pip**.

Posterior inference for model parameters is performed with respect to the posterior distributions of model parameters over all models. For instance, with model enumeration, posterior mean and variance for regression coefficients are computed based on the posterior distribution of $\boldsymbol{\beta}$ and defined as follows,

$$\boldsymbol{\beta}_{\text{BMA}} = \boldsymbol{\mu}_{\text{BMA}} = E(\boldsymbol{\beta}|\mathbf{y}) = \sum_{j=1}^{2^p} P_a(M_j|\mathbf{y})\boldsymbol{\mu}_j$$

$$\text{Var}(\boldsymbol{\beta}|\mathbf{y}) = \sum_{j=1}^{2^p} P_a(M_j|\mathbf{y}) \left\{ \text{Var}(\boldsymbol{\beta}_j) + \boldsymbol{\mu}_j\boldsymbol{\mu}_j' \right\} - \boldsymbol{\mu}_{\text{BMA}}\boldsymbol{\mu}_{\text{BMA}}'$$

where $\boldsymbol{\mu}_j$ is the posterior mean of regression coefficients $\boldsymbol{\beta}_j$ based on model M_j.

Also see *Interpretation of BMA regression coefficients*.

Prediction. The early use of BMA was in the context of prediction. The posterior predictive density of **y** over all models is defined as a weighted PMP average of posterior predictive densities of **y** conditional on each model (for example, Moral-Benito [2015]). See [BMA] **bmapredict** for details.

Regression modeling and model space. As with a standard linear regression, you should also carefully consider the parameterization of the regression function in a BMA linear regression. This essentially relates to the definition of the BMA model space. Most theoretical results for BMA are developed under the assumption that the "true" model is included in the model space, although Steel (2020) comments that BMA can be shown to be robust to this assumption.

For instance, any interaction and nonlinear terms that might improve the explanatory power of the regression model need to be considered. If they are omitted from the model space at the onset, there is no chance for BMA to explore the models involving these terms. But care should be taken with how to include such terms properly in a BMA regression depending on the research objective; see *Groups of predictors* and *Handling factor variables and interactions*.

Model diagnostics. In addition to MCMC convergence diagnostics as described in *Convergence of BMA*, any model diagnostics such as residual analysis used for a standard linear regression should be considered for a BMA linear regression as well. But because there are multiple models, the application of model checks is not as straightforward for BMA. Ideally, diagnostics should be explored for each model. This may not be feasible with many models. One recommendation in the literature is to perform model checks for the full model before the estimation and check diagnostics for at least the models with high PMP after estimation (Hoeting et al. 1999). Also see Banner and Higgs (2017) for other recommendations.

For various usages and applications of BMA in practice, see *Usage of BMA* in *Remarks and examples* of [BMA] **Intro**. For technical details about BMA computations, see *Methods and formulas*.

Convergence of BMA

As with standard Bayesian estimation, whenever sampling is used, the convergence of BMA needs to be checked before proceeding with the analysis. In addition to standard MCMC convergence diagnostics for model parameters (see *Convergence of MCMC* in *Remarks and examples* of [BAYES] **bayesmh**), the convergence of the model MCMC sample must be established.

For models with many predictors, convergence may be difficult (or even infeasible) to achieve unless the model space is restricted using a strong model prior. An example of a strong model prior is an independent Bernoulli prior, where a small group of predictors is assigned a high probability of inclusion and the rest are assigned a low probability of inclusion. The default model prior used by bmaregress specifies an equal probability for a model of any size.

Sampling correlation (Fernández, Ley, and Steel 2001a) is used to evaluate the MCMC convergence of the BMA model. It is defined as the correlation between the analytical and frequency PMPs. It measures the agreement between the expected and observed PMPs. When sampling correlation has a low positive value (less than 10%) or even negative, nonconvergence is suspected. You should review your model specification carefully and potentially increase the MCMC sample size.

Sampling correlation can also be missing. This is always the case when there is only one visited model. It is possible that there is only one plausible model in the BMA model space, which BMA found. In this case, there is no reason to suspect nonconvergence. However, in general, unless the regression model

has no predictors, one visited model is insufficient to explore the full model space, and you should not trust the reported results. You may also get a missing sampling correlation when there is an insufficient number of MCMC iterations.

Sampling correlation is not a formal measure of MCMC convergence in BMA. Its precision may vary from one sample to another. In fact, with a random g, there is no reliable way to estimate sampling correlation, because there is no analytical formula for PMP. In this case, sampling correlation is computed based on the harmonic-mean approximation of the analytical PMP distribution and is thus subject to the accuracy of the approximation, in addition to the inherent sampling variability. You should interpret a sampling-correlation value as a guide for BMA convergence rather than a formal assessment of it.

There is no formal way to determine BMA convergence. Sampling correlation is just one way to explore MCMC convergence numerically. You should also look at graphical convergence diagnostics (see [BMA] **bmagraph pmp**) and use other standard MCMC diagnostic tools (see *Convergence of MCMC* in *Remarks and examples* of [BAYES] **bayesmh**).

Interpretation of BMA regression coefficients

To avoid the infamous problem of mixing "apples and oranges" during averaging in BMA, it is important that a quantity being averaged has a common interpretation across all models (for example, Steel [2020] and Fragoso, Bertoli, and Louzada [2018]). The motivation for BMA originally was in the context of predicting new outcome values. Because such predictions have the same interpretation across all models, the application of averaging is justified.

A BMA linear model also provides model-averaged estimates for the regression coefficients, intercept, and error variance. The error variance has a common meaning across the models, so the averaging is applicable. During estimation, BMA recenters predictors to make them orthogonal to the intercept, so the intercept also has the same interpretation across all models. The regression coefficients, however, should be evaluated more carefully.

In one of the comments to Hoeting et al. (1999), Draper (1999) points out that the meaning of the coefficient of X_1 in a regression of Y on X_1 is different from that in a regression of Y on X_1 and X_2. Hoeting et al. (1999) respond that one can recast BMA into a one-model space by viewing it as a standard Bayesian linear regression including all predictors, in which the prior allows some of the regression coefficients to be zero or close to zero. Banner and Higgs (2017) further investigate the use of BMA in practice for the interpretation of partial regression coefficients and provide some guidance and graphical tools to aid interpretation.

In summary, when the predictors are orthogonal or in the case of a simple linear regression, the (partial) regression coefficients have the same interpretation across models and their averaged BMA estimates are appropriate. This might be rare in practice, especially with observational data. In more realistic situations such as in the presence of collinear predictors or interaction terms, care should be taken with the interpretation of BMA coefficients as partial regression coefficients. In these cases, perhaps the BMA inference should shift toward estimation of marginal effects instead.

In a Bayesian context, the interpretation of regression coefficients, particularly in the presence of collinearity, should also be considered in the context of their prior distribution. As indicated by Leamer (1973), "a well-defined prior distribution can have no problem interpreting the sample evidence." From this perspective, the interpretation of regression coefficients should account for their prior information, especially in cases when the observed data do not provide sufficient evidence. Of course, the exploration of the sensitivity of the results to the assumed prior becomes even more important in such cases.

Using the bmaregress command

The `bmaregress` command performs BMA for linear regression. It explores the model space either exhaustively, when this is feasible, or by using MCMC. It reports various summaries of the visited models and included predictors, and posterior distributions of model parameters.

The simplest syntax of `bmaregress` is the same as for `regress` (see [R] **regress**)—you list the dependent and predictor variables following the command:

```
. bmaregress y x1 x2 x3
```

With this specification, each of the three variables `x1`, `x2`, and `x3` will be in and out of the model, that is, will be either included or excluded from the model.

If you wish to always include some of the predictors in the model, you can use the following specification:

```
. bmaregress y (x1 x3, always) x2
```

You can also group predictors so that each group is in and out of the model. This is useful with factor variables and variables that are functions of other variables that you may want to include together in the regression models:

```
. bmaregress y x1 (x2 x3)
```

In the above, `x2` and `x3` will always be included or excluded together. So, technically, `bmaregress` determines the model space by the number of groups p_g. In this example, the model space will include $2^2 = 4$ models and not $2^3 = 8$. Also see *Groups of predictors*.

In the presence of interactions, `bmaregress` always includes all main effects whenever the corresponding interaction is in the model, but you can change this with the `heredity()` option. See *Handling factor variables and interactions*.

`bmaregress` uses the default beta-binomial prior with shape parameters of one on the model space, which assumes the uniform prior for the model size, and the default benchmark (fixed) prior for the g parameter. You can change them by specifying the `mprior()` and `gprior()` options, respectively. Default priors are provided for convenience. They are chosen based on the recommendations in the literature, which found them to be fairly uninformative in a variety of cases. But they may not apply to all cases. It is important that you carefully evaluate the choice of priors and specify the priors that are appropriate for your model and research questions. You cannot simply rely on the provided defaults. It is also important that you consider various priors to check the sensitivity of the results to the prior assumptions. See *Priors* in *Introduction to BMA for linear regression*.

By default, `bmaregress` uses model enumeration to explore the model space fully with up to 12 predictors or, more precisely, with up to 12 groups of predictors, ignoring the always group. For more than 12 predictors, it uses the MC3 sampling algorithm. If desired, you can use sampling with 12 predictors or fewer by specifying the `sampling` option. You can also use model enumeration with up to 24 predictors by specifying the `enumeration` option. With a random g, the model space cannot be explored fully, and a modified version of MC3 is used: MC3 is used to sample models, and an adaptive MH, similar to that described in [BAYES] **bayesmh**, is used to sample g.

When a sampling method is used, `bmaregress` uses 2,500 burn-in iterations and an MCMC sample size of 10,000. You can change this, respectively, with options `burnin()` and `mcmcsize()`. Also, with sampling, the results produced by `bmaregress` are stochastic: they can change from one run of the command to another. You can specify the `rseed()` option for reproducibility.

bmaregress also supports various reporting options. For instance, with many predictors or with a random g, you may want to specify the dots and related options to see the simulation progress. By default, the predictors are listed in the order from highest to lowest PIP. You can use the inputorder option to display them in the order they are specified with bmaregress.

Finally, you can use the saving() option to save your BMA MCMC results to a file. This can be done during or after estimation. Some postestimation commands such as bmacoefsample, estimates store, and estimates save require that you save your BMA results before running the commands.

As with any MCMC algorithms, the MC3 algorithms require that you check sampling convergence; see *Convergence of BMA*. You should always use bmagraph pmp to explore sampling convergence graphically after bmaregress.

You can use bmastats models and bmagraph varmap to explore model and variable-inclusion summaries and bmastats pip to explore variable-inclusion probabilities. bmastats msize and bmagraph msize provide information about the model size. bmastats jointness reports whether pairs of predictors tend to be included in the model together or separately. bmastats lps can be used to check the predictive performance of the fitted BMA model and compare it with other potential BMA models. bmacoefsample samples regression coefficients and other model parameters from their posterior distributions, and bmagraph coefdensity plots these distributions. Finally, if you are interested in a BMA prediction, use bmapredict.

There are also two Stata commands that help you prepare your data and specify predictors before your BMA analysis: splitsample and vl. splitsample divides your sample into several random subsamples, which can be used for training, validation, and prediction (testing). vl is useful for managing a potentially large set of predictors for use with bmaregress. It helps you specify predictors conveniently without typing every variable name. See [D] **splitsample** and [D] **vl** for details.

Groups of predictors

The bmaregress command allows you to group predictors using the parentheses notation. In the context of BMA, a group of predictors is treated as one in-and-out term—all predictors in the group are either included or excluded from linear regression models. The model space of a BMA linear regression is thus defined by the grouping of predictors. The number of groups p_g determines the size of the model space, 2^{p_g}. The model prior distribution, specified with option mprior(), is also imposed on the space of groups.

For example, suppose we have three predictors x1, x2, and x3, and x2 and x3 are grouped together:

        . bmaregress y x1 (x2 x3)

The model space comprises four possible regression models: no predictors, only x1, only x2 and x3, and all variables x1, x2, and x3. The default model prior distribution is beta-binomial with shape parameters of 1.

We can use the mprior() option to specify, say, a binomial prior with the following prior probabilities of inclusion of predictors and groups of predictors:

        . bmaregress y x1 (x2 x3), mprior(binomial x1 0.4 (x2 x3) 0.6)

In the above command, we set the probability of inclusion to 0.4 for x1 and to 0.6 for the group that includes x2 and x3.

In the output of bmaregress, the Group column contains the information about the groups of predictors.

Grouping restricts the model space and may diminish the predictive performance of a BMA linear regression, so it should be used thoughtfully.

Handling factor variables and interactions

The levels of a categorical predictor, excluding the base level, are commonly treated as independent in-and-out terms in variable-selection methods. Suppose a categorical predictor `a` has four levels, with one being the base level, and is included as a factor variable in the model:

```
. bmaregress y i.a
```

The above introduces three regression terms: `2.a`, `3.a`, and `4.a`, which will be in and out of the model independently. The model space will then include the regression models `regress y 2.a`, `regress y 3.a`, `regress y 4.a`.

If you wish to treat these three terms as one group, you can type

```
. bmaregress y (i.a)
```

or you can use the `groupfv` option,

```
. bmaregress y i.a, groupfv
```

In this case, the model space will include only the regression model `regress y 2.a 3.a 4.a` and the constant-only model.

The `groupfv` option will group all specified factor variables. For example, the specification

```
. bmaregress y i.a i.b i.a#i.b, groupfv
```

will introduce three groups: `(i.a)`, `(i.b)`, and `(i.a#i.b)`.

However, the following specification will group all interactions and main effects in one group `(i.a##i.b)`:

```
. bmaregress y i.a##i.b, groupfv
```

If you include an interaction in your BMA model, you need to consider the relationship between this interaction and the main effects. Do you want the interaction to be included in regression models independently of the main effects or not? If your goal is inference and interpretation of coefficients, you might want to include the main effects whenever the interaction is included. For prediction, you might choose to keep the model space larger (by allowing the interaction terms to be included without main effects) for more optimal prediction.

This relationship between the main effects and interactions is controlled by the `heredity()` option. By default, `heredity(strong)` is assumed, which does not allow the interactions to be in a regression model without all corresponding main effects. You can specify `heredity(weak)`, which will include at least one of the main effects together with the interaction. Or you can specify `heredity(none)`, in which case the interactions and main effects will be included in a model independently.

For instance, consider binary predictors `a` and `b`. Under strong heredity,

```
. bmaregress y i.a i.b i.a#i.b, heredity(strong)
```

will consider five models that include 1) only intercept, 2) `i.a`, 3) `i.b`, 4) `i.a` and `i.b`, and 5) `i.a`, `i.b`, and `i.a#i.b`. Note that the intercept will also be included in all models.

Under weak heredity,

```
. bmaregress y i.a i.b i.a#i.b, heredity(weak)
```

will additionally consider two more models that include 6) `i.a` and `i.a#i.b` and 7) `i.b` and `i.a#i.b`.

With no heredity, `heredity(none)`, a model with only an interaction term `i.a#i.b` will also be considered. This gives us a total of eight models.

Getting started examples

In our first series of examples, we use a small dataset, `performance`, surveying the employee satisfaction with their supervisors in a large financial organization. The dataset is adapted from Chatterjee and Hadi (2012, sec. 3.3). It contains 7 variables and 30 observations, representing 30 different departments in the surveyed organization.

```
. use https://www.stata-press.com/data/r19/performance
(Data on employee satisfaction with supervisor)

. describe

Contains data from https://www.stata-press.com/data/r19/performance.dta
 Observations:            30                  Data on employee satisfaction
                                               with supervisor
    Variables:             7                  23 Feb 2025 12:57
                                              (_dta has notes)
```

Variable name	Storage type	Display format	Value label	Variable label
rating	byte	%8.0g		Overall rating of supervisor's performance
complaints	byte	%8.0g		Handling of employee complaints
privileges	byte	%8.0g		Does not allow special privileges
learning	byte	%8.0g		Opportunity to learn new things
raises	byte	%8.0g		Raises based on performance
critical	byte	%8.0g		Supervisor is too critical
advance	byte	%8.0g		Rating of employee's advancement

```
Sorted by:
```

The outcome of interest is the overall department rating with values in the 0 to 100 range. The observed values for the dependent variable `rating` are between 40 and 85. They are computed based on the percentage of favorable survey responses from each department. The other variables record the percentages of favorable responses to each of six survey questions. Although all variables have integer values, we regard them as continuous and model the response using a linear regression. In their analysis of this dataset in Chatterjee and Hadi (2012), the authors conclude that the `complaints` and `learning` variables explain most of the variation in the response. Let's apply BMA to this dataset.

▷ Example 1: BMA linear regression (model enumeration)

Using the `performance` dataset described above, we apply BMA to `rating` with all other variables as predictors by using the `bmaregress` command. We use *varlist* notation `complaints-advance` to refer to all variables in the dataset between `complaints` and `advance`.

```
. bmaregress rating complaints-advance
Enumerating models ...
Computing model probabilities ...
Bayesian model averaging                      No. of obs        =        30
Linear regression                             No. of predictors =         6
Model enumeration                                      Groups =          6
                                                       Always =          0
Priors:                                       No. of models     =        64
  Models: Beta-binomial(1, 1)                     For CPMP >= .9 =        10
   Cons.: Noninformative                      Mean model size   =     1.699
   Coef.: Zellner's g
       g: Benchmark, g = 36                   Shrinkage, g/(1+g) =   0.9730
  sigma2: Noninformative                      Mean sigma2       =    52.302
```

rating	Mean	Std. dev.	Group	PIP
complaints	.7052859	.1224289	1	.99973
learning	.0603014	.1285281	3	.25249
advance	-.0167921	.073883	6	.13148
privileges	-.0074174	.0488635	2	.10998
raises	.0070789	.0670475	4	.10642
critical	.0009713	.0437848	5	.098534
Always				
_cons	14.8472	7.874219	0	1

```
Note: Coefficient posterior means and std. dev. estimated from 64 models.
Note: Default priors are used for models and parameter g.
```

The BMA model summary and regression results are displayed in the header and estimation table, which we describe in detail below.

Header. The header reports 30 observations and $p = 6$ predictors. By default, `Model enumeration` is used, so `bmaregress` visited all possible $2^6 = 64$ models. Of those models, 10 contribute to the cumulative PMP (CPMP) of at least 0.9. The mean model size is 1.699, which suggests that on average, the models included roughly two predictors.

The default priors are as described in *Using the bmaregress command*. `bmaregress` always uses the same noninformative priors for the constant and error variance, which are proportional to $1/\sigma^2$, and a Zellner's g-prior for regression coefficients. But there are several options for the model and parameter g priors. The default model prior is `Beta-binomial(1, 1)`, which assigns an equal probability for each model size. The default g-prior is the `Benchmark` prior with a fixed value for $g = \max(n, p^2) = \max(30, 36) = 36$, following Fernández, Ley, and Steel (2001a). This corresponds to the shrinkage parameter $\delta = g/(1 + g) = 0.973$, where $\delta = 1$ means no shrinkage and $\delta = 0$ means complete shrinkage (a coefficient is essentially forced to be zero). In this example, we assume little shrinkage a priori. But we explore the effect of different priors on these data in examples 10, 11, and 12.

From the header, the posterior mean estimate of the error variance, `Mean sigma2`, is 52.3.

`bmaregress` uses model enumeration by default when there are few predictors or, more precisely, groups of predictors (no more than 12) and when g is fixed.

Estimation table. The estimation table reported by bmaregress includes estimates of posterior means and standard deviations of coefficients for each predictor, as well as predictors' estimated posterior inclusion probabilities, PIPs. In the case of enumeration, as in our example, these estimates are based on analytical formulas and thus are exact. The table also reports the grouping information—each variable is its own group in our example.

The complaints predictor has, by far, the highest PIP of all predictors—the total posterior probability of the models that include complaints is almost 1, or 0.99973 to be exact. It is followed by learning with a PIP of 0.25. All other predictors have lower PIPs between 0.13 and 0.1. Both variables have positive posterior mean coefficients. Thus, we can conclude that a better handling of complaints, complaints, and more opportunities to learn new things, learning, appear to positively affect the rating scores. These findings agree with Chatterjee and Hadi (2012).

In a BMA regression, the constant term is always included, so it is displayed in a separate equation labeled Always. If we specify any other predictors to be included in all models, they will be listed in this same equation. The intercept or, more generally, the Always group does not affect the grouping information.

Notes. The command concludes with a couple of notes. The first note reminds us that the reported posterior means and standard deviations are estimated from 64 visited models. We need to interpret these averaged estimates thoughtfully; see *Interpretation of BMA regression coefficients*. We are also reminded that the default priors were used for models and parameter *g*; see *Priors* in *Introduction to BMA for linear regression* and discussion about the default priors in *Using the bmaregress command*. With many predictors, you might also see a note about some predictors being omitted from the output because of their PIP being less than 0.01. You can use the allcoef option (or, equivalently, pipcutoff(0)) to see all predictors or specify a higher cutoff in pipcutoff() to further restrict the list of displayed predictors.

Displaying results in input order. By default, bmaregress displays results according to PIP, from highest to lowest. You can use the inputorder option to display results in the input variable order. This might be useful when you want to compare results with other Stata commands such as regress.

```
. bmaregress, noheader inputorder
```

rating	Mean	Std. dev.	Group	PIP
complaints	.7052859	.1224289	1	.99973
privileges	-.0074174	.0488635	2	.10998
learning	.0603014	.1285281	3	.25249
raises	.0070789	.0670475	4	.10642
critical	.0009713	.0437848	5	.098534
advance	-.0167921	.073883	6	.13148
Always				
_cons	14.8472	7.874219	0	1

Note: Coefficient posterior means and std. dev. estimated from 64 models.
Note: Default priors are used for models and parameter *g*.

Displaying PIP. Instead of replaying all the results from bmaregress, we can display only the PIP estimates by using the bmastats pip command.

```
. bmastats pip
Posterior inclusion probability (PIP)

No. of obs          =        30
No. of predictors   =         6
          Groups =              6
          Always =              0
        Reported =              6
No. of models       =        64
Mean model size     =     1.699
```

	PIP	Group
complaints	.99973	1
learning	.25249	3
advance	.13148	6
privileges	.10998	2
raises	.10642	4
critical	.098534	5
Always		
_cons	1	0

```
Note: Using analytical PMPs.
```

bmastats pip is particularly useful if we would like to see PIP only for a subset of predictors.

```
. bmastats pip complaints learning
Posterior inclusion probability (PIP)

No. of obs          =        30
No. of predictors   =         6
          Groups =              6
          Always =              0
        Reported =              2
No. of models       =        64
Mean model size     =     1.699
```

	PIP	Group
complaints	.99973	1
learning	.25249	3
Always		
_cons	1	0

```
Note: Using analytical PMPs.
```

Saving BMA results. Before we proceed, let's store our current estimation results. Similarly to other Bayesian estimation commands, we first need to save BMA simulation or, in this example, enumeration results in a dataset. We can then use `estimates store` or `estimates save` (see [R] **estimates**) to store BMA estimation results.

```
. bmaregress, saving(bmadata_enum)
note: file bmadata_enum.dta saved.
. estimates store bmareg
```

We saved the BMA enumeration results in the bmadata_enum dataset, and we stored the BMA estimation results as bmareg. Remember to remove the generated dataset when it is no longer needed; see example 14.

◁

▷ Example 2: Exploring high-posterior probability models

Exploring models with high PMPs is an integral part of BMA inference. We can use `bmastats models` to explore individual models after bmaregress.

```
. bmastats models
Computing model probabilities ...
```

Model summary   Number of models:
                                         Visited = 64
                                        Reported = 5

		Analytical PMP	Model size
Rank			
	1	.5556	1
	2	.1169	2
	3	.04072	2
	4	.03932	2
	5	.03804	2

Variable-inclusion summary

	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
complaints	x	x	x	x	x
learning		x			
raises			x		
privileges				x	
advance					x

Legend:
    x - estimated

By default, bmastats models lists the top five models ranked by PMP along with the table listing the included predictors for each model. The top ranked model has a PMP of 0.56 and contains only one predictor, complaints. The second-ranked model adds learning as a predictor and has a much lower PMP of 0.12. It appears that the default BMA model for these data tends to favor regression models with two predictors, one of which is always complaints.

The variable-inclusion summary can be explored more conveniently visually by using the `bmagraph varmap` command.

```
. bmagraph varmap
Computing model probabilities ...
```
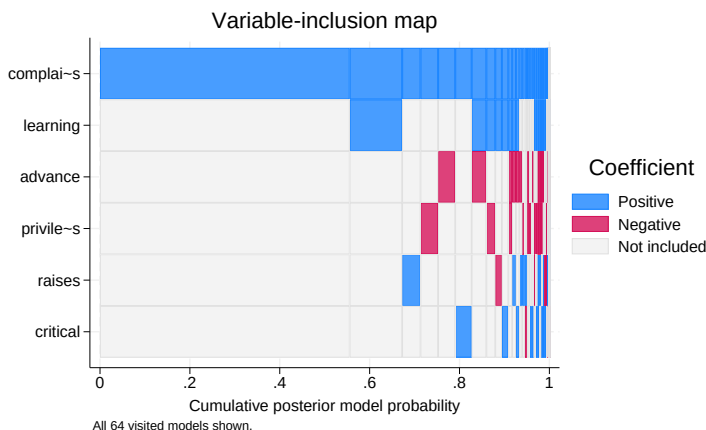

By default, the first top 100 models are plotted. In our example, this means that all 64 models are shown on the graph. We can see that the top model contains only `complaints` with a PMP or, equivalently, CPMP of roughly 0.56 (the width of the first bar). The next highest-PMP model contains both `complaints` and `learning`, with a CPMP of roughly 0.67 (the sum of widths of the first two bars). Essentially all models include `complaints`, and `learning` is the next most frequently included predictor. Predictors `raises` and `critical` appear in some models too. All of these predictors have positive coefficients, although the latter two change the sign for some models with small PMPs. Predictors `advance` and `privileges` are included in some models too, but they have negative coefficients in all of those models. See [BMA] **bmagraph varmap** for details.

In the output header, bmaregress reported that there are 10 models that contribute to the CPMP of 0.9. We can see the actual CPMPs for each model by using the cumulative() option:

```
. bmastats models, cumulative(0.9)

Computing model probabilities ...

Model summary              Number of models:
                                    Visited = 64
                                   Reported = 10
```

	Analytical CPMP	Model size
Rank		
1	.5556	1
2	.6724	2
3	.7132	2
4	.7525	2
5	.7905	2
6	.8271	2
7	.8598	3
8	.8796	3
9	.8942	3
10	.9086	3

Variable-inclusion summary

	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5	Rank 6
complaints	x	x	x	x	x	x
learning		x				
raises			x			
privileges				x		
advance					x	
critical						x

Legend:
  x – estimated

	Rank 7	Rank 8	Rank 9	Rank 10
complaints	x	x	x	x
learning	x	x	x	x
raises			x	
privileges		x		
advance	x			
critical				x

Legend:
  x – estimated

We can also plot the CPMPs by using the bmagraph pmp command with the cumulative option. We also specify options to add several reference lines to the graph:

```
. bmagraph pmp, cumulative xline(10 30) yline(.907 1) xlabel(10 30, add)
note: frequency estimates not available with model enumeration; option
      nofreqline implied.
```
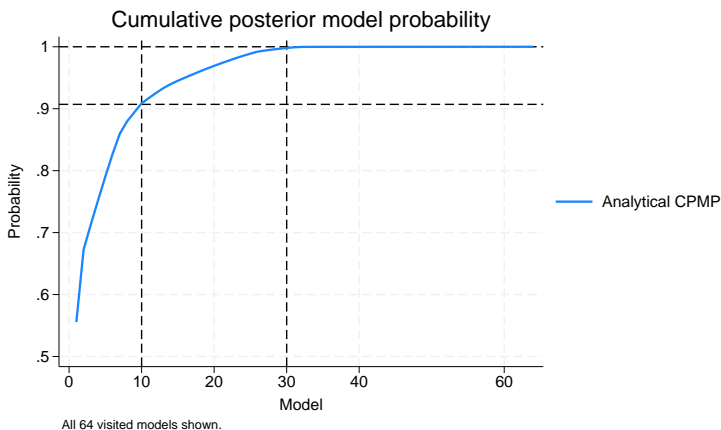

Again, the default top 100 models and thus all 64 models in our example are shown on the plot. We can see that the CPMP of about 0.9 is reached with 10 models and of 1 with roughly 30 models, so the rest of the visited models have essentially 0 PMPs and do not contribute to the averaged results.

With sampling, when an MCMC model sample is available, bmagraph pmp also reports the frequency PMP distribution estimated from the MCMC sample. This estimate is not available with model enumeration.

◁

▷ Example 3: Model-size distribution

Another aspect of exploring the model space is the model size weighted by PMP. It is useful to compare the prior and posterior distributions of the size of the visited models.

```
. bmagraph msize
note: frequency posterior model-size distribution not available.
```


Model-size distributions
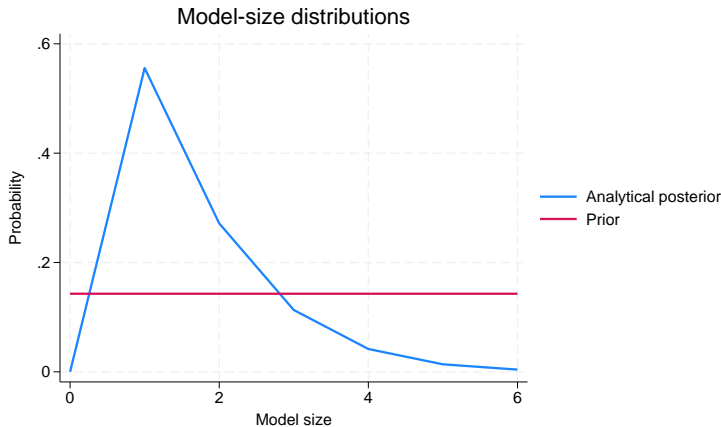
The default beta-binomial prior is uniform over the model size. Compared with the prior, the posterior prior is skewed to the left, favoring models with fewer predictors. In fact, the mode of the distribution corresponds to the model with only one predictor, which we know is `complaints`. Thus, the data suggest more parsimonious models than our prior expectation.

Instead of the entire distribution, we can use `bmastats msize` to explore the summaries of the model-size distributions.

```
. bmastats msize
Model-size summary
Number of models = 64
Model size:
  Minimum = 0
  Maximum = 6
```

	Mean	Median
Prior		
Analytical	3.0000	3
Posterior		
Analytical	1.6986	1

Note: Frequency summaries not available.

The posterior mean size is the same as the one reported by `bmaregress` in the header. The posterior mean model size of 1.7 is smaller than the assumed prior mean model size of 3.

◁

▷ Example 4: Posterior distributions of regression coefficients

The `bmaregress` command computes and reports only the posterior means and standard deviations of the regression coefficients. But we can use `bmagraph coefdensity` to plot the entire posterior densities for the regression coefficients.

For a fixed g, as in our example, the posterior distributions for the regression coefficients can be computed analytically, so we can use `bmagraph coefdensity` directly after `bmaregress`. With a random g, the analytical computation is not available, and the MCMC sampling is needed to approximate these distributions. In that case, the `bmacoefsample` command needs to be run first to obtain posterior samples of regression coefficients.

Let's look at the distributions of regression coefficients for `complaints` and `learning`. Similarly to other Stata Bayesian commands, we use the curly-braces notation to refer to model parameters—the regression coefficients here:

```
. bmagraph coefdensity {complaints}
```
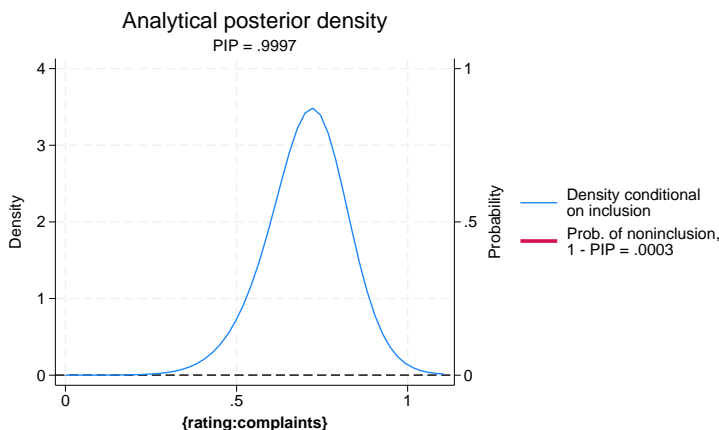

The posterior density of a regression coefficient is a mixture of a point mass at zero, which represents the probability of not being included in the model and a continuous density conditional on being included. For coefficient {complaints} (or using its full name {rating:complaints}), the probability of non-inclusion is very low, 0.0003, so the red line that represents it is not even visible on the graph. So the posterior density of {complaints} is essentially a continuous density, with a mean of roughly 0.7 and with a slight skewness to the left. Most of the mass of the distribution is between roughly 0.025 and 1.1, away from 0.

Let's plot the posterior density for {learning}:

```
. bmagraph coefdensity {learning}
```
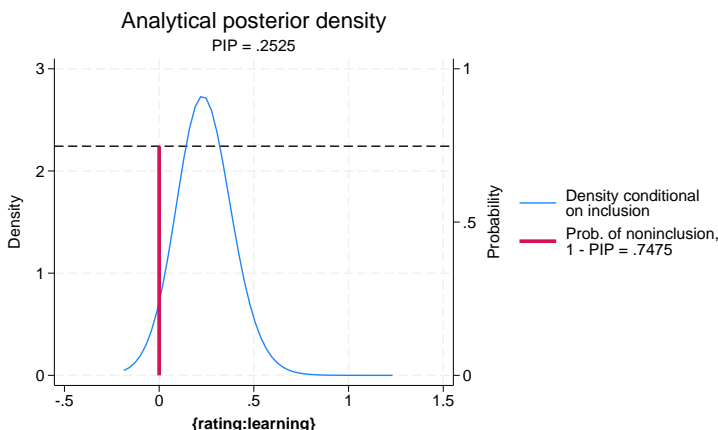

The probability of inclusion for {learning} is about 0.25, so we see the red vertical line with the horizontal reference line at a noninclusion probability of roughly $1 - 0.25 = 0.75$. Conditional on the inclusion, the continuous density has its mass between roughly -0.2 and 0.7, which includes zero.

The posterior means and standard deviations reported by bmaregress are computed with respect to the above mixtures of distributions.

◁

▷ Example 5: Credible intervals

The analytical formulas for posterior credible intervals (CrIs) are not as straightforward as for the posterior means and standard deviations, so bmaregress does not report them by default. But they can be computed based on a sample from the posterior distributions of regression coefficients. The bmacoefsample command (see [BMA] **bmacoefsample**) generates such samples. Once the posterior samples of regression coefficients are available, we can use a standard Bayesian postestimation command, bayesstats summary (see [BAYES] **bayesstats summary**), to report the posterior summaries for regression coefficients, including CrIs. To run bmacoefsample, we must first save the MCMC simulation dataset from bmaregress, which we already did in example 1.

```
. bmacoefsample, rseed(18)

Simulation (10000): ....5000....10000 done

. bayesstats summary
```

Posterior summary statistics   MCMC sample size =   10,000

					Equal-tailed	
	Mean	Std. dev.	MCSE	Median	[95% cred. interval]	
rating						
complaints	.7046698	.1224187	.001224	.7098586	.4528384	.9331236
privileges	-.0076884	.0498252	.000491	0	-.1663752	.0341648
learning	.0601684	.1283288	.001288	0	0	.4350166
raises	.0070129	.069632	.000675	0	-.077316	.2114686
critical	.0009327	.0442536	.000443	0	-.08034	.1035208
advance	-.016395	.073175	.000732	0	-.2646079	.0299074
_cons	14.92624	7.832332	.078323	14.97213	-1.027763	30.11586
sigma2	52.113	14.80086	.148009	49.78668	30.43213	87.38376
g	36	0	0	36	36	36

The results above are posterior summaries based on the MCMC samples of model parameters from their
respective posterior distributions. The MCMC estimates of posterior means and standard deviations for
regression coefficients are very similar to the exact values reported by `bmaregress`. But `bayesstats
summary` additionally reports CrIs. The equal-tailed CrIs are reported by default, but we can request the
highest posterior density (HPD) CrIs instead:

```
. bayesstats summary {complaints} {learning}, hpd
```

Posterior summary statistics   MCMC sample size =   10,000

rating	Mean	Std. dev.	MCSE	Median	HPD [95% cred. interval]	
complaints	.7046698	.1224187	.001224	.7098586	.4584306	.9362436
learning	.0601684	.1283288	.001288	0	0	.3942063

We used the `hpd` option to report HPD CrIs and requested results only for regression coefficients
`{complaints}` and `{learning}`. (To distinguish between the actual variable names and the regres-
sion coefficients associated with those variables, `bayesstats summary` uses the {} specification.)

◁

➢ Example 6: Comparison with standard linear regression

Let's redisplay our earlier BMA results here for convenience:

```
. bmaregress, noheader inputorder
```

rating	Mean	Std. dev.	Group	PIP
complaints	.7052859	.1224289	1	.99973
privileges	-.0074174	.0488635	2	.10998
learning	.0603014	.1285281	3	.25249
raises	.0070789	.0670475	4	.10642
critical	.0009713	.0437848	5	.098534
advance	-.0167921	.073883	6	.13148
Always				
_cons	14.8472	7.874219	0	1

Note: Coefficient posterior means and std. dev. estimated from 64 models.
Note: Default priors are used for models and parameter g.

We can compare our BMA findings with the results from a standard linear regression (see [R] **regress**):

```
. regress rating complaints-advance
```

Source	SS	df	MS			
Model	3147.96634	6	524.661057			
Residual	1149.00032	23	49.9565359			
Total	4296.96667	29	148.171264			

Number of obs	=	30
F(6, 23)	=	10.50
Prob > F	=	0.0000
R-squared	=	0.7326
Adj R-squared	=	0.6628
Root MSE	=	7.068

rating	Coefficient	Std. err.	t	P>|t|	[95% conf. interval]	
complaints	.6131876	.1609831	3.81	0.001	.2801687	.9462066
privileges	-.0730501	.1357247	-0.54	0.596	-.3538181	.2077178
learning	.3203321	.1685203	1.90	0.070	-.0282787	.668943
raises	.0817321	.2214777	0.37	0.715	-.3764293	.5398936
critical	.0383814	.1469954	0.26	0.796	-.2657018	.3424647
advance	-.2170567	.1782095	-1.22	0.236	-.5857111	.1515977
_cons	10.78708	11.58926	0.93	0.362	-13.18713	34.76128

As expected, the coefficient estimates for complaints are similar between the two commands, because complaints was included in almost all models by bmaregress. The PIP of 0.9997 is comparable "in spirit" with the low p-value of 0.001 from regress, which suggests that complaints is an important predictor of rating. Unlike the p-value, however, we can interpret PIP as how likely (or unlikely, $1 - $ PIP) it is for complaints to be included in the model. In general, we would expect the coefficient estimates between bmaregress and regress to be similar for predictors with high PIP (in the absence of collinearity).

For regression coefficients with lower PIP, we see the effect of shrinking toward zero by bmaregress, which is reasonable—one would expect that the effect of weak predictors should be downweighted. For example, the coefficient for learning from bmaregress is 0.06, while the one from regress is 0.32.

We can also compare posterior standard deviations from bmaregress with standard errors from regress. Both represent the uncertainty in the coefficient estimates. The Std. dev. for complaints from bmaregress is 0.12 and the Std. err. from regress is 0.16. Unlike standard errors, the posterior standard deviations additionally incorporate the uncertainty about the inclusion of the specified predictors in the model.

The 95% equal-tailed CrI for `complaints` from example 5 is $[0.45, 0.93]$ and is narrower than the 95% confidence interval (CI) of $[0.28, 0.95]$. Unlike the CI, CrI can be interpreted as the range to which the coefficient for `complaints` belongs with the 0.95 probability. For {`learning`}, the difference between CrI and CI is larger, as expected for predictors with lower PIPs.

If we force all variables to always be in the model, which we can do by specifying all the predictors in a group with the `always` suboption, our BMA model will mimic the regression results more closely.

```
. bmaregress rating (complaints-advance, always)
Enumerating models ...
Computing model probabilities ...
Bayesian model averaging                    No. of obs       =       30
Linear regression                           No. of predictors =       6
Model enumeration                                     Groups =        0
                                                      Always =        6
Priors:                                     No. of models    =        1
  Models: Beta-binomial(1, 1)                    For CPMP >= .9 =      1
   Cons.: Noninformative                   Mean model size    =   6.000
   Coef.: Zellner's g
       g: Benchmark, g = 36                Shrinkage, g/(1+g) = 0.9730
  sigma2: Noninformative                   Mean sigma2        =  45.707
```

rating	Mean	Std. dev.	Group	PIP
Always				
complaints	.596615	.1518883	0	1
privileges	-.0710758	.1280569	0	1
learning	.3116745	.1589997	0	1
raises	.0795232	.2089652	0	1
critical	.0373441	.1386909	0	1
advance	-.2111903	.1681415	0	1
_cons	12.24238	10.9364	0	1

```
Note: Coefficient posterior means and std. dev. estimated from 1 model.
Note: Default priors are used for models and parameter g.
```

There is only one model and all predictors have PIPs of 1. The above model corresponds to the BMA model with the highly informative model prior distribution that assigns the probability of 1 to the full model and 0 to all other models. Equivalently, this prior assumes that the probability of inclusion for each predictor is exactly 1. We investigate the effect of priors on the results in more detail in examples 10, 11, and 12. The above model is also equivalent to a Bayesian linear regression with noninformative priors for the error variance and intercept and with a Zellner's g-prior for the regression coefficients.

◁

▷ Example 7: Jointness of predictors

In BMA, in addition to exploring the importance of individual predictors, we can investigate whether this importance depends on the presence of other predictors across the models. The tendency of the predictors to appear together or separately in the models is known as jointness. The `bmastats jointness` command computes various jointness measures for pairs of predictors.

As a quick demonstration, let's investigate the jointness between, for instance, `complaints` and `raises`.

```
. bmastats jointness complaints raises
Variables: complaints raises
```

	Jointness
Doppelhofer–Weeks	−2.388135
Ley–Steel type 1	.1062777
Ley–Steel type 2	.1189158
Yule's Q	−.831836

Notes: Using analytical PMPs. See thresholds.

`bmastats jointness` reports four measures of jointness. Based on the thresholds described in *Remarks and examples* in [BMA] **bmastats jointness**, such as $−2.39 < −2$ for the Doppelhofer–Weeks measure, all four measures agree that the two predictors are disjoint or the so-called substitutes. This means that when one is included in the model, the inclusion of the other does not provide any additional information for explaining the outcome. See [BMA] **bmastats jointness** for more information and example 20 for another example.

◁

▷ Example 8: Exploring model space using MC3 (model sampling)

For models with many predictors, complete enumeration of all possible models becomes infeasible. The alternative is to use MCMC to sample the model space. The MCMC model composition sampling, MC3 (Madigan and York 1995), is commonly used in BMA to explore the model space.

In bmaregress, MC3 sampling can be requested by specifying the sampling option. The MC3 sampling is actually the default when the number of predictors (or groups of predictors) is greater than 12.

Because MC3 sampling is stochastic, for reproducibility, we need to specify a random-number seed, for example, rseed(18). The sampling includes the burn-in phase, during which the model space is explored for models with high posterior probabilities, followed by an active sampling phase, during which all visited models are saved and accounted for during estimation.

```
. bmaregress rating complaints-advance, sampling rseed(18)

Burn-in ...
Simulation ...
Computing model probabilities ...

Bayesian model averaging                    No. of obs        =       30
Linear regression                           No. of predictors =        6
MC3 sampling                                         Groups   =        6
                                                     Always   =        0
                                            No. of models      =       32
                                               For CPMP >= .9  =       10
Priors:                                     Mean model size    =    1.699
  Models: Beta-binomial(1, 1)               Burn-in            =    2,500
   Cons.: Noninformative                    MCMC sample size   =   10,000
   Coef.: Zellner's g                       Acceptance rate    =   0.2417
       g: Benchmark, g = 36                 Shrinkage, g/(1+g) =   0.9730
  sigma2: Noninformative                    Mean sigma2        =   52.292
Sampling correlation = 0.9990
```

rating	Mean	Std. dev.	Group	PIP
complaints	.705479	.1218881	1	1
learning	.0601919	.1282869	3	.25234
advance	-.0167514	.0737415	6	.13141
privileges	-.0074265	.048844	2	.10996
raises	.0069949	.0666406	4	.10629
critical	.0009699	.0437742	5	.098526
Always				
_cons	14.84478	7.871046	0	1

Note: Coefficient posterior means and std. dev. estimated from 32 models.
Note: Default priors are used for models and parameter *g*.

bmaregress now uses MC3 sampling. The header additionally reports standard MCMC summaries, including the number of burn-in and MCMC iterations and the acceptance rate. The sampling correlation is also reported. It is used to check convergence; see *Convergence of BMA*.

The reported sampling correlation, 0.9990, is very close to 1, which suggests that the sampling algorithm has converged and the sample of models approximates the true model posterior distribution well. Consequently, the posterior mean and standard deviation estimates are very close to their exact values obtained by enumeration in example 1. The same is true for the PIP estimates, which match their exact counterparts closely.

◁

▷ Example 9: Checking BMA convergence

A graphical summary of the MC3 convergence is provided by the bmagraph pmp command. It shows the analytical and MCMC sampling probabilities of the visited models, ordered decreasingly by their PMPs. Upon convergence, these two curves should overlap closely, as is the case in our example.

. bmagraph pmp

Posterior model probability

All 32 visited models shown.
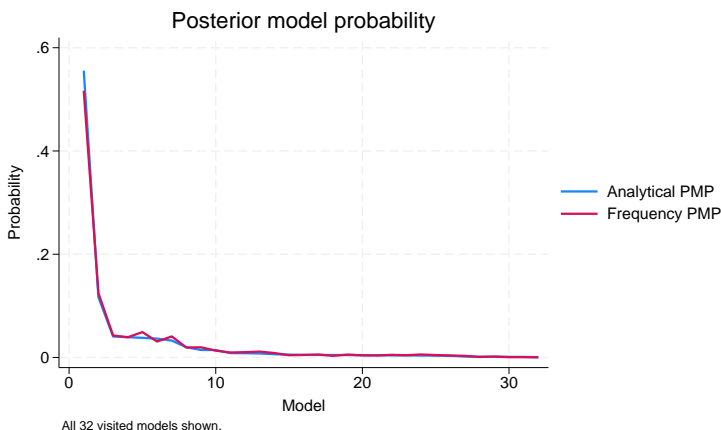
We can also explore the CPMPs by specifying the cumulative option. The analytical and sampling cumulative probabilities overlap closely too.

. bmagraph pmp, cumulative

Cumulative posterior model probability

All 32 visited models shown.
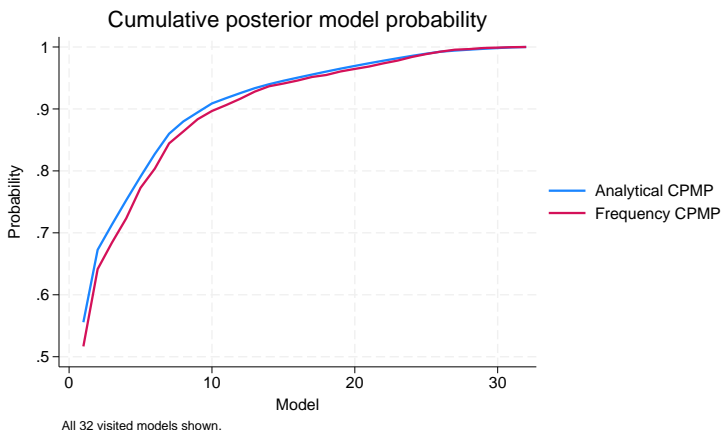
The cumulative graph also shows the contributions of the top PMP models. In our example, the top 6 models contribute to about 80% of the total probability mass on the model space.

◁

▷ Example 10: BMA linear regression using different g-priors

As we mentioned in *Using the bmaregress command*, bmaregress provides default priors for convenience. It is important to explore other available prior choices during your analysis. Here we focus on the g-priors, and we explore the model priors in example 11.

By default, bmaregress uses the benchmark prior from Fernández, Ley, and Steel (2001a), which is $g = \max(n, p^2)$, where $g = n$ corresponds to the unit-information prior and $g = p^2$ corresponds to the risk inflation criterion prior. Because the number of observations, 30, is less than the number of predictors squared, the benchmark g-prior is equivalent to using the risk inflation criterion, gprior(ric).

```
. bmaregress rating complaints-advance, gprior(ric) saving(bmadata_ric)

Enumerating models ...
Computing model probabilities ...

Bayesian model averaging                    No. of obs      =       30
Linear regression                           No. of predictors =      6
Model enumeration                                   Groups =        6
                                                    Always =        0
Priors:                                     No. of models   =       64
  Models: Beta-binomial(1, 1)                   For CPMP >= .9 =     10
   Cons.: Noninformative                    Mean model size =    1.699
   Coef.: Zellner's g
       g: Risk inflation, g = 36            Shrinkage, g/(1+g) = 0.9730
  sigma2: Noninformative                    Mean sigma2     =   52.302
```

rating	Mean	Std. dev.	Group	PIP
complaints	.7052859	.1224289	1	.99973
learning	.0603014	.1285281	3	.25249
advance	−.0167921	.073883	6	.13148
privileges	−.0074174	.0488635	2	.10998
raises	.0070789	.0670475	4	.10642
critical	.0009713	.0437848	5	.098534
Always				
_cons	14.8472	7.874219	0	1

```
Note: Coefficient posterior means and std. dev. estimated from 64 models.
Note: Default prior is used for models.

file bmadata_ric.dta saved.

. estimates store ric
```

We stored BMA estimation results for later comparison.

Let's specify the unit-information g-prior, gprior(uip):

```
. bmaregress rating complaints-advance, gprior(uip) saving(bmadata_uip)

Enumerating models ...
Computing model probabilities ...
```

Bayesian model averaging				No. of obs	=	30
Linear regression				No. of predictors	=	6
Model enumeration				Groups	=	6
				Always	=	0
Priors:				No. of models	=	64
Models: Beta-binomial(1, 1)				For CPMP >= .9	=	11
Cons.: Noninformative				Mean model size	=	1.777
Coef.: Zellner's g						
g: Unit-information, g = 30				Shrinkage, g/(1+g)	=	0.9677
sigma2: Noninformative				Mean sigma2	=	52.739

rating	Mean	Std. dev.	Group	PIP
complaints	.6988904	.1242031	1	.99969
learning	.0658469	.1334993	3	.27425
advance	-.0194169	.0791369	6	.14818
privileges	-.0083983	.0519289	2	.12391
raises	.0079323	.071638	4	.11984
critical	.0012079	.0467515	5	.11142
Always				
_cons	15.05253	7.983872	0	1

```
Note: Coefficient posterior means and std. dev. estimated from 64 models.
Note: Default prior is used for models.

file bmadata_uip.dta saved.

. estimates store uip
```

Using this prior reduces the fixed value of g from 36 to 30 and the shrinkage factor from 0.973 to 0.968. The reported posterior means are only slightly different from the ric results, which are the same as in example 1, and the PIP order of the predictors remains the same. A theoretical benefit of the unit-information prior is its consistency, in the sense that if a true data-generating model is in the considered model space, its PMP will go to one as the sample size goes to infinity (Fernández, Ley, and Steel 2001a).

Other fixed g-priors often considered in the literature are the square-root n prior, gprior(sqrtn), and the empirical Bayes (local) prior, gprior(ebl).

The square-root n prior sets $g = \sqrt{n}$, which is $g = \sqrt{30} = 5.48$ in our example.

```
. bmaregress rating complaints-advance, gprior(sqrtn) saving(bmadata_sqrtn)

Enumerating models ...
Computing model probabilities ...
```

Bayesian model averaging		No. of obs	=	30
Linear regression		No. of predictors	=	6
Model enumeration		Groups	=	6
		Always	=	0
Priors:		No. of models	=	64
Models: Beta-binomial(1, 1)		For CPMP >= .9	=	22
Cons.: Noninformative		Mean model size	=	2.818
Coef.: Zellner's g				
g: g = sqrt(n) = 5.48		Shrinkage, g/(1+g)	=	0.8456
sigma2: Noninformative		Mean sigma2	=	64.848

rating	Mean	Std. dev.	Group	PIP
complaints	.585586	.1441692	1	.99743
learning	.1134179	.1641032	3	.48829
advance	-.0512107	.126477	6	.36728
privileges	-.0201474	.0851126	2	.3289
raises	.0203943	.1270875	4	.32382
critical	.0053258	.0832724	5	.31268
Always				
_cons	20.79312	9.495701	0	1

```
Note: Coefficient posterior means and std. dev. estimated from 64 models.
Note: Default prior is used for models.
file bmadata_sqrtn.dta saved.
. estimates store sqrtn
```

This prior substantively reduces parameter g from 36 to 5.48 and the shrinkage factor from 0.973 to 0.846. The posterior mean error variance is noticeably higher. The posterior mean estimates of coefficients are somewhat different. The coefficient for complaints is slightly smaller (0.6 versus 0.7), the coefficient for learning is larger (0.11 versus 0.06), and the other coefficients are about 10 times larger, but they are still close to 0. The PIPs of all predictors except complaints are much higher too: about twice as high for learning and three times as high for the other predictors. These results appear to be closer to those from regress in example 6. Under this more informative prior, the BMA model favors models with more predictors, which is indicated by the increased posterior mean model size from 1.7 to 2.8. We explore these differences further in examples 11 and 12.

The local empirical Bayes prior uses a model-specific g estimated from the data for each model.

```
. bmaregress rating complaints-advance, gprior(ebl) saving(bmadata_ebl)

Enumerating models ...
Computing model probabilities ...

Bayesian model averaging                    No. of obs        =       30
Linear regression                           No. of predictors =        6
Model enumeration                                     Groups  =        6
                                                      Always  =        0
Priors:                                     No. of models     =       64
  Models: Beta-binomial(1, 1)                   For CPMP >= .9 =       14
   Cons.: Noninformative                    Mean model size   =    1.872
   Coef.: Zellner's g
       g: Empirical Bayes (local)           Shrinkage, g/(1+g) = 0.9696
  sigma2: Noninformative                    Mean sigma2       =   52.442
```

rating	Mean	Std. dev.	Group	PIP
complaints	.6989413	.1302979	1	.99949
learning	.0694007	.1373262	3	.28635
advance	-.0234091	.0869118	6	.16924
privileges	-.0096619	.0563268	2	.14424
raises	.0092523	.0797441	4	.14018
critical	.001859	.0519251	5	.13209
Always				
_cons	14.95335	8.048125	0	1

```
Note: Coefficient posterior means and std. dev. estimated from 64 models.
Note: Default prior is used for models.
```
file **bmadata_ebl.dta** saved.
```
. estimates store ebl
```

The results are again similar to the default ones and those from other priors, except the previous square-root n prior.

We can also specify one of the random priors, for example, a hyper-g prior with hyperparameter 3:

```
. bmaregress rating complaints-advance, gprior(hyperg 3) saving(bmadata_hyperg3)

Burn-in ...
Simulation ...
Computing model probabilities ...
```

Bayesian model averaging   No. of obs   =   30
Linear regression   No. of predictors =   6
MC3 and adaptive MH sampling   Groups =   6
                                                            Always =   0
                                                 No. of models   =   38
                                                    For CPMP >= .9 =   17
Priors: Mean model size   =   2.096
  Models: Beta-binomial(1, 1)   Burn-in   =   2,500
  Cons.: Noninformative   MCMC sample size   =   10,000
  Coef.: Zellner's g   Acceptance rate   =   0.5895
      g: Hyper-g(3)
  sigma2: Noninformative   Mean sigma2   =   55.351

Sampling correlation = 0.9935

rating	Mean	Std. dev.		Group	PIP
complaints	.6741553	.1420621		1	.9984
learning	.0765859	.1437933		3	.316
advance	−.0307362	.099772		6	.2259
privileges	−.012216	.0648199		2	.1925
raises	.0126483	.0932019		4	.182
critical	.0027903	.0617168		5	.1814
Always					
_cons	16.36026	8.877669		0	1

Note: Coefficient posterior means and std. dev. estimated from 38 models.
Note: Default prior is used for models.

	Mean	Std. dev.	MCSE	Median	Equal-tailed [95% cred. interval]	
g	843.7276	27512.51	581.21	23.32369	3.531358	587.8747
Shrinkage	.9410376	.0591481	.00196	.9588878	.7793152	.9983018

file **bmadata_hyperg3.dta** saved.

```
. estimates store hyperg3
```

Or the robust prior:

```
. bmaregress rating complaints-advance, gprior(robust) rseed(18)
> saving(bmadata_robust)

Burn-in ...
Simulation ...
Computing model probabilities ...
```

```
Bayesian model averaging                  No. of obs        =        30
Linear regression                         No. of predictors =         6
MC3 and adaptive MH sampling                       Groups   =         6
                                                   Always    =         0
                                          No. of models     =        34
                                             For CPMP >= .9  =        12
Priors:                                   Mean model size   =     1.734
  Models: Beta-binomial(1, 1)             Burn-in           =     2,500
   Cons.: Noninformative                  MCMC sample size  =    10,000
   Coef.: Zellner's g                     Acceptance rate   =    0.4232
       g: Robust
  sigma2: Noninformative                  Mean sigma2       =    53.095
Sampling correlation = 0.9994
```

rating	Mean	Std. dev.	Group	PIP
complaints	.7000463	.1273543	1	.9998
learning	.0594904	.1286095	3	.25
advance	−.0192712	.0797935	6	.1503
raises	.0079416	.0727859	4	.1201
privileges	−.0072591	.0487009	2	.1069
critical	.0014397	.0466476	5	.1067
Always				
_cons	15.24911	7.988166	0	1

Note: Coefficient posterior means and std. dev. estimated from 34 models.
Note: Default prior is used for models.

	Mean	Std. dev.	MCSE	Median	Equal-tailed [95% cred. interval]	
g	152.668	1968.132	43.5265	33.81024	8.205076	610.6026
Shrinkage	.9656427	.0276071	.001234	.9712728	.8913639	.9983649

```
file bmadata_robust.dta saved.
```

```
. estimates store robust
```

With a random g, the sampling algorithm is a modified MC3 that uses MC3 for model exploration and an adaptive MH for sampling g. Also, the parameter g and shrinkage are now stochastic, and their posterior summaries are displayed at the bottom of the estimation table.

By specifying a hyperprior for g, we let the data guide the value for g instead of using a fixed value. For instance, for the robust prior, the posterior mean estimate for g of roughly 153 is substantially larger than the values we assumed in previous examples. But it also has a high variability, which is not surprising for a sample of only 30 observations. With this prior, although g is much higher, the results are still similar, and our conclusions remain unchanged.

Overall, the results in this example appear to be fairly robust to a variety of g-priors, except the square-root n prior. See example 12 for a comparison of these priors by using the log predictive-score. We explore the sensitivity of the results to different priors further in example 11.

See Fernández, Ley, and Steel (2001a) for a comparison of various fixed g priors, Liang et al. (2008) for a discussion of local empirical Bayes, hyper-g, and hyper-g/n priors, and Ley and Steel (2012) and Porwal and Raftery (2022b) for a comparison of many other priors.

In this example, we generated several MCMC simulation datasets, which you might want to erase at the end of the analysis; see example 14.

◁

▷ Example 11: Sensitivity analysis of model priors

In example 10, we considered several g-priors. In this example, we explore several model priors.

As a reference, we refit our default bmaregress model, but this time specifying the default Beta-binomial(1, 1) prior explicitly. We also save the BMA MCMC results in bmadata_betabinom.dta and store estimation results as betabinom.

```
. bmaregress rating complaints-advance, mprior(betabinomial)
> saving(bmadata_betabinom)

Enumerating models ...
Computing model probabilities ...

Bayesian model averaging                    No. of obs        =        30
Linear regression                           No. of predictors =         6
Model enumeration                                   Groups =           6
                                                    Always =           0
Priors:                                     No. of models     =        64
  Models: Beta-binomial(1, 1)                   For CPMP >= .9 =        10
   Cons.: Noninformative                    Mean model size   =     1.699
   Coef.: Zellner's g
       g: Benchmark, g = 36                 Shrinkage, g/(1+g) =    0.9730
  sigma2: Noninformative                    Mean sigma2       =    52.302
```

rating	Mean	Std. dev.	Group	PIP
complaints	.7052859	.1224289	1	.99973
learning	.0603014	.1285281	3	.25249
advance	−.0167921	.073883	6	.13148
privileges	−.0074174	.0488635	2	.10998
raises	.0070789	.0670475	4	.10642
critical	.0009713	.0437848	5	.098534
Always				
_cons	14.8472	7.874219	0	1

```
Note: Coefficient posterior means and std. dev. estimated from 64 models.
Note: Default prior is used for parameter g.
file bmadata_betabinom.dta saved.
. estimates store betabinom
```

The above prior is the uniform (noninformative) prior for the model size. Alternatively, we can assume the uniform prior for the model space, which assumes that each model is equally likely, with a probability of $1/64$ in our example. This is equivalent to assuming that each predictor has the same probability of being included in the model.

```
. bmaregress rating complaints-advance, mprior(uniform) saving(bmadata_uniform)
Enumerating models ...
Computing model probabilities ...
```

Bayesian model averaging			No. of obs	=	30
Linear regression			No. of predictors	=	6
Model enumeration			Groups	=	6
			Always	=	0
Priors:			No. of models	=	64
Models: Uniform			For CPMP >= .9	=	13
Cons.: Noninformative			Mean model size	=	2.043
Coef.: Zellner's g					
g: Benchmark, g = 36			Shrinkage, g/(1+g)	=	0.9730
sigma2: Noninformative			Mean sigma2	=	51.603

rating	Mean	Std. dev.	Group	PIP
complaints	.6903531	.1291635	1	.99968
learning	.0918832	.1486692	3	.38708
advance	−.0243892	.087595	6	.19757
privileges	−.0109805	.0591344	2	.16143
raises	.0098927	.0801092	4	.15484
critical	.0008905	.0524265	5	.14236
Always				
_cons	14.40123	8.301086	0	1

```
Note: Coefficient posterior means and std. dev. estimated from 64 models.
Note: Default prior is used for parameter g.
file bmadata_uniform.dta saved.
. estimates store uniform
```

The posterior mean model size is slightly larger, 2 versus 1.7, under the uniform prior. But the results are very similar to those using the default beta-binomial prior.

When we assume a noninformative prior, we allow the data to "speak for themselves" when estimating model parameters. Sometimes, however, the data may have little to say. This is especially common with small datasets. In this case, it might be reasonable to explore more informative priors. The performance dataset is a relatively small dataset. We have already seen some dependency of this dataset on one of the g-priors in example 10. Let's see what happens as we introduce more information about the models.

Let's continue with the beta-binomial prior but make it informative. The beta-binomial prior assumes a binomial distribution for the inclusion of predictors in a model with the same IP and assumes a beta distribution for the IP. One way to modify this prior is to specify a desired expected mean model size instead of assuming that models of all sizes are equally likely. Given that we have two predictors that stand out, we may consider a prior that gives more weight to smaller models. On the other hand, all predictors have PIP above 10%, so it may not be unreasonable to also consider larger models. Let's use beta-binomial priors with the expected mean model size of 2 and 5. We specify the mprior(betabinomial 2) and mprior(betabinomial 5) options, respectively.

```
. bmaregress rating complaints-advance, mprior(betabinomial 2)
> saving(bmadata_betabinom2)

Enumerating models ...
Computing model probabilities ...

Bayesian model averaging                     No. of obs        =       30
Linear regression                            No. of predictors =        6
Model enumeration                                     Groups =         6
                                                      Always =         0
Priors:                                      No. of models     =       64
  Models: Beta-binomial, mean = 2                 For CPMP >= .9 =       7
   Cons.: Noninformative                    Mean model size    =    1.522
   Coef.: Zellner's g
       g: Benchmark, g = 36                 Shrinkage, g/(1+g) =   0.9730
  sigma2: Noninformative                    Mean sigma2        =   52.639
```

rating	Mean	Std. dev.	Group	PIP
complaints	.7116651	.1185252	1	.99975
learning	.0458966	.1135214	3	.19841
advance	−.0105735	.0597458	6	.094173
privileges	−.0051842	.041615	2	.080238
raises	.005258	.0565843	4	.078173
critical	.0004494	.0371854	5	.071644
Always				
_cons	15.00536	7.675558	0	1

```
Note: Coefficient posterior means and std. dev. estimated from 64 models.
Note: Default prior is used for parameter g.

file bmadata_betabinom2.dta saved.

. estimates store betabinom2
```

The results are very similar to those using the default prior.

```
. bmaregress rating complaints-advance, mprior(betabinomial 5)
> saving(bmadata_betabinom5)

Enumerating models ...
Computing model probabilities ...
```

Bayesian model averaging			No. of obs	=	30
Linear regression			No. of predictors	=	6
Model enumeration			Groups	=	6
			Always	=	0
Priors:			No. of models	=	64
Models: Beta-binomial, mean = 5			For CPMP >= .9	=	15
Cons.: Noninformative			Mean model size	=	2.159
Coef.: Zellner's g					
g: Benchmark, g = 36			Shrinkage, g/(1+g)	=	0.9730
sigma2: Noninformative			Mean sigma2	=	51.509

rating	Mean	Std. dev.	Group	PIP
complaints	.6914498	.1304325	1	.99971
learning	.0923695	.1546786	3	.35955
advance	−.0362158	.104821	6	.22875
privileges	−.0138814	.0646952	2	.19699
raises	.0133366	.0926326	4	.19141
critical	.003734	.0601699	5	.1824
Always				
_cons	14.52747	8.31467	0	1

Note: Coefficient posterior means and std. dev. estimated from 64 models.
Note: Default prior is used for parameter *g*.

file **bmadata_betabinom5.dta** saved.

```
. estimates store betabinom5
```

The posterior mean model size is slightly higher, but the results are still similar to the above.

With the beta-binomial prior, the IP is allowed to be inferred from the data. It may be difficult to estimate the IP reliably with small datasets. If we want to fix this probability, we can use the binomial prior. Let's specify the binomial priors with mean model sizes of 2 and 5. With the binomial prior, we need to specify the IP. For the mean model size of 2, the corresponding IP is $2/p = 2/6 = 0.33$, and for the mean size of 5, it is $5/p = 5/6 = 0.83$.

```
. bmaregress rating complaints-advance, mprior(binomial 0.33)
> saving(bmadata_binom2)

Enumerating models ...
Computing model probabilities ...

Bayesian model averaging                    No. of obs         =        30
Linear regression                           No. of predictors  =         6
Model enumeration                                     Groups  =         6
                                                      Always  =         0
Priors:                                     No. of models      =        64
  Models: Binomial, IP = 0.33                  For CPMP >= .9  =         7
   Cons.: Noninformative                    Mean model size    =     1.561
   Coef.: Zellner's g
       g: Benchmark, g = 36                 Shrinkage, g/(1+g) =    0.9730
  sigma2: Noninformative                    Mean sigma2        =    52.561
```

rating	Mean	Std. dev.	Group	PIP
complaints	.7090898	.119952	1	.99975
learning	.0499472	.1158814	3	.22295
advance	−.0091067	.0564581	6	.096086
privileges	−.0051623	.0422812	2	.084251
raises	.0056663	.057094	4	.082806
critical	.0002176	.0379403	5	.075179
Always				
_cons	14.87536	7.771313	0	1

```
Note: Coefficient posterior means and std. dev. estimated from 64 models.
Note: Default prior is used for parameter g.

file bmadata_binom2.dta saved.

. estimates store binom2
```

The results for the binomial prior with the mean model size of 2 are again similar to the previous results.

```
. bmaregress rating complaints-advance, mprior(binomial 0.83)
> saving(bmadata_binom5)
Enumerating models ...
Computing model probabilities ...
Bayesian model averaging                       No. of obs        =       30
Linear regression                              No. of predictors =        6
Model enumeration                                       Groups  =        6
                                                        Always  =        0
Priors:                                        No. of models     =       64
  Models: Binomial, IP = 0.83                      For CPMP >= .9 =       23
   Cons.: Noninformative                       Mean model size   =    3.836
   Coef.: Zellner's g
       g: Benchmark, g = 36                     Shrinkage, g/(1+g) =  0.9730
  sigma2: Noninformative                        Mean sigma2       =   48.343
```

rating	Mean	Std. dev.	Group	PIP
complaints	.6368386	.1435817	1	.99951
learning	.2237682	.1794828	3	.80435
advance	−.1102228	.1561712	6	.61442
privileges	−.0376032	.0996713	2	.49344
raises	.0313871	.1470874	4	.47092
critical	.0112057	.0948876	5	.45346

Always				
_cons	13.47056	9.637186	0	1

```
Note: Coefficient posterior means and std. dev. estimated from 64 models.
Note: Default prior is used for parameter g.
file bmadata_binom5.dta saved.
. estimates store binom5
```

The results for the binomial prior with the mean model size of 5 are different and more similar to the results from `regress` in example 6. This is not surprising considering that this prior favors larger models—the posterior mean model size is now roughly 4.

It appears that the results are somewhat robust to the considered model priors except for `gprior(binomial 0.83)`. We evaluate these priors more formally in example 12.

◁

▷ Example 12: Comparing model fit using the log predictive-score

As we discussed in *Model diagnostics* and *Regression modeling and model space* in *Introduction to BMA for linear regression*, the definition of the model space and model diagnostics are important after BMA linear regression. Before you proceed with `bmaregress`, it is beneficial that you explore the regression diagnostics for at least the full model first. In addition to checking standard regression assumptions, you will benefit from exploring alternative regression specifications to ensure that you include all relevant predictors in the BMA model space; see [R] **regress postestimation**. After fitting `bmaregress`, you should check regression diagnostics for models with higher PMPs, such as the two models we identified in example 2. You should also check the sensitivity of the results to the default priors, as we showed in examples 10 and 11 and continue exploring in this example. And, if sampling is used, we should also check BMA convergence; see example 9.

In example 10, the choice of prior does not appear to affect the results except for one prior. Let's go ahead and compare those different prior specifications more formally by using the log predictive-score (LPS). The LPS is defined as the negative of the log of the posterior predictive density evaluated at an observation. The smaller its value, the better the model fit. The `bmastats lps` command computes LPS. We can use this command for in-sample observations used during estimation and for out-of-sample observations. The former can be used for comparing model goodness of fit, and the latter can be used for checking predictive performance of the model. We will compare the goodness of fit here.

```
. bmastats lps ric uip sqrtn ebl hyperg3 robust, compact
Log predictive-score (LPS)
Number of observations = 30
```

LPS	Mean	Minimum	Maximum
ric	3.33244	2.890314	4.429047
uip	3.3322	2.89472	4.422744
sqrtn	3.36909	2.999145	4.388221
ebl	3.331938	2.892466	4.418862
hyperg3	3.339529	2.916094	4.413267
robust	3.335522	2.897331	4.432305

```
Notes: Results using analytical and frequency PMPs.
       Result ebl has the smallest mean LPS.
```

`bmastats lps` reports that the `ebl` prior has the lowest mean LPS, but all mean LPS values (and other summaries), except `sqrtn`, are similar.

Similarly to g-priors, we can evaluate the model priors from example 10 by using the LPS.

```
. bmastats lps uniform betabinom betabinom2 betabinom5 binom2 binom5, compact
Log predictive-score (LPS)
Number of observations = 30
```

LPS	Mean	Minimum	Maximum
uniform	3.324591	2.884711	4.394576
betabinom	3.33244	2.890314	4.429047
betabinom2	3.336546	2.892654	4.445658
betabinom5	3.323516	2.885564	4.386189
binom2	3.335636	2.891732	4.438463
binom5	3.295903	2.865979	4.258077

```
Notes: Using analytical PMPs.
       Result binom5 has the smallest mean LPS.
```

The `binom5` prior, `mprior(binomial 0.83)`, with the mean model size of 5, has the smallest mean LPS. It also has the smaller mean LPS than the above `ebl` prior.

In conclusion, all considered priors identified `complaints` as a highly important predictor of `rating` and `learning` as the predictor with the next highest PIP. The actual estimates of PIPs, except for `complaints`, and posterior summaries appeared to be dependent on whether the BMA model favored the smaller or larger models. This may be explained by the presence of correlation between some of the predictors. The noninformative model priors such as `betabinomial` (the default) and `uniform` and g-priors assuming shrinkage $\delta = g/(g+1)$ closer to 1 favored smaller models for these data. The `sqrtn` g-prior and `binomial 0.83` model prior (with the mean model size of five) favored the larger models.

In the above, we considered the *g*-priors and model priors independently. But it is important to consider their combination when evaluating model performance. To explore model fit, we computed LPS using the entire estimation sample. We could have instead split our data into a training and test subsamples and evaluated the out-of-sample performance of the priors by fitting the models using the training subsample and computing LPS using the test subsample; see the examples in [BMA] **bmastats lps**, including how to use cross-validation to check the model performance. Such an approach is important for evaluating the predictive performance of the model, especially when prediction is the final goal of the analysis. But, because of the smaller sample size, this approach may not be as viable for the performance dataset.

LPS is used to evaluate the predictive performance of models in the context of BMA (Fernández, Ley, and Steel (2001b); Ley and Steel (2012)), but we can also use other diagnostic measures such as mean squared error or CrI coverage to compare model fit and its predictive performance; see example 16 and *Remarks and examples* in [BMA] **bmapredict**.

In this example, we generated several datasets with BMA results, which we can remove at the end of the analysis; see example 14.

◁

▷ Example 13: BMA predictions

In addition to identifying high PMP models and important predictors, BMA is also used for prediction that accounts for model uncertainty. In fact, BMA was originated in the context of prediction. In prediction, it is particularly important that the considered model space is as rich as possible to obtain accurate predictions.

You can use the bmapredict command to compute posterior predictive summaries such as mean, standard deviations, and CrIs, or you can simulate an entire predictive sample; see [BMA] **bmapredict**.

For posterior means and standard deviations, analytical expressions are available with fixed *g*, so you can use bmapredict directly after bmaregress to compute those summaries.

Recall our BMA model from example 1. Let's compute posterior predictive means for rating based on this model.

```
. estimates restore bmareg
(results bmareg are active now)
. bmapredict pmean, mean
note: computing analytical posterior predictive means.
```

Let's now compute the 95% equal-tailed predictive CrIs. The analytical expressions for these are not as straightforward and require approximation. But we can compute them based on a predictive MCMC sample. To generate this sample, we must first obtain the MCMC sample of model parameters. This is done by bmacoefsample. To use bmapredict, we must also save the MCMC model parameter sample with bmacoefsample.

```
. bmacoefsample, saving(bmacoef) rseed(18)
Simulation (10000): ....5000....10000 done
file bmacoef.dta saved.
. bmapredict cri_l cri_u, cri
note: computing credible intervals using simulation.
Computing predictions ...
```

Let's summarize the predictions:

```
. summarize rating pmean cri_l cri_u
    Variable │        Obs        Mean    Std. dev.         Min         Max
─────────────┼──────────────────────────────────────────────────────────
      rating │         30    64.63333    12.17256          40          85
       pmean │         30    64.63333    9.788015    43.71523    82.16643
       cri_l │         30    49.68274       9.812    27.45753    66.24698
       cri_u │         30    79.67859    9.794941    59.52461    98.29689
```

The predicted summaries look reasonable for `rating`. The mean of posterior predictive means over observations is 64.63 and matches the mean of `rating`. The mean posterior predictive lower 95% credible bound is 49.7 and the upper is 79.7. Also see *Remarks and examples* in [BMA] **bmapredict** for how to evaluate the quality of predictions.

◁

▷ Example 14: Cleanup after BMA analysis

With BMA, we often need to generate and save many datasets that contain simulation summaries. Remember to remove them whenever you no longer need them. We remove the following BMA datasets generated by *Getting started examples*:

```
. erase bmadata_enum.dta
. erase bmadata_ric.dta
. erase bmadata_uip.dta
. erase bmadata_sqrtn.dta
. erase bmadata_ebl.dta
. erase bmadata_hyperg3.dta
. erase bmadata_robust.dta
. erase bmadata_betabinom.dta
. erase bmadata_betabinom2.dta
. erase bmadata_betabinom5.dta
. erase bmadata_binom2.dta
. erase bmadata_binom5.dta
. erase bmadata_uniform.dta
. erase bmacoef.dta
```

◁

BMA predictive performance for the USA crime rate data

Consider a study from 1960 on the factors influencing criminal activity in the USA (Ehrlich 1973; Becker 1968; and Brier and Fienberg 1980). The earlier work by Ehrlich (1973) focused on the relationship between crime rate and the probability of imprisonment and average time served in state prisons. The dataset `uscrime` is from Vandaele (1978). It contains 16 variables and 47 observations, one for 47 different states in the USA. The outcome of interest, `ln_offenses`, accounts for the rate of criminal activity per head of population on the log scale. The rest of the variables measure different socio-economic and punishment-related factors, also on the log scale. Variables `ln_prisonp` and `ln_prisont` correspond to the predictors of interest in the analysis by Ehrlich (1973) in the log scale. (In what follows, when we mention predictors, we will imply their log-transformed versions without stating this explicitly.)

```
. use https://www.stata-press.com/data/r19/uscrime
(1960 crime rates for 47 states in the USA)

. describe

Contains data from https://www.stata-press.com/data/r19/uscrime.dta
 Observations:              47                    1960 crime rates for 47 states
                                                    in the USA
    Variables:              16                    23 Feb 2025 13:08
                                                  (_dta has notes)
```

Variable name	Storage type	Display format	Value label	Variable label
ln_offenses	float	%9.0g		Number of offenses per million population, log scale
ln_malepop	float	%9.0g		Number of males of age 14-24 per 1,000 population, log scale
southern	byte	%8.0g		Indicator variable for southern state
ln_meaneduc	float	%9.0g		Mean years of schooling, log scale
ln_police60	float	%9.0g		Police expenditure in 1960, log scale
ln_police59	float	%9.0g		Police expenditure in 1959, log scale
ln_labor	float	%9.0g		Labor force participation rate, log scale
ln_mtofpop	float	%9.0g		Number of males per 1,000 females, log scale
ln_pop	float	%9.0g		State population size in hundred thousands, log scale
ln_nonwhite	float	%9.0g		Number of nonwhites per 1,000 people, log scale
ln_unemp24	float	%9.0g		Unemployment rate of urban males, age 14-24, log scale
ln_unemp39	float	%9.0g		Unemployment rate of urban males, age 35-39, log scale
ln_wealth	float	%9.0g		Wealth in tens of dollars, log scale
ln_ineq	float	%9.0g		Income inequality, log scale
ln_prisonp	float	%9.0g		Probability of imprisonment, log scale
ln_prisont	float	%9.0g		Average time served in state prisons, log scale

```
Sorted by:
```

Raftery, Madigan, and Hoeting (1997) provide an extensive analysis of this dataset in the context of BMA. We will follow some of their analyses by using bmaregress.

▷ Example 15: BMA regression for the USA crime rate

We first use bmaregress to fit a BMA linear regression of ln_offenses on the remaining 15 variables. With more than 12 predictors, bmaregress uses stochastic MC3 sampling, so we specify the rseed() option for reproducibility. Alternatively, we could still use enumeration to visit all $2^{15} = 32{,}768$ models by specifying the enumeration option, but we stick with the MC3 sampling to mimic the setup of Raftery, Madigan, and Hoeting (1997) more closely. We also specify the uniform prior on the model space to be more consistent with the authors' setup, because, by default, bmaregress assumes a uniform prior on the model size.

```
. bmaregress ln_offenses ln_malepop-ln_prisont, mprior(uniform) rseed(18)

Burn-in ...
Simulation ...
Computing model probabilities ...

Bayesian model averaging                      No. of obs        =       47
Linear regression                             No. of predictors =       15
MC3 sampling                                           Groups =        15
                                                       Always =         0
                                              No. of models     =      803
                                                 For CPMP >= .9 =      245
Priors:                                       Mean model size   =    6.428
  Models: Uniform                             Burn-in           =    2,500
   Cons.: Noninformative                      MCMC sample size  =   10,000
   Coef.: Zellner's g                         Acceptance rate   =   0.2902
       g: Benchmark, g = 225                  Shrinkage, g/(1+g) =  0.9956
  sigma2: Noninformative                      Mean sigma2       =    0.037

Sampling correlation = 0.9380
```

ln_offenses	Mean	Std. dev.	Group	PIP
ln_ineq	1.460748	.3458405	13	.99948
ln_meaneduc	1.855513	.6249968	3	.9692
ln_prisonp	-.1834373	.1198622	14	.81149
ln_malepop	1.091052	.732965	1	.77907
ln_police60	.6916081	.5425088	4	.66525
ln_nonwhite	.0517939	.0582191	9	.51981
ln_unemp39	.1545517	.198343	11	.4623
ln_police59	.3509272	.5211463	5	.36779
ln_pop	-.0142643	.0336552	8	.20569
ln_prisont	-.0427634	.1211644	15	.17169
ln_wealth	.0973078	.2782722	12	.15905
southern	.0192679	.0682639	2	.11977
ln_unemp24	-.0033401	.1010699	10	.088322
ln_mtofpop	.0522426	.4667382	7	.063663
ln_labor	.008854	.140245	6	.045103
Always				
_cons	-21.67062	6.336048	0	1

```
Note: Coefficient posterior means and std. dev. estimated from 803 models.
Note: Default prior is used for parameter g.
```

From the output, the uniform prior is used for the model space, and the benchmark prior of Fernández, Ley, and Steel (2001a) with $g = 15^2 = 225$ for these data is used for the regression coefficients. The shrinkage factor of $225/(225 + 1) = 0.9956$ corresponds to little shrinkage of coefficients toward zero. The MC3 sampling has an acceptance rate of about 30% and fairly high sampling correlation of about 94%. We do not suspect nonconvergence, but it is a good practice to look at the graphical convergence summary as well. We will leave you to verify this by running bmagraph pmp. This means that the visited 803 models are among models with high posterior probabilities, and the drawn MCMC sample provides a good representation of the model posterior distribution.

The top three predictors with PIP above 80% are income inequality, ln_ineq, mean years of schooling, ln_meaneduc, and probability of imprisonment, ln_prisonp. Other important predictors for crime rate are percentage of males, ln_malepop, and police expenditure in 1960, ln_police60. These factors were also selected in Raftery, Madigan, and Hoeting (1997) by using various model-selection criteria.

We use the `bmastats models` command to inspect the top 5 models by PMP.

```
. bmastats models
Computing model probabilities ...
Model summary                                   Number of models:
                                                       Visited = 803
                                                       Reported =    5
```

	Analytical PMP	Frequency PMP	Model size
Rank			
1	.0412	.0339	6
2	.03963	.0454	7
3	.0266	.0293	6
4	.02633	.0126	5
5	.02605	.0142	5

Note: Using analytical PMP for model ranking.

Variable-inclusion summary

	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
ln_malepop	x	x	x	x	x
ln_meaneduc	x	x	x	x	x
ln_police60	x	x		x	x
ln_unemp39	x	x	x	x	
ln_ineq	x	x	x	x	x
ln_prisonp	x	x	x		x
ln_nonwhite		x			
ln_police59			x		

Legend:
  x - estimated

The top model has PMP of 0.04 and includes six predictors: `ln_malepop`, `ln_meaneduc`, `ln_police60`, `ln_unemp39`, `ln_ineq`, and `ln_prisonp`. The posterior mean estimate for probability of imprisonment from the earlier estimation table is negative, -0.18, which indicates its deterrent "effect" on the outcome. (As mentioned in Raftery, Madigan, and Hoeting [1997], the term "effect" in this analysis is used loosely.) In addition to `ln_prisonp`, the model considered in Ehrlich (1973), includes time served in prison, `ln_prisont`, as a deterrent to criminal activity. In view of our BMA analysis, this predictor has a PIP of only 0.17 and is not included in the top five models. In the context of model selection, when we work with a single model, there is uncertainty about the inclusion of this predictor (Raftery, Madigan, and Hoeting 1997).

◁

▷ Example 16: Assessing BMA predictive performance

To assess the predictive performance of our BMA regression, we randomly split the dataset into two equal subsets, fit the model to one of them and test the predictive performance on the other. We adopt the prediction analysis of Raftery, Madigan, and Hoeting (1997) for comparison. We will obtain the 90% predictive CrIs for the test subsample and compute the percentage of the observed values that fall within these intervals.

We use the `splitsample` command to create the `sample` variable, which identifies the training and test subsamples.

```
. splitsample, generate(sample) nsplit(2) rseed(18)
```

We then refit our model using the first subsample. We also save the MCMC model simulation results in a dataset, `bma1model.dta`, to use `bmacoefsample` later.

```
. bmaregress ln_offenses ln_malepop-ln_prisont if sample == 1,
> mprior(uniform) rseed(18) saving(bma1model)

Burn-in ...
Simulation ...
Computing model probabilities ...

Bayesian model averaging                    No. of obs          =      24
Linear regression                           No. of predictors   =      15
MC3 sampling                                         Groups     =      15
                                                     Always     =       0
                                            No. of models        =     845
                                               For CPMP >= .9 =     246
Priors:                                     Mean model size     =   4.254
  Models: Uniform                           Burn-in             =   2,500
   Cons.: Noninformative                    MCMC sample size    =  10,000
   Coef.: Zellner's g                       Acceptance rate     =  0.2532
       g: Benchmark, g = 225                Shrinkage, g/(1+g)  =  0.9956
  sigma2: Noninformative                    Mean sigma2         =   0.043

Sampling correlation = 0.9556
```

ln_offenses	Mean	Std. dev.	Group	PIP
ln_ineq	1.742633	.571148	13	.96936
ln_police60	1.066626	.7762218	4	.7983
ln_meaneduc	.9848326	1.016487	3	.56043
ln_prisonp	−.0533234	.1007384	14	.2904
ln_labor	.531245	1.073081	6	.26204
ln_police59	.1248871	.7554263	5	.2585
ln_pop	−.029143	.0621607	8	.23924
ln_unemp39	.1347758	.3466462	11	.19688
ln_wealth	.1413427	.471408	12	.13636
ln_unemp24	−.0771369	.2646401	10	.12362
ln_prisont	.0209606	.1144098	15	.10844
southern	.0139217	.0754495	2	.092931
ln_malepop	.0731975	.3506361	1	.083313
ln_nonwhite	.0034271	.0257528	9	.068867
ln_mtofpop	.0052071	.6672984	7	.065453
Always				
_cons	−17.13352	8.459582	0	1

Note: Coefficient posterior means and std. dev. estimated from 845 models.
Note: Default prior is used for parameter *g*.

file **bma1model.dta** saved.

The MCMC model simulation file saved by bmaregress contains the information about the visited models and model-specific posterior means and standard deviations of regression coefficients. But to compute CrIs, we also need an MCMC sample from posterior distributions of the model parameters. The bmacoefsample command generates such a sample. We also save the MCMC results from bmacoefsample in bma1coef.dta, as required by bmapredict, which we will use later to generate CrIs.

```
. bmacoefsample, saving(bma1coef) rseed(18)
Simulation (10000): ....5000....10000 done
file bma1coef.dta saved.
```

The MCMC dataset bma1coef contains the draws from the posterior distribution of model parameters, which will be used to generate CrIs.

We now use bmapredict to compute the 90% lower and upper CrIs using the test sample and save them in new variables bma1l and bma1u.

```
. bmapredict bma1l bma1u if sample == 2, cri clevel(90) rseed(18)
note: computing credible intervals using simulation.
Computing predictions ...
```

Now we compute the actual CrI coverage.

```
. generate bma1cov90 = ln_offenses < bma1u & ln_offenses > bma1l if sample == 2
(24 missing values generated)
. summarize bma1cov90
```

Variable	Obs	Mean	Std. dev.	Min	Max
bma1cov90	23	.8695652	.3443502	0	1

We obtain about 87% coverage for the observed outcome in the test sample, which is close to the theoretical 90%. In Raftery, Madigan, and Hoeting (1997), the authors report a predictive coverage of 80% for Occam's window algorithm and a 67% coverage or below for various other model selection procedures.

For comparison, let's also compute predictive CrI coverages based on two regression models: one that includes all predictors, the full model, and one that corresponds to the highest posterior model (HPM), rank 1, model from example 15.

We can still use bmaregress to fit these models, but we now need to specify the relevant predictors to be always included in the model.

```
. bmaregress ln_offenses (ln_malepop-ln_prisont, always) if sample == 1,
> mprior(uniform) saving(bma2model) notable
Enumerating models ...
Computing model probabilities ...
Bayesian model averaging                    No. of obs        =      24
Linear regression                           No. of predictors =      15
Model enumeration                                    Groups =       0
                                                     Always =      15
Priors:                                     No. of models     =       1
  Models: Uniform                             For CPMP >= .9 =       1
   Cons.: Noninformative                     Mean model size   = 15.000
   Coef.: Zellner's g
       g: Benchmark, g = 225                 Shrinkage, g/(1+g) = 0.9956
  sigma2: Noninformative                     Mean sigma2       =   0.010
file bma2model.dta saved.
```

As expected, there is only one visited model in the model space.

We now repeat the earlier computations of CrIs and their coverage for the full model.

```
. bmacoefsample, saving(bma2coef) rseed(18)
Simulation (10000): ....5000....10000 done
file bma2coef.dta saved.
. bmapredict bma2l bma2u if sample==2, cri clevel(90) rseed(18)
note: computing credible intervals using simulation.
Computing predictions ...
. generate bma2cov90 = ln_offenses < bma2u & ln_offenses > bma2l if sample == 2
(24 missing values generated)
. summarize bma2cov90
```

Variable	Obs	Mean	Std. dev.	Min	Max
bma2cov90	23	.3913043	.4990109	0	1

The full model achieves only 39% coverage, which is substantially lower than the BMA coverage.

Next we compute predictions using the HPM model reported by the bmaregress command in example 15. It includes the following six predictors: ln_malepop, ln_meaneduc, ln_police60, ln_unemp39, ln_ineq, and ln_prisonp. We specify these variables in the always group and then compute the predictive CrI coverage as before.

```
. bmaregress ln_offenses
> (ln_malepop ln_meaneduc ln_police60 ln_unemp39 ln_ineq ln_prisonp, always)
> if sample == 1, mprior(uniform) saving(bma3model) notable
Enumerating models ...
Computing model probabilities ...
```

Bayesian model averaging		No. of obs	=	24
Linear regression		No. of predictors	=	6
Model enumeration		Groups	=	0
		Always	=	6
Priors:		No. of models	=	1
Models: Uniform		For CPMP >= .9	=	1
Cons.: Noninformative		Mean model size	=	6.000
Coef.: Zellner's g				
g: Benchmark, g = 36		Shrinkage, g/(1+g)	=	0.9730
sigma2: Noninformative		Mean sigma2	=	0.036

```
file bma3model.dta saved.
. bmacoefsample, saving(bma3coef) rseed(18)
Simulation (10000): ....5000....10000 done
file bma3coef.dta saved.
. bmapredict bma3l bma3u if sample==2, cri clevel(90) rseed(18)
note: computing credible intervals using simulation.
Computing predictions ...
. generate bma3cov90 = ln_offenses < bma3u & ln_offenses > bma3l if sample == 2
(24 missing values generated)
. summarize bma3cov90
```

Variable	Obs	Mean	Std. dev.	Min	Max
bma3cov90	23	.826087	.3875534	0	1

This model produces the predictive CrI coverage of about 83%, but it is still lower than the BMA model with coverage of 87%.

Once we are done with our analysis, we can remove the extra variables and datasets that we created:

```
. drop bma*l bma*u bma*cov90
. erase bma1model.dta
. erase bma2model.dta
. erase bma3model.dta
. erase bma1coef.dta
. erase bma2coef.dta
. erase bma3coef.dta
```

◁

BMA analysis of cross-country economic growth data

In the following series of examples, we apply BMA to an extensively studied problem of economic growth. We follow some of the methodology and use the dataset from Fernández, Ley, and Steel (2001b), which is considered one of the fundamental papers on BMA. Also see, for instance, Eicher, Papageorgiou, and Raftery (2011), who investigated the effect of different BMA priors in a study of economic growth determinants and found that a uniform model prior and a unit-information g-prior performed best among considered priors. And see Ley and Steel (2009) for the effect of prior assumptions in economic growth modeling.

The econgrowth dataset contains information about 72 countries, including their average per capita gross domestic product (GDP) computed over the period 1960 through 1992, size of labor force, life expectancy, etc. It is a subset from an earlier study in Sala-I-Martin (1997). Below, we describe the variables in the dataset.

```
. use https://www.stata-press.com/data/r19/econgrowth
(Economic growth data)

. describe

Contains data from https://www.stata-press.com/data/r19/econgrowth.dta
 Observations:            72                  Economic growth data
    Variables:            43                  8 Mar 2025 10:17
                                              (_dta has notes)
```

Variable name	Storage type	Display format	Value label	Variable label
gdpgrowth	float	%9.0g		* Average annual growth in GDP per capita, 1960–1992
abslat	float	%9.0g		Absolute latitude (degrees)
age	byte	%8.0g		Age
area	int	%8.0g		Surface area (in 1,000s of square kilometers)
blmktpm	float	%9.0g		Premium for black market
brit	byte	%8.0g		British colony dummy
buddha	float	%9.0g		Fraction Buddhist
catholic	float	%9.0g		Fraction Catholic
civllib	float	%9.0g		* Civil liberties index
confucian	float	%9.0g		Fraction Confucian
ecoorg	byte	%8.0g		* Degree of capitalism
english	float	%9.0g		Fraction speaking English
equipinv	float	%9.0g		Share of output invested in physical equipment
ethnol	float	%9.0g		Ethnolinguistic fractionalization
foreign	float	%9.0g		Fraction speaking language other than English

```
     french          byte     %8.0g                    French colony dummy
     gdp60           float    %9.0g                    Log of GDP per capita in 1960
     highenroll      float    %9.0g                    Higher education enrollment
     hindu           float    %9.0g                    Fraction Hindu
     jewish          float    %9.0g                    Fraction Jewish
     labforce        float    %9.0g                    Labor force (in 1,000s)
     latamerica      byte     %8.0g                    Latin America dummy
     lifeexp         float    %9.0g                    Life expectancy
     mining          float    %9.0g                    Fraction of GDP in mining
     muslim          float    %9.0g                    Fraction Muslim
     nequipinv       float    %9.0g                    Share of output invested in items
                                                          other than equipment
     outwaror        byte     %8.0g                    Outward orientation
     polrights       float    %9.0g                 *  Political rights index
     popg            float    %9.0g                    Annual percent change in the
                                                          population
     prexports       float    %9.0g                    Proportion of exports that are
                                                          primary, 1970
     protestants     float    %9.0g                    Fraction Protestants
     prscenroll      float    %9.0g                    Primary school enrollment
                                                          (proportion)
     publedupct      float    %9.0g                    Public education share
     revncoup        float    %9.0g                    Revolutions and coups
     rfexdist        int      %8.0g                    Exchange rate distortions
     ruleoflaw       float    %9.0g                    Index developed by the World
                                                          Justice Project (WJP)
     spanish         byte     %8.0g                    Spanish colony dummy
     stdbmp          float    %9.0g                    Std. dev. of black market premium
     subsahara       byte     %8.0g                    Sub-Saharan dummy
     wardummy        byte     %8.0g                    War dummy
     workpop         float    %9.0g                    Log of employment-population
                                                          ratio
     yrsopen         float    %9.0g                    Fraction of years an economy has
                                                          been open between 1950 and 1990
     y               float    %9.0g                 *  Average annual growth in GDP per
                                                          capita, 1960-1992
                                                      * indicated variables have notes
```

Sorted by:

We are interested in identifying the factors explaining the growth rate by considering a regression model of GDP growth over a host of country characteristics. BMA will be used to account for model uncertainty.

▷ Example 17: BMA linear regression of economic growth

We fit the same BMA regression model as described in Fernández, Ley, and Steel (2001b). It uses a uniform prior on the model space, where each model has the same prior probability and a Zellner's g-prior for the regression coefficients with $g = \max(n, p^2) = 1,681$, where $n = 72$ is the sample size and $p = 41$ is the number of predictors. This benchmark prior, gprior(benchmark), is the default in bmaregress.

With so many possible predictors, the model space is too big to explore by model enumeration, so the MC3 sampling will be used. We increase the MCMC sample size to 200,000 and set a random-number seed for reproducibility. With that many iterations, the command will take a few moments to run, so we display a dot every 5,000 iterations to monitor the progress.

```
. bmaregress gdpgrowth abslat-yrsopen, mprior(uniform) mcmcsize(200000)
> rseed(18) dots(5000)

Burn-in (2500):   done
Simulation (200000): ...................................... done

Computing model probabilities ...
```

Bayesian model averaging	No. of obs	=	72
Linear regression	No. of predictors	=	41
MC3 sampling	Groups	=	41
	Always	=	0
	No. of models	=	22,019
	For CPMP >= .9	=	3,911
Priors:	Mean model size	=	9.593
Models: Uniform	Burn-in	=	2,500
Cons.: Noninformative	MCMC sample size	=	200,000
Coef.: Zellner's g	Acceptance rate	=	0.1724
g: Benchmark, g = 1,681	Shrinkage, g/(1+g)	=	0.9994
sigma2: Noninformative	Mean sigma2	=	.000055

Sampling correlation = 0.9427

gdpgrowth	Mean	Std. dev.	Group	PIP
gdp60	-.0161835	.0029825	16	.99977
confucian	.0563652	.0128534	9	.99861
lifeexp	.0008438	.0003119	22	.95731
equipinv	.1648326	.0619083	12	.95457
subsahara	-.0118906	.0079718	38	.77207
muslim	.0087556	.007237	24	.67147
ruleoflaw	.0082255	.0084833	35	.53652
yrsopen	.0071409	.008048	41	.48922
ecoorg	.0013196	.0014889	10	.48642
protestants	-.0058382	.0070492	30	.46548
nequipinv	.0252486	.0319193	25	.4349
mining	.0169622	.0224533	23	.41827
latamerica	-.0014542	.0036659	21	.16994
prscenroll	.0035024	.0087742	31	.16556
buddha	.0018523	.0050603	6	.14496
blmktpm	-.0010587	.0029964	4	.13458
catholic	-.0003605	.0027147	7	.098655
hindu	-.0018908	.007526	18	.079742
civllib	-.0001846	.0007183	8	.078893
prexports	-.0005476	.002645	29	.054696
polrights	-.0000816	.0004163	27	.050608
rfexdist	-2.71e-06	.0000142	34	.046846
age	-1.79e-06	.0000106	2	.038836
wardummy	-.000138	.0008583	39	.035901
foreign	.0001651	.0010763	14	.035183
english	-.0002166	.0013975	11	.034352
labforce	1.95e-09	1.77e-08	20	.029665
ethnol	.0001246	.0010726	13	.023176
french	.000088	.0007816	15	.021807
spanish	.000089	.0009374	36	.021585
stdbmp	-3.24e-07	2.83e-06	37	.021269
abslat	1.84e-07	.000019	1	.016268
workpop	-.0001042	.0013305	40	.015206
outwaror	-.0000245	.00035	26	.014345
popg	.0021388	.0272493	28	.01341
highenroll	-.0003523	.0045785	17	.01317
brit	-.000021	.0003203	5	.010822
jewish	-.000091	.0016002	19	.010737
publedupct	.000255	.0143465	32	.010524

```
Always
        _cons   .0705111    .0208966                        0          1
```

Note: Coefficient posterior means and std. dev. estimated from 22,019 models.
Note: Default prior is used for parameter *g*.
Note: 2 predictors with PIP less than .01 not shown.

The number of models visited by MC3 is 22,019. Of those, 3,911 contribute to the CPMP of at least 0.9, so we can expect many models with low PMPs. The mean model size is 9.6, so, on average, models tend to include about 10 predictors. The overall acceptance rate is about 17%, which is a reasonable value for MC3 sampling in a high-dimensional space. The sampling correlation is about 0.94, so nonconvergence should not be suspected. Recall that this is the correlation between the analytical posterior model probabilities and their MCMC sampling frequencies. We can use the bmagraph pmp command to plot these frequencies for visual inspection.

. bmagraph pmp

Top 100 models shown out of 22,019 visited.
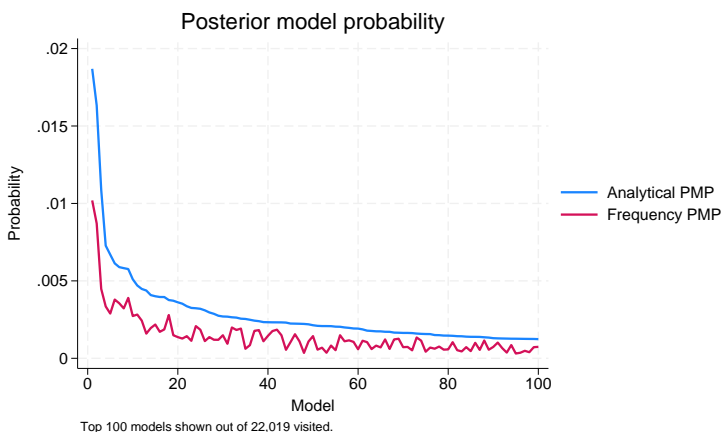
For the first 100 models, the frequency PMPs are below the analytical PMPs because we explored only a small fraction of all possible models, and the PMP distribution has a long and heavy tail. With the increase of the MCMC sample size, the differences will diminish. We do not have a reason to suspect nonconvergence.

bmaregress reports results for 39 out of 41 predictors. Two of the predictors have PIPs less than 0.01 and thus are not reported. If we want, we can specify the allcoef option on replay to see all coefficients.

```
. bmaregress, allcoef noheader
```

gdpgrowth	Mean	Std. dev.	Group	PIP
gdp60	-.0161835	.0029825	16	.99977
confucian	.0563652	.0128534	9	.99861
lifeexp	.0008438	.0003119	22	.95731
equipinv	.1648326	.0619083	12	.95457
subsahara	-.0118906	.0079718	38	.77207
muslim	.0087556	.007237	24	.67147
ruleoflaw	.0082255	.0084833	35	.53652
yrsopen	.0071409	.008048	41	.48922
ecoorg	.0013196	.0014889	10	.48642
protestants	-.0058382	.0070492	30	.46548
nequipinv	.0252486	.0319193	25	.4349
mining	.0169622	.0224533	23	.41827
latamerica	-.0014542	.0036659	21	.16994
prscenroll	.0035024	.0087742	31	.16556
buddha	.0018523	.0050603	6	.14496
blmktpm	-.0010587	.0029964	4	.13458
catholic	-.0003605	.0027147	7	.098655
hindu	-.0018908	.007526	18	.079742
civllib	-.0001846	.0007183	8	.078893
prexports	-.0005476	.002645	29	.054696
polrights	-.0000816	.0004163	27	.050608
rfexdist	-2.71e-06	.0000142	34	.046846
age	-1.79e-06	.0000106	2	.038836
wardummy	-.000138	.0008583	39	.035901
foreign	.0001651	.0010763	14	.035183
english	-.0002166	.0013975	11	.034352
labforce	1.95e-09	1.77e-08	20	.029665
ethnol	.0001246	.0010726	13	.023176
french	.000088	.0007816	15	.021807
spanish	.000089	.0009374	36	.021585
stdbmp	-3.24e-07	2.83e-06	37	.021269
abslat	1.84e-07	.000019	1	.016268
workpop	-.0001042	.0013305	40	.015206
outwaror	-.0000245	.00035	26	.014345
popg	.0021388	.0272493	28	.01341
highenroll	-.0003523	.0045785	17	.01317
brit	-.000021	.0003203	5	.010822
jewish	-.000091	.0016002	19	.010737
publedupct	.000255	.0143465	32	.010524
revncoup	7.98e-06	.0005547	33	.0090622
area	-2.05e-09	5.24e-08	3	.0088675
Always				
_cons	.0705111	.0208966	0	1

Note: Coefficient posterior means and std. dev. estimated from 22,019 models.
Note: Default prior is used for parameter g.

Commonly in BMA, a predictor is considered important if its PIP is greater than 0.5. In this example, we see that several predictors, such as log of GDP in 1960, fraction Confucian, life expectancy, equipment investment, sub-Saharan indicator, fraction Muslim, and rule of law, play an important role in explaining economic growth. On the other hand, predictors with low PIPs, such as surface area with PIP below 0.01, contribute little to explaining the economic growth.

◁

▷ Example 18: Model and variable-inclusion summaries

Let's continue with example 17 and use `bmastats models` to explore the top models visited by `bmaregress`. To limit the number of predictors displayed for each model to only those with PIP above 0.1, we specify the `pipcutoff(0.1)` option.

```
. bmastats models, pipcutoff(0.1)
Computing model probabilities ...
Model summary                              Number of models:
                                                  Visited = 22,019
                                                  Reported =      5
```

	Analytical PMP	Frequency PMP	Model size
Rank			
1	.01869	.01019	10
2	.01632	.008705	9
3	.01088	.00448	8
4	.007274	.00335	7
5	.006697	.00289	7

Note: Using analytical PMP for model ranking.

Variable-inclusion summary

	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
confucian	x	x	x	x	x
ecoorg	x	x			
equipinv	x	x	x	x	x
gdp60	x	x	x	x	x
lifeexp	x	x	x	x	x
muslim	x	x	x	x	x
nequipinv	x	x			
protestants	x			x	
ruleoflaw	x	x			
subsahara	x	x	x		x
mining			x		
yrsopen			x	x	x

```
Legend:
  x - estimated
```

By default, the command displays the top five models ranked by PMP. It reports both analytical and frequency PMPs, which are similar because the model converged. It also reports the model sizes. The predictors included in each reported model are displayed in a separate table. The `pipcutoff(0.1)` option did not have an effect in our example, because all predictors in these top five models have a PIP greater than 0.1, which can be verified by running the command without this option.

The top model has a rather low PIP of 0.019 and includes 10 predictors, among which there are all 7 important predictors. We can list the PIPs for the predictors of the top model by using bmastats pip:

```
. bmastats pip confucian ecoorg equipinv gdp60 lifeexp muslim nequipinv
> protestants ruleoflaw subsahara

Posterior inclusion probability (PIP)

No. of obs          =      72
No. of predictors   =      41
           Groups   =      41
           Always   =       0
         Reported   =      10
No. of models       = 22,019
Mean model size     =   9.593

                            PIP        Group

        gdp60           .99977           16
     confucian          .99861            9
       lifeexp          .95731           22
      equipinv          .95457           12
      subsahara         .77207           38
        muslim          .67147           24
     ruleoflaw          .53652           35
        ecoorg          .48642           10
    protestants         .46548           30
      nequipinv          .4349           25

    Always
         _cons               1            0

Note: Using analytical PMPs.
```

The other three predictors have PIPs above 0.4.

The rank 2 model with a PMP of 0.016 includes the same predictors as the top model, except for `protestants`. The remaining models have relatively lower PMPs. The presence of so many models with similar low probabilities means that there are many plausible models that can be considered for these data. Thus, if we were to choose just one, it would have been difficult to select the "best" one.

We may be also interested in some specific regression models from the BMA model sample. For example, we may be interested in models that include any of the seven important predictors, which we can specify in the include() option. And, in addition to the HPM, we may want to explore the median probability model (MPM). MPM is the model that includes only predictors with a PIP greater or equal to 0.5 (Fletcher 2018). The include() option may select many models, so we consider only those with PMP above 0.0025.

```
. bmastats models, hpm mpm pmpcutoff(0.0025)
> include(gdp60 confucian lifeexp equipinv subsahar muslim ruleoflaw)

Computing model probabilities ...
```

Model summary

Number of models:

Visited = 22,019
Reported =   13

	Analytical PMP	Frequency PMP	Model size
Rank			
(HPM) 1	.01869	.01019	10
2	.01632	.008705	9
7	.005889	.003555	8
9	.005764	.0039	10
15	.004007	.00218	11
16	.003959	.00171	9
17	.003956	.001865	11
18	.003762	.002795	11
21	.003527	.001275	8
22	.003362	.00143	11
26	.003097	.00111	10
30	.002701	.00149	11
(MPM) 33	.002628	.00183	7

Notes: Using analytical PMP for model ranking.
       3,932 models with PMP less than .0025 not shown.

Note: Use option **vartable** to display variable-inclusion table for more than
      12 models.

There are 13 models that include the important predictors and have a PMP above 0.0025, and there are 3,932 more models with a PMP below that. The MPM includes our seven important predictors. Its PMP is only 0.0026. With many predictors, it is not unreasonable to see so many models with low PMP in the absence of the strong information in the data about the model.

The bmastats msize command summarizes the sizes of the explored models. The model size is the number of predictors included in the model. In addition to the posterior mean size, which is also reported in the header of bmaregress, the command reports the posterior median size and the minimum and maximum model sizes. For comparison, it also reports the summaries for the prior model-size distribution.

```
. bmastats msize
```

Model-size summary

Number of models = 22,019
Model size:
  Minimum = 4
  Maximum = 22

	Mean	Median
Prior		
Analytical	11.0877	11
Posterior		
Analytical	9.5933	10
Frequency	10.4171	10

The smallest model includes only 4 predictors, and the largest model includes 22 predictors. The analytical posterior median size is 10 compared with the mean of 9.59. The posterior model-size summaries are similar to the prior ones. And frequency model-size summaries computed from the MCMC sample are similar to the analytical ones, as would be expected for the converged model.

We can plot the entire distributions of model sizes by using `bmagraph msize`.

```
. bmagraph msize
```
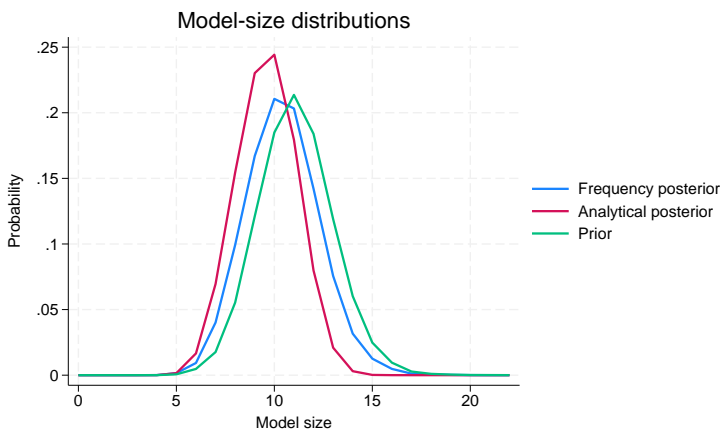

The model-size distributions are fairly similar with the posterior one shifted to the left, favoring slightly smaller models than what was assumed a priori.

◁

▷ Example 19: Coefficient summaries

Let's use `bmagraph coefdensity` to look at the distributions of regression coefficients for gdp60 and ruleoflaw.

```
. bmagraph coefdensity {gdp60}
```
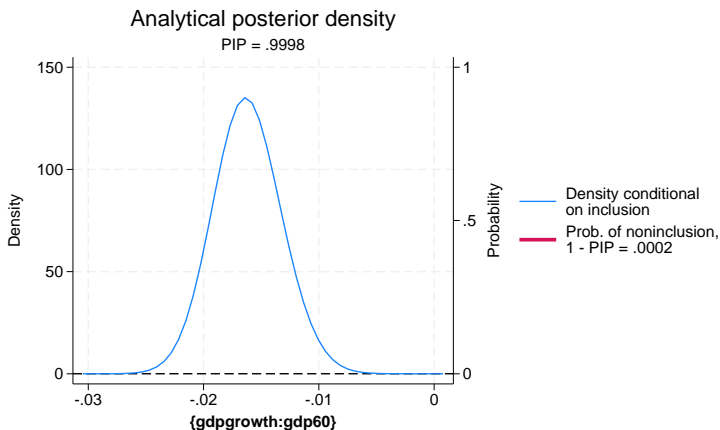
For coefficient {gdp60} (or using its full name {gdpgrowth:gdp60}), the probability of noninclusion is very low, 0.0002, so the red line that represents it is not even visible on the graph. Thus the posterior density of {gdp60} is essentially a continuous normal-like density, centered at about -0.015 with most of its mass between roughly -0.025 and -0.005, away from 0.

We now look at the posterior density for the {ruleoflaw} coefficient.

```
. bmagraph coefdensity {ruleoflaw}
```
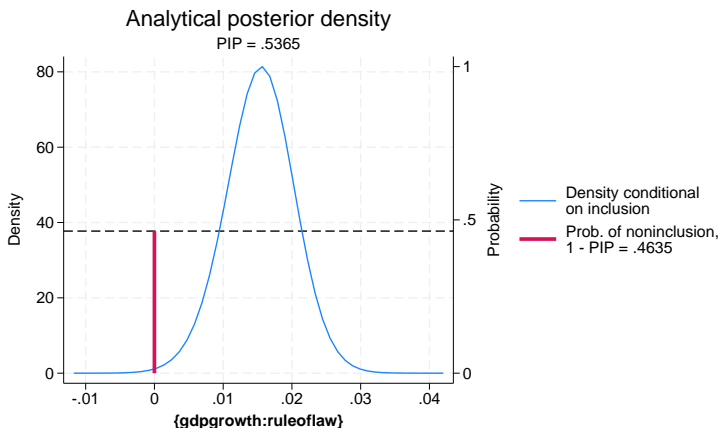

The probability of noninclusion for {ruleoflaw} is about 0.46, so we see the red vertical line meets the horizontal reference line at roughly 0.46. Conditional on the inclusion, the continuous density is centered around 0.015 with most of its mass between 0 and 0.03.

The posterior means and standard deviations reported by bmaregress are computed with respect to the above mixtures of distributions.

If we want to compute CrIs, we need to use bmacoefsample first to obtain samples of regression coefficients from their posterior distributions, shown above for {gdp60} and {ruleoflaw}. To run bmacoefsample, we need to save our BMA sampling results first, which we can do on replay:

```
. bmaregress, saving(bmadata_econgrowth)
note: file bmadata_econgrowth.dta saved.
. bmacoefsample, mcmcsize(10000) rseed(18)
Simulation (10000): ....5000....10000 done
```

bmacoefsample uses the same MCMC size as bmaregress to generate the sample, unless the mcmcsize() option is specified. Here we do not need 200,000 samples of coefficients, so we specify a smaller number of 10,000. The size of this sample affects the accuracy of the MCMC-based computations of the posterior summaries of regression coefficients.

Once we have an MCMC sample of coefficients, many Bayesian postestimation commands are available. For example, we can use bayesstats summary to compute the 95% HPD CrIs for the coefficients of, say, equipinv and ruleoflaw.

```
. bayesstats summary {gdpgrowth: equipinv ruleoflaw}, hpd
Posterior summary statistics                        MCMC sample size =     10,000
```

					HPD	
gdpgrowth	Mean	Std. dev.	MCSE	Median	[95% cred. interval]	
equipinv	.1653323	.0627102	.000627	.1684731	0	.2607861
ruleoflaw	.0082453	.0084968	.000085	.0078141	0	.0219461

The first two columns report the posterior means and standard deviations based on the MCMC simulation, which approximate the analytical ones reported by bmaregress. The MCMC-based estimates are similar to the analytical ones. The MCSE column reports the Monte Carlo standard errors, which describes the precision of the posterior-mean estimates and depends on the size of the MCMC sample specified in mcmcsize() with bmacoefsample. The last two columns report the 95% HPD CrIs. For instance, the reported 95% CrI for {ruleoflaw} is consistent with its posterior density shown above.

We can also use bayesstats summary to compute posterior summaries for expressions of coefficients. Suppose we are interested in estimating the posterior probability that both coefficients for equipinv and ruleoflaw are positive. We can do this as follows:

```
. bayesstats summary
((({gdpgrowth:ruleoflaw} > 0) & ({gdpgrowth:equipinv} > 0)))
Posterior summary statistics                        MCMC sample size =     10,000
      expr1 : ({gdpgrowth:ruleoflaw} > 0) & ({gdpgrowth:equipinv} > 0)
```

					Equal-tailed	
	Mean	Std. dev.	MCSE	Median	[95% cred. interval]	
expr1	.496	.500009	.004948	0	0	1

The probability of both of these coefficients being positive is about 50%.

◁

▷ Example 20: Jointness measures

In the context of BMA, we can explore additional characteristics about the included predictors such as their tendencies of being included in a model together, separately, or independently. This is described by the so-called jointness measures.

Let's compute jointness measures for some predictors. First, we compute these measures for a pair of predictors with high PIPs: gdp60 and lifeexp. We use the `bmastats jointness` command for this.

```
. bmastats jointness gdp60 lifeexp
Computing model probabilities ...
Variables: gdp60 lifeexp
```

	Jointness
Doppelhofer–Weeks	6.944544
Ley–Steel type 1	.9575255
Ley–Steel type 2	22.54354
Yule's Q	.9980741

```
Notes: Using analytical PMPs. See
       thresholds.
```

All reported jointness measures suggest that the two predictors are complements, which means that they each contain additional information that helps explain the outcome. For instance, the Doppelhofer–Weeks measure is $6.94 > 2$, which means "strong jointness", and the Ley–Steel type 2 measure is $22.54 > 10$, which also indicates "strong jointness"; see [BMA] **bmastats jointness** for details.

Second, let's compute jointness measures for a pair of predictors with lower PIPs, for example, polrights and civllib.

```
. bmastats jointness polrights civllib
Computing model probabilities ...
Variables: polrights civllib
```

	Jointness
Doppelhofer–Weeks	-2.669263
Ley–Steel type 1	.0024346
Ley–Steel type 2	.0024405
Yule's Q	-.8703767

```
Notes: Using analytical PMPs. See
       thresholds.
```

All reported jointness measures suggest that the two predictors are substitutes, which means that they do not bring any additional information to help explain the outcome when included together. For instance, the Doppelhofer–Weeks measure is $-2.67 < -2$, which means "strong disjointness", and the Ley–Steel type 2 measure is $0.0024 < 0.01$, which indicates "decisive disjointness".

◁

▷ Example 21: BMA regression of economic growth using random parameter g

By default, `bmaregress` uses a fixed value for the g parameter of a Zellner's g-prior, which limits the class of explored regression models. There is no one optimal value for g. A more general class of models can be considered by allowing g to vary between models according to a prespecified hyperprior distribution.

The `bmaregress` command supports a number of hyperpriors for g: betashrink, betabench, hyperg, hypergn, zsiow, and robust. Below, we use the betabench prior for illustration. This hyperprior is controlled by a parameter a, $a > 0$, and it is equivalent to specifying the Beta$\{a \times \max(n, p^2), a\}$ prior distribution on the shrinkage $\delta = g/(g+1)$. We choose $a = 10$, which corresponds to the

Beta(16,810, 10) prior. The prior mean for the shrinkage is thus very close to 1, 0.9994 to be exact. We first run the model without showing the output table. We do not want to focus on the results before we check the MCMC convergence and make sure that the sample we generated is representative of the model posterior distribution.

```
. bmaregress gdpgrowth abslat-yrsopen, mprior(uniform) gprior(betabench 10)
> rseed(18) notable

Burn-in ...
Simulation ...
Computing model probabilities ...

Bayesian model averaging                         No. of obs      =        72
Linear regression                                No. of predictors =      41
MC3 and adaptive MH sampling                                Groups =      41
                                                            Always =       0
                                                 No. of models   =     1,685
                                                   For CPMP >= .9 =     1,070
Priors:                                          Mean model size =    10.870
  Models: Uniform                                Burn-in         =     2,500
   Cons.: Noninformative                         MCMC sample size =    10,000
   Coef.: Zellner's g                            Acceptance rate =    0.5655
       g: Benchmark-beta-shrinkage(10)
  sigma2: Noninformative                         Mean sigma2     =   .000054

Sampling correlation = 0.1080
```

Here the MC3 algorithm did not converge. It visited only 1,685 models, a very small portion of the entire model space, and, not surprisingly, the sampling correlation is relatively low, about 11%, which suggests inadequate exploration of the model space.

But if we look at the diagnostics plot of the g parameter, it shows sufficient mixing with diminishing autocorrelation after 10 lags and does not raise any convergence issues.

```
. bayesgraph diagnostics {g}
```
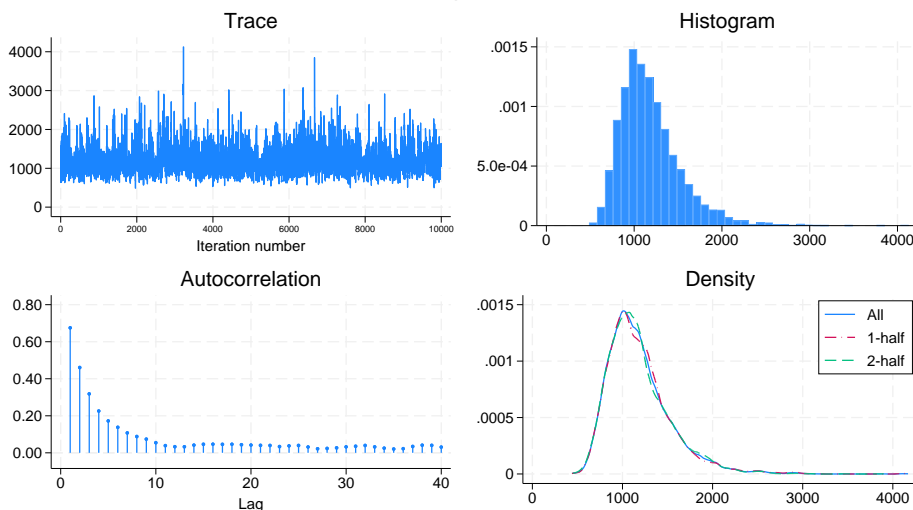
So the convergence for g does not imply convergence over the model space. To improve the latter, we specify a larger initial value for g, 1,000, which is closer to the prior mean of g, and increase the MCMC sample size to 40,000.

```
. bmaregress gdpgrowth abslat-yrsopen, mprior(uniform) gprior(betabench 10)
> ginit(1000) mcmcsize(40000) rseed(18)

Burn-in ...
Simulation ...
Computing model probabilities ...

Bayesian model averaging                    No. of obs        =        72
Linear regression                           No. of predictors =        41
MC3 and adaptive MH sampling                          Groups  =        41
                                                      Always  =         0
                                            No. of models     =     5,985
                                              For CPMP >= .9 =     3,683
Priors:                                     Mean model size   =    10.974
  Models: Uniform                           Burn-in           =     2,500
   Cons.: Noninformative                    MCMC sample size  =    40,000
   Coef.: Zellner's g                       Acceptance rate   =    0.5590
       g: Benchmark-beta-shrinkage(10)
  sigma2: Noninformative                    Mean sigma2       =   .000053

Sampling correlation = 0.3115
```

gdpgrowth	Mean	Std. dev.	Group	PIP
gdp60	-.0162014	.0031171	16	1
confucian	.0571647	.0148002	9	.98813
lifeexp	.0008541	.0003227	22	.95098
equipinv	.1569886	.0662325	12	.93085
muslim	.0098991	.0079137	24	.69912
subsahara	-.0105384	.0087295	38	.68073
yrsopen	.0072551	.007865	41	.51823
ruleoflaw	.0072071	.0081836	35	.4953
protestants	-.0060056	.0070656	30	.48492
mining	.0191716	.0228082	23	.47828
ecoorg	.0012255	.0014384	10	.47625
nequipinv	.0271594	.0324673	25	.46795
prscenroll	.0047258	.0098377	31	.24322
buddha	.0030845	.0063681	6	.23272
latamerica	-.0016883	.0041869	21	.21848
blmktpm	-.0014396	.0033853	4	.18812
civllib	-.0004035	.0010327	8	.17472
catholic	-.0000176	.0032274	7	.14413
polrights	-.0001868	.0006425	27	.12987
hindu	-.0033699	.0118713	18	.11532
prexports	-.0010904	.0037068	29	.1103
age	-4.86e-06	.0000174	2	.10203
english	-.000586	.0022521	11	.099375
wardummy	-.0003553	.0013681	39	.099225
rfexdist	-4.92e-06	.0000187	34	.09735
foreign	.0004337	.0017529	14	.09035
labforce	7.55e-09	3.71e-08	20	.07665
abslat	-5.66e-07	.0000433	1	.06855
ethnol	.0003364	.0017694	13	.06445
spanish	.0002713	.0016564	36	.060425
outwaror	-.0001108	.0007231	26	.05725
workpop	-.0002883	.0023882	40	.052925
popg	.0066005	.0543594	28	.052025
stdbmp	-5.66e-07	3.70e-06	37	.049025
brit	-.0000933	.0007186	5	.046375

area	-9.87e-09	1.34e-07		3	.04465
jewish	-.0003108	.0031789		19	.0432
french	.0001371	.0009942		15	.04185
highenroll	-.0008954	.0075328		17	.03785
publedupct	.0009074	.0244793		32	.03205
revncoup	-8.58e-06	.0009061		33	.03055

Always					
_cons	.0709041	.0237199		0	1

Note: Coefficient posterior means and std. dev. estimated from 5,985 models.

	Mean	Std. dev.	MCSE	Median	Equal-tailed [95% cred. interval]	
g	1184.255	343.643	5.20567	1128.086	700.552	2033.428
Shrinkage	.9990924	.000239	3.5e-06	.9991143	.9985746	.9995085

The sampling correlation improves; it is 31% now compared with 11% earlier. In absolute terms, this may still seem low but, given the size of the model space, 2^{41}, is probably acceptable. Achieving very high sampling correlation, say, 90%, for this model may take a long time. The diagnostic plots for g are better too.

. bayesgraph diagnostics {g}
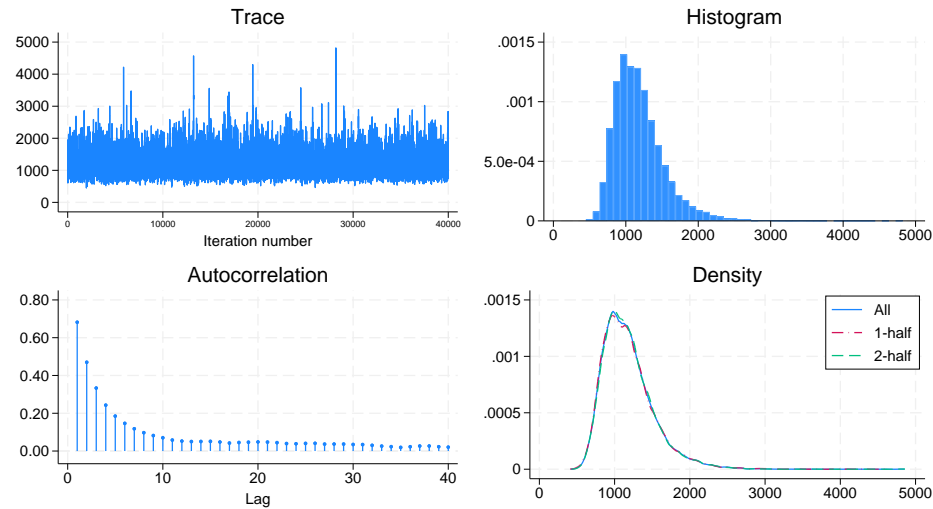
We obtain results similar to those for a fixed g in example 17. The estimated shrinkage posterior mean of 0.999 matches closely the fixed value of 0.994. Thus, the chosen prior for g does not seem to have much of an effect on the results.

Stored results

bmaregress stores the following in e():

Scalars
e(N)	number of observations
e(p)	number of predictors
e(p_groups)	number of groups of predictors
e(p_always)	number of always included predictors
e(k_models)	number of visited models
e(k_models_cpmp)	number of highest PMP models accounting for at least 90% of CPMP
e(k_models_all)	number of all available models
e(msize_mean)	posterior mean model size as displayed in the header
e(msize_mean_a)	analytical posterior mean model size; not available with random g
e(msize_mean_f)	frequency posterior mean model size; not available with model enumeration
e(msize_mean_prior)	prior mean model size
e(mcmcsize)	MCMC size with sampling
e(burnin)	number of burn-in iterations with sampling
e(thinning)	thinning interval with sampling
e(arate)	acceptance rate with sampling
e(sampcorr)	correlation between frequency and analytical PMP; not available with model enumeration
e(sigma2)	posterior mean of the error variance
e(g)	value of g when g is a fixed constant
e(g_init)	initial g-value with sampling
e(shrinkage)	shrinkage with fixed g
e(groupfv)	1 if groupfv is specified; 0 otherwise
e(clevel)	credible interval level with random g
e(hpd)	1 if hpd is specified; 0 otherwise (random g)

Macros
e(cmd)	bmaregress
e(cmdline)	command as typed
e(title)	first title appearing in header
e(title2)	second title appearing in header
e(depvar)	name of dependent variable
e(wtype)	weight type
e(wexp)	weight expression
e(indepvars)	names of predictors included in the models
e(alwaysvars)	names of predictors that are always included
e(omitvars)	names of predictors marked as omitted
e(sampling)	sampling algorithm
e(sampling_lab)	label for sampling algorithm
e(heredity)	heredity rule; only available in the presence of interactions
e(mprior)	model prior option
e(mprior_dist)	name of model prior distribution
e(mprior_lab)	label for model prior distribution
e(gprior)	g-prior option
e(gprior_dist)	name of g-prior distribution
e(gprior_lab)	label for g-prior distribution
e(minitial)	model prior initial option
e(ginitial)	g-prior initial option
e(modelfilename)	name of the file with simulation results
e(rngstate)	random-number state at the time of simulation

Matrices
e(b_bma)	posterior means
e(V_bma)	posterior variance–covariance matrix
e(b_bma_c)	posterior means in estimation (recentered) metric
e(V_bma_c)	posterior variance–covariance matrix in estimation (recentered) metric
e(pip)	probabilities of inclusion

e(group)	group indices for predictors
e(mprior_params)	parameters of model prior distribution (if any)
e(gprior_params)	parameters of g-prior (if any)
e(modelinit)	initial model state binary vector with sampling

Functions
e(sample)	marks estimation sample

Methods and formulas

Methods and formulas are presented under the following headings:

Model assumptions and generic formulas

Consider a regression model with p predictors. Let $\mathcal{M}_F = \{M_1, M_2, \ldots, M_{2^p}\}$ denote the full space of models formed by considering all 2^p possible subsets of p variables, and let $J_F = \{1, 2, \ldots, 2^p\}$ denote the full set of the corresponding indices. Let model $M_j \in \mathcal{M}_F$ include a distinct subset of the p variables for each $j \in J_F$, and let p_j be the number of variables included in M_j; that is, the model size $|M_j| = p_j$.

Consider a sample of observations y_i's and x_{ik}'s on, respectively, outcome Y and p predictors X_1, $X_2, \ldots, X_p$, where $i = 1, 2, \ldots, n$ and $k = 1, 2, \ldots, p$. For each $j \in J_F$, consider a linear regression model M_j using a subset of p_j variables,

$$y_i = \alpha + \mathbf{x}_{i,j}\boldsymbol{\beta}_j + \epsilon_{i,j}$$

where α is an unknown intercept, $\boldsymbol{\beta}_j$ is a $p_j \times 1$ vector of unknown (model-specific) regression coefficients, $\mathbf{x}_{i,j}$ is a $1 \times p_j$ vector of observed values on the variables included in the model, and error terms $\epsilon_{i,j}$'s $\sim$ i.i.d. $N(0, \sigma^2)$.

We can write the above using a matrix notation,

$$\mathbf{y} = \alpha\mathbf{1}_n + \mathbf{X}_j\boldsymbol{\beta}_j + \boldsymbol{\epsilon}_j$$

where $\mathbf{y} = (y_1, y_2, \ldots, y_n)'$ is an $n \times 1$ vector of outcome values, $\mathbf{1}_n$ is a $n \times 1$ vector of ones, $\mathbf{X}_j$ is an $n \times p_j$ design matrix, and $\boldsymbol{\epsilon}_j = (\epsilon_{1,j}, \epsilon_{2,j}, \ldots, \epsilon_{n,j})'$ is an $n \times 1$ vector of model-specific error terms.

In a Bayesian linear regression, model parameters α, $\boldsymbol{\beta}_j$, and σ^2 (or, equivalently, σ) are assumed to have prior distributions, which are conditional on model M_j. In a BMA framework, model M_j or, more precisely, its index is an unknown itself and thus assigned a prior distribution just like any other model parameter. A model prior is a discrete prior, $\{P(M_j)\}_{j \in J_F}$, specified over model space $\mathcal{M}_F$.

The priors for a BMA linear regression are

$$
\begin{aligned}
M_j &\sim P(M_j) \\
\boldsymbol{\beta}_j | \alpha, \sigma, M_j &\sim N_{p_j}\{\mathbf{0}, \sigma^2 g(\mathbf{X}_j{}'\mathbf{X}_j)^{-1}\} \\
\alpha | \sigma, M_j &\propto 1 \\
\sigma | M_j &\propto \sigma^{-1}
\end{aligned}
$$

where $N_{p_j}(\cdot, \cdot)$ denotes a p_j-dimensional multivariate normal distribution and the choices for $P(M_j)$ are described in *Priors on the model space*. The intercept α and the error standard deviation σ are assumed to have noninformative priors. The regression coefficients $\boldsymbol{\beta}_j$ are assumed to follow a Zellner's g-prior (1986), where $g > 0$ controls the shrinkage of coefficients toward 0, and the excluded coefficients are assumed to have a prior point mass at 0; that is, they are assumed to be exact zeros. In the above and throughout, we also implicitly condition on $\mathbf{X}_j$.

The parameter g in Zellner's g-prior can be viewed as fixed or random. With a random g, a joint prior $P(g, M_j) = P(g|M_j)P(M_j)$ is considered, where $P(g|M_j)$ is a hyperprior (or the so-called g-prior) assumed for g. The choices for a g-prior are discussed in *Priors for parameter g*.

With a random g, the priors for a BMA linear regression are

$$
\begin{aligned}
M_j &\sim P(M_j) \\
g &\sim P(g|M_j) \\
\boldsymbol{\beta}_j | \alpha, \sigma, g, M_j &\sim N_{p_j}\{\mathbf{0}, \sigma^2 g(\mathbf{X}_j{}'\mathbf{X}_j)^{-1}\} \\
\alpha | \sigma, g, M_j &\propto 1 \\
\sigma | g, M_j &\propto \sigma^{-1}
\end{aligned}
$$

In Bayesian analysis, the inference about model parameters is based on their posterior distributions, and the prediction of new data is based on the posterior predictive density. So the estimation of these distributions is central to Bayesian estimation. In BMA, we are also interested in estimating PMPs and PIPs. And, because we consider multiple models, we need to distinguish between the posterior distributions conditional on a model and those over all models. In what follows, we give general definitions for these distributions and probabilities and provide specific formulas in later sections.

The PMP for model M_j is

$$
P(M_j|\mathbf{y}) = \frac{f(\mathbf{y}|M_j)P(M_j)}{\sum_{l \in J_F} f(\mathbf{y}|M_l)P(M_l)}
$$

where $f(\mathbf{y}|M_j)$ is the marginal likelihood of M_j.

The PIP for variable k is

$$
\text{PIP}(X_k) = \sum_{j \in J_F} I(X_k \in M_j)P(M_j|\mathbf{y})
$$

where $I(\cdot)$ is the indicator function.

The posterior distribution of $\boldsymbol{\beta}$ over all models is

$$g(\boldsymbol{\beta}|\mathbf{y}) = \sum_{j \in J_F} g(\boldsymbol{\beta}|\mathbf{y}, M_j) P(M_j|\mathbf{y})$$

where $g(\boldsymbol{\beta}|\mathbf{y}, M_j)$ is the posterior distribution of $\boldsymbol{\beta}$ for a Bayesian linear regression model M_j. Posterior distributions of α and σ^2 can be defined in the same fashion.

Similarly, for a new observation $(y^\star, \mathbf{x}^\star)$, the posterior predictive density over all models is

$$f(y^\star|\mathbf{y}, \mathbf{X}, \mathbf{x}^\star) = \sum_{j \in J_F} f(y^\star|\mathbf{y}, \mathbf{X}, \mathbf{x}_j^\star, M_j) P(M_j|\mathbf{y})$$

where $f(y^\star|\mathbf{y}, \mathbf{X}, \mathbf{x}_j^\star, M_j)$ is the posterior predictive density for model M_j.

Priors on the model space

The BMA framework incorporates model uncertainty by specifying a prior on the model space. Each model in the model space can be uniquely represented by a p-dimensional binary vector $\boldsymbol{\gamma} = \{\gamma_k : k = 1, 2, \ldots, p\}$, in which element γ_k equals 1 if and only if the kth predictor is included in the model. Therefore, specifying a prior on the model space is equivalent to specifying a prior on the space of p-dimensional binary vectors. The model size, denoted by $\omega = |M|$, refers to the number of predictors included in the model. That is,

$$\omega = \sum_{k=1}^{p} \gamma_k$$

bmaregress supports the following model priors in the mprior() option: betabinomial, the default, meaning betabinomial 1 1; betabinomial # #; betabinomial #; uniform; binomial #; binomial (where # $=$ 0.5); and binomial ... (with predictor-specific IPs). We define these priors below.

Beta-binomial prior, option mprior(betabinomial ...). This is the default prior, with shape parameters of 1. Consider a binomial prior defined below, where all predictors have a common probability of inclusion p_{inc}. Then specify a hierarchical prior for the parameter p_{inc} as a beta distribution. Stata provides two options to specify the hyperparameters for this beta prior.

A user can specify mprior(betabinomial $\#_1$ $\#_2$) with two hyperparameters $a = \#_1$ and $b = \#_2$, which are the shape parameters of the beta distribution. In this case, the prior model probability is

$$p(\boldsymbol{\gamma}) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \frac{\Gamma(a+\omega)\Gamma(b+p-\omega)}{\Gamma(a+b+p)}$$

where $\Gamma(\cdot)$ is a gamma function.

Under this prior, the expected model size is

$$E(\omega) = \frac{a}{a+b} p$$

A user can specify `mprior(betabinomial #)` with the prior expected model size $w = \#$ instead, where $0 < w < p$. The distribution for p_{inc} is then $\text{Beta}\{1, (p - w)/w\}$. In this case, $E(\omega) = w$.

This alternative parameterization may be more convenient because it might be easier to think of a prior value for the expected model size than for the shape parameters of the beta distribution.

Uniform prior, option `mprior(uniform)`. It assigns an equal probability of $1/2^p$ for each model M_j in the model space.

Binomial prior, option `mprior(binomial ...)`. Let $p_{\text{inc},k}$ be the prior probability of inclusion for the kth predictor. Assume that the inclusion of each predictor is independent; then the binomial prior specifies

$$p(\boldsymbol{\gamma}) = \prod_{k=1}^{p} p_{\text{inc},k}^{\gamma_k} (1 - p_{\text{inc},k})^{1-\gamma_k}$$

If all predictors have a common probability of inclusion p_{inc}, the binomial prior becomes

$$p(\boldsymbol{\gamma}) = p_{\text{inc}}^{\omega} (1 - p_{\text{inc}})^{p-\omega}$$

A detailed discussion of different choices for a prior on the model space can be found in Ley and Steel (2009).

Priors for parameter g

Parameter g in a Zellner's g-prior can be fixed or random with a given hyperprior. The fixed g cases supported by `bmaregress` in the `gprior()` option are `bench` (the default), `uip`, `ric`, `sqrtn`, `fixed #`, and `ebl`. We define these cases below.

Fixed g priors

Benchmark prior, $g = \max(n, p^2)$, option `gprior(bench)`. This prior is the default. It was suggested by Fernández, Ley, and Steel (2001a) and is a combination of the unit-information and risk-inflation-criterion priors, defined below. The authors found it to perform well in a variety of cases with respect to a model's predictive performance.

Unit-information prior, $g = n$, option `gprior(uip)`. Introduced in Kass and Raftery (1995), the unit-information prior specifies a prior with the variance proportional to the sample size. For this choice of g, the log Bayes-factors behave asymptotically like the BIC. In this case, the BIC, given by the negative log likelihood plus a penalty proportional to $\log(n)$, corresponds to the negative log posterior.

Risk inflation criterion, $g = p^2$, option `gprior(ric)`. Proposed by Foster and George (1994), the risk inflation criterion is based on a minimax criterion for variable selection. Specifically, the criterion proposed by the authors is the negative log likelihood plus a penalty proportional to $2\log(p)$, and this choice corresponds to $g = p^2$.

Square-root n prior, $g = \sqrt{n}$, option `gprior(sqrtn)`. One of the fixed g priors suggested by Porwal and Raftery (2022b).

Empirical Bayes local, option `gprior(ebl)`. The local empirical Bayes prior (Liang et al. 2008) uses a different fixed g_j for each model M_j. It is the maximum marginal-likelihood estimate constrained to be nonnegative. For model M_j with $p_j > 0$, it is defined as

$$\hat{g}_j^{\text{EBL}} = \max(F_j - 1, 0)$$

where F_j is the standard F statistic for model M_j defined as

$$F_j = \frac{(n - 1 - p_j)R_{j,\text{ols}}^2}{p_j(1 - R_{j,\text{ols}}^2)}$$

$R_{j,\text{ols}}^2$ in the above is the R^2 from the ordinary least-squares (OLS) linear regression corresponding to model M_j.

$\hat{g}_j^{\text{EBL}}$ maximizes the marginal likelihood (2). For the null model, with $p_j = 0$, we set $\hat{g}_j^{\text{EBL}} = 1$, by convention.

Random g priors

The supported random g priors in the gprior() option are betashrink $\#_1$ $\#_2$, betabench #, hyperg #, hypergn #, zsiow, and robust. These are commonly referred to as g-priors.

Beta-shrinkage prior, option gprior(betashrink $\#_1$ $\#_2$). This is a general beta-shrinkage prior that assumes a beta prior with shape parameters $a = \#_1$ and $b = \#_2$ for the shrinkage $\delta = g/(g + 1)$:

$$\delta \sim \text{Beta}(a, b)$$

This corresponds to the following prior on g:

$$p(g) = \frac{\Gamma(a + b)}{\Gamma(a)\Gamma(b)}g^{(a-1)}(1 + g)^{-(a+b)}$$

Benchmark beta-shrinkage priors, option gprior(betabench #). The benchmark beta-shrinkage prior is a special case of the beta-shrinkage prior as suggested by Ley and Steel (2012). This prior considers the following beta prior on the shrinkage $\delta = g/(g + 1)$ with $b = \#$:

$$\delta \sim \text{Beta}\{b \times \max(n, p^2), b\}, \quad b > 0$$

Hyper-g prior, option gprior(hyperg #). This prior is suggested by Liang et al. (2008). It is a special case of a beta-shrinkage prior with the following beta prior on shrinkage δ with $c = \#$:

$$\delta \sim \text{Beta}\left(1, \frac{c}{2} - 1\right), \quad 2 < c \leq 4$$

Hyper-g/n prior, option gprior(hypergn #). This prior is suggested by Liang et al. (2008). It considers the following prior on g with $c = \#$:

$$p(g) = \frac{c - 2}{2n}\left(1 + \frac{g}{n}\right)^{-\frac{c}{2}}, \quad 2 < c \leq 4$$

This is equivalent to specifying the prior $\text{Beta}(1, c/2 - 1)$ on $g/g + n$.

Zellner–Siow prior, option gprior(zsiow). This prior was introduced by Zellner and Siow (1980):

$$g \sim \text{Inverse-gamma}\left(\frac{1}{2}, \frac{n}{2}\right)$$

Robust prior, option gprior(robust). The robust prior is a special case in a class of priors introduced in Bayarri et al. (2012). It has the following analytical form:

$$p(g) \sim (1+g)^{-3/2}, \quad g > \frac{n+1}{p_j+1} - 1$$

The prior is model specific because it depends on the model size p_j. This particular form of the robust prior has some desired theoretical properties such as model-selection consistency, which states that the posterior probability of a model that generated the data should go to one as sample size goes to infinity (Bayarri et al. 2012, sec. 3.4).

The Zellner–Siow, benchmark beta-shrinkage, and hyper-g/n priors are consistent in the sense defined in Ley and Steel (2012), while the rest are not. If the data are generated by a single model from the model space and we use a consistent hyperprior for g, then, as the sample size increases, all posterior mass will tend to be allocated to the true model. It can be proven that hyperpriors that do not depend on n cannot be consistent. More details about g-priors and their properties can be found in Ley and Steel (2012).

Centering

Let's continue with our BMA setup from *Model assumptions and generic formulas*.

A BMA linear regression always includes the intercept in all regressions M_j's. BMA first recenters predictors to make them orthogonal to the intercept. The computation is then performed using the recentered predictors, and the results are transformed back for final reporting.

The model parameters before and after centering are related by a linear transformation

$$\begin{pmatrix} \boldsymbol{\beta} \\ \alpha \end{pmatrix} = \begin{pmatrix} \mathbf{I}_p & 0 \\ -\frac{1}{n}\mathbf{1}'_n\mathbf{X} & 1 \end{pmatrix} \begin{pmatrix} \widetilde{\boldsymbol{\beta}} \\ \tilde{\alpha} \end{pmatrix}$$

where $\widetilde{\boldsymbol{\beta}} = \boldsymbol{\beta}$, $\tilde{\alpha}$ are the model parameters associated with the recentered predictor values $\mathbf{Z}$, and $\mathbf{I}_p$ is the identity matrix of dimension p. The only parameter affected by centering is the intercept.

Conditional posterior distribution of model parameters

See *Model assumptions and generic formulas* for a general BMA model setup.

In this section, we describe a posterior distribution of model parameters conditional on a model and parameter g. This section is based on Steel (2020) and Fernández, Ley, and Steel (2001b).

Consider model M_j, with $j \in J_F$ (without loss of generality) and parameter g, fixed or random, of a Zellner's g-prior. If g is fixed, it is independent of model parameters, and the conditioning on it in the formulas below is unnecessary. Consider the following definitions:

$$\tilde{\boldsymbol{\theta}}_j = (\boldsymbol{\beta}_j', \tilde{\alpha})'$$

$$\delta = \frac{g}{1+g} \quad \text{(shrinkage factor)}$$

$$\widehat{\boldsymbol{\beta}}_{j,\text{ols}} = (\mathbf{Z}_j'\mathbf{Z}_j)^{-1}\mathbf{Z}_j'\mathbf{y}$$

$$\bar{\mathbf{y}} = \frac{1}{n}\mathbf{1}_n'\mathbf{y}$$

$$\text{TSS}_{\text{ols}} = (\mathbf{y} - \bar{\mathbf{y}})'(\mathbf{y} - \bar{\mathbf{y}})$$

$$\text{RSS}_{j,\text{ols}} = (\mathbf{y} - \bar{\mathbf{y}} - \mathbf{Z}_j\widehat{\boldsymbol{\beta}}_{j,\text{ols}})'(\mathbf{y} - \bar{\mathbf{y}} - \mathbf{Z}_j\widehat{\boldsymbol{\beta}}_{j,\text{ols}})$$

$$R_{j,\text{ols}}^2 = 1 - \text{RSS}_{j,\text{ols}}/\text{TSS}_{\text{ols}}$$

where $\widehat{\boldsymbol{\beta}}_{j,\text{ols}}$, TSS_{ols}, $\text{RSS}_{j,\text{ols}}$ and $R_{j,\text{ols}}^2$ are, respectively, a $p_j \times 1$ vector of regression coefficient estimates, the total sum of squares, the residual sum of squares, and the R^2 for the OLS regression of $\mathbf{y}$ on $\mathbf{Z}_j$ with the intercept $\tilde{\alpha}$.

Define the scale factor:

$$s_{j,\delta}^2 = \delta\text{RSS}_{j,\text{ols}} + (1-\delta)\text{TSS}_{\text{ols}} \tag{1}$$

Conditional likelihood. The marginal distribution of $\mathbf{y}$ given g and model M_j has the following analytical form,

$$p(\mathbf{y}|g, M_j) \propto \text{TSS}_{\text{ols}} (g+1)^{\frac{n-p_j-1}{2}} \left\{1 + g(1 - R_{j,\text{ols}}^2)\right\}^{-\frac{n-1}{2}} \tag{2}$$

where the proportionality constant is the same for all the models. If $p_j = 0$, that is, the model includes only the intercept, the marginal distribution is $p(\mathbf{y}|g, M_j) \propto \text{TSS}_{\text{ols}}$.

Conditional posterior for model parameters. Given g and model M_j, the posterior distribution of $(\boldsymbol{\beta}_j, \tilde{\alpha})$ is a multivariate t-distribution with $(n-1)$ degrees of freedom, a $(p+1) \times 1$ location vector $\boldsymbol{\mu}_j$, and a $(p+1) \times (p+1)$ scale matrix $s_{j,\delta}^2\boldsymbol{\Sigma}_j/(n-1)$, where

$$\boldsymbol{\mu}_j = \begin{pmatrix} \delta\widehat{\boldsymbol{\beta}}_{j,\text{ols}} \\ \bar{\mathbf{y}} \end{pmatrix} \tag{3}$$

and

$$\boldsymbol{\Sigma}_j = \begin{Bmatrix} \delta(\mathbf{Z}_j'\mathbf{Z}_j)^{-1} & \mathbf{0} \\ \mathbf{0} & \frac{1}{n} \end{Bmatrix} \tag{4}$$

The posterior distribution of σ^2 given g and M_j is inverse gamma with shape parameter $(n-1)/2$ and scale parameter $s_{j,\delta}^2/2$ defined in (1).

Conditional posterior moments. The conditional posterior mean and variance of $(\boldsymbol{\beta}_j, \tilde{\alpha})$ are

$$E\{(\boldsymbol{\beta}_j, \tilde{\alpha})|\mathbf{y}, g, M_j\} = \boldsymbol{\mu}_j$$

$$\text{Var}\{(\boldsymbol{\beta}_j, \tilde{\alpha})|\mathbf{y}, g, M_j\} = \frac{s_{j,\delta}^2}{n-3}\boldsymbol{\Sigma}_j$$

where $\boldsymbol{\mu}_j$, $\boldsymbol{\Sigma}_j$, and $s_{j,\delta}^2$ are defined in (3), (4), and (1), respectively.

The conditional posterior mean of σ^2 is

$$E(\sigma^2|\mathbf{y}, g, M_j) = \frac{s_{j,\delta}^2}{n-3}$$

Conditional posterior predictive distribution

Consider predicted values $\mathbf{y}^*$ of size $(q \times 1)$ and new centered predictor values $\mathbf{Z}^*$ of size $(q \times p)$. Let $q \times p_j$ matrix $\mathbf{Z}_j^*$ be the predictor values corresponding to the centered predictors included in model M_j. Under model M_j, the posterior predictive distribution is a multivariate t-distribution with $(n-1)$ degrees of freedom, $q \times 1$ location parameter vector $\boldsymbol{\mu}_j^*$, and $q \times q$ scale matrix $\boldsymbol{\Sigma}_j^*$, where

$$\boldsymbol{\mu}_j^* = \bar{\mathbf{y}}\,\mathbf{1}_q + \delta\mathbf{Z}_j^*\,\widehat{\boldsymbol{\beta}}_{j,\text{ols}} \tag{5}$$

$$\boldsymbol{\Sigma}_j^* = \frac{s_{j,\delta}^2}{n-1}\left\{\mathbf{I}_q + \frac{1}{n}\mathbf{1}_q\mathbf{1}_q' + \delta\mathbf{Z}_j^*(\mathbf{Z}_j'\,\mathbf{Z}_j)^{-1}(\mathbf{Z}_j^*)'\right\} \tag{6}$$

and $s_{j,\delta}^2$ is defined in (1).

MCMC algorithms

When the number of predictors is small, $p \leq 24$, bmaregress uses model enumeration to visit all 2^P models. In this case, the formulas provided in the previous sections can be computed exactly. However, when p is large, visiting all possible models is practically infeasible. In this case, MCMC methods are used to approximate the large model space by sampling from it.

Fixed g parameter

A standard procedure used to explore a large model space is the MC3 algorithm (Madigan and York 1995). Recall from *Priors on the model space* that model M can be represented by a binary inclusion vector $\boldsymbol{\gamma}$. A Markov chain $\{\boldsymbol{\gamma}^t\}_{t=1}^T$ is constructed on the model space with the following target distribution:

$$p(\boldsymbol{\gamma}|\mathbf{y}) \propto p(\mathbf{y}|\boldsymbol{\gamma})p(\boldsymbol{\gamma})$$

where $p(\mathbf{y}|\boldsymbol{\gamma}) \equiv p(\mathbf{y}|M)$ is defined in (2) with $M = M_j$, except we do not need to condition on g here because it is fixed.

MC3 sampler for $\boldsymbol{\gamma}$.

1. Initialize model $\boldsymbol{\gamma}^{(0)}$.

2. Let model $\boldsymbol{\gamma}^{(t)} = (\gamma_1^t, \gamma_2^t, \dots, \gamma_p^t)$ be the current state of the chain at iteration t. Uniformly choose coordinate i of $\boldsymbol{\gamma}^{(t)}$, and propose the new model:

$$\boldsymbol{\gamma}^* = (\gamma_1^t, \dots, 1 - \gamma_i^t, \dots, \gamma_p^t)$$

3. Jump to the model $\boldsymbol{\gamma}^*$ with probability

$$\alpha(\boldsymbol{\gamma}, \boldsymbol{\gamma}^*) = \min\left\{\frac{p(\boldsymbol{\gamma}^*|\mathbf{y})}{p(\boldsymbol{\gamma}|\mathbf{y})}, 1\right\}$$

or stay at $\boldsymbol{\gamma}$ if $\alpha(\boldsymbol{\gamma}, \boldsymbol{\gamma}^*) = 1$.

Random g parameter

In this case, both g and $\boldsymbol{\gamma}$ are random, and we construct a Markov chain based on the following conditional statements:

$$p(\boldsymbol{\gamma}|g, \mathbf{y}) \propto p(\mathbf{y}|g, \boldsymbol{\gamma})p(\boldsymbol{\gamma})$$

$$p(g|\boldsymbol{\gamma}, \mathbf{y}) \propto p(\mathbf{y}|g, \boldsymbol{\gamma})p(g)$$

where $p(\mathbf{y}|g, \boldsymbol{\gamma}) \equiv p(\mathbf{y}|g, M)$ is defined in (2) with $M = M_j$.

For sampling the model $\boldsymbol{\gamma}$, the MC3 sampler described above is used. For sampling g, an adaptive random-walk Metropolis sampler is used (Atchadé and Rosenthal 2005). This includes an MH step with a lognormal proposal centered at the previous value. The variance of the lognormal proposal is tuned to result in an optimal acceptance rate $\alpha_{\text{opt}} = 0.44$ (Gelman, Gilks, and Roberts 1997).

For a and A, where $0 < a < A$, we consider the following truncation function:

$$h_{a,A}(x) = \begin{cases} a & \text{if } x \le a \\ x & \text{if } a < x \le A \\ A & \text{if } x > A \end{cases}$$

Let $\text{LN}(\mu, \sigma)$ be the lognormal distribution with mean μ and standard deviation σ. Its density is denoted by $f_{\text{LN}}(\cdot; \mu, \sigma)$.

Modified MC3 sampler for $(g, \boldsymbol{\gamma})$.

This algorithm is based on Ley and Steel (2012).

Let $\{g^{(t)}, \boldsymbol{\gamma}^{(t)}\}$ be the current state of the chain at iteration t, where $\boldsymbol{\gamma}^{(t)} = (\gamma_1^t, \gamma_2^t, \dots, \gamma_p^t)$. For each t, we set $\sigma^{(t)} = h_{a,A}(\rho^{(t)})$, where $\rho^{(t)}$ is an adaptation parameter updated periodically during the burn-in period.

1. Initialize $\{g^{(0)}, \boldsymbol{\gamma}^{(0)}\}$ according to the initialization options `minitial()` and `ginitial()`, and let $a = 0.0001$, $A = 10,000$, and $\rho^{(0)} = 2.38$.

2. Sample $\boldsymbol{\gamma}^{(t+1)}$.

2.1. Uniformly choose coordinate i of $\boldsymbol{\gamma}$ and propose a new model:

$$\boldsymbol{\gamma}^* = (\gamma_1^t, \ldots, 1 - \gamma_i^t, \ldots, \gamma_p^t)$$

2.2. Move to a new model, $\boldsymbol{\gamma}^{(t+1)} = \boldsymbol{\gamma}^*$, with probability

$$\alpha(\boldsymbol{\gamma}^{(t)}, \boldsymbol{\gamma}^*) = \min\left\{\frac{p(\boldsymbol{\gamma}^*|g^{(t)}, \mathbf{y})}{p(\boldsymbol{\gamma}^{(t)}|g^{(t)}, \mathbf{y})}, 1\right\}$$

Otherwise, stay at the current model $\boldsymbol{\gamma}^{(t+1)} = \boldsymbol{\gamma}^{(t)}$.

3. Sample $g^{(t+1)}$.

3.1. Generate $g^* \sim \mathrm{LN}(g^{(t)}, \sigma^{(t)})$.

3.2. Move to the new value $g^{(t+1)} = g^*$ with probability

$$\alpha(g^{(t)}, g^*) = \min\left\{\frac{p(g^*|\boldsymbol{\gamma}^{(t+1)}, \mathbf{y})}{p(g^{(t)}|\boldsymbol{\gamma}^{(t+1)}, \mathbf{y})} \frac{f_{\mathrm{LN}}(g^{(t)}; g^*, \sigma^{(t)})}{f_{\mathrm{LN}}(g^*; g^{(t)}, \sigma^{(t)})}, 1\right\}$$

Otherwise, stay at the current value $g^{(t+1)} = g^{(t)}$.

3.3. After each period of a constant number of iterations, 200 by default, update $\rho^{(t)}$ according to the adaptation procedure explained in *Adaptive MH algorithm* in [BAYES] **bayesmh**, and update $\sigma^{(t)} = h_{a,A}(\rho^{(t)})$.

Inference

As we mentioned in *Model assumptions and generic formulas*, the BMA inference is focused on PMPs and PIPs. We define these quantities below. We also provide the formulas for the posterior means and standard deviations of model parameters reported by bmaregress. The formulas in this section depend on formulas in *Conditional posterior distribution of model parameters*.

The computations depend on whether parameter g is fixed or random. For a fixed g, the computations also depend on whether the model space was fully explored by enumeration or sampling was used. Below, we provide a setup for each case, which will be used in subsequent subsections.

Fixed g, model enumeration. With model enumeration, analytical formulas are available for PMPs, PIPs, and posterior distributions of model parameters. They depend on the fully enumerated BMA model space $\mathcal{M}_F$ indexed by $J_F = \{1, 2, \ldots, 2^p\}$. In bmaregress, these computations are available by default when $p \leq 12$ or when the enumeration option is specified. enumeration is not available with $p > 24$.

Fixed g, sampling. Model enumeration may not feasible with many predictors, for example, $p > 24$. Sampling of the model space is used in this case instead of enumeration. So we have a subspace of distinct models $\mathcal{M}_J$ visited by the Markov chain indexed by $J \subset J_F$, and we have the Markov chain, a sample of models, $\{m_t\}_{t=1}^T$. We can still compute various quantities analytically, but now they will be conditional on the visited model space $\mathcal{M}_J$. Alternatively, we can compute these quantities from the MCMC sample of models $\{m_t\}_{t=1}^T$ by using MCMC frequencies. You can use bmaregress's sampling option to request sampling. bmaregress uses it automatically whenever $p > 12$. Some postestimation

commands also support the `frequency` option to request that MCMC-based frequency estimates of PMP, PIP, and other quantities are used instead of the analytical formulas. Other postestimation commands provide both analytical and frequency computations automatically, whenever they are available.

Random g. With a random g, model enumeration is difficult (if not impossible), and sampling is always used. In general, analytical formulas are not available, so all computations are based on an MCMC sample. In this case, the sample also includes g, $\{g_t, m_t\}_{t=1}^{T}$, and represents a sample from the joint posterior distribution of g and M. bmaregress automatically uses sampling with random g, and it is the only option in this case.

In what follows, we provide formulas for each of the above cases, except we consider a fixed g as one case, where a subset J is either a full set J_F or a subset of the full set.

Posterior model probability

The PMP of model M_j is denoted by $P(M_j|\mathbf{y})$. There are two ways to compute PMP. The first way, referred to as analytical PMP, is based on the analytical form of the marginal likelihood in (2). The second way, referred to as MCMC frequency or simply frequency PMP, is based on the frequency of the models visited by the Markov chain.

1. **Fixed g**.

The analytical PMP of a model M_j is computed by the formula

$$P_a(M_j|\mathbf{y}) = \frac{P(\mathbf{y}|M_j)P(M_j)}{\sum_{l \in J} P(\mathbf{y}|M_l)P(M_l)} \tag{7}$$

where $J = J_F$ with model enumeration and $P(\mathbf{y}|M_j)$ is defined in (2).

When an MCMC sample of models is available, the frequency PMP is computed as follows,

$$P_f(M_j|\mathbf{y}) = \frac{T_j}{T}; \; T_j = \sum_{t=1}^{T} I(m_t = M_j) \tag{8}$$

where $I(m_t = M_j)$ denotes the indicator function, which is 1 if $m_t = M_j$ and 0 otherwise.

The bmaregress command computes results based on analytical PMP whenever it is available. Therefore, for a fixed g, the term PMP refers to analytical PMP, if not specified otherwise.

The comparison of the frequency PMP to the analytical PMP is used to assess convergence of MCMC. The sampling correlation reported in the header of bmaregress is the correlation between the analytical and frequency PMPs.

2. **Random g**.

In general, when g is random, the formula for the analytical PMP is not available. In this case, bmaregress uses the frequency PMP as defined in (8).

Because analytical PMP is not available, the sampling correlation in the random g case is computed as the correlation between the sequences $\{P_h(M_j|\mathbf{y})\}$ and $\{P_f(M_j|\mathbf{y})\}$, where $P_h(M_j|\mathbf{y})$ is the harmonic-mean estimator of $P(M_j|\mathbf{y})$ (Geweke 1989):

$$P_h(M_j|\mathbf{y}) = \frac{P_h(\mathbf{y}|M_j)P(M_j)}{\sum_{l \in J} P_h(\mathbf{y}|M_l)P(M_l)}$$

where

$$P_h(\mathbf{y}|M_j) = \left\{ \frac{1}{T_j} \sum_{t:\, m_t = M_j} p^{-1}(\mathbf{y}|g_t, m_t) \right\}^{-1}$$

Posterior inclusion probability

Consider a predictor X_k for $1 \leq k \leq p$. The PIP of X_k is defined as the sum of posterior probabilities of models that include X_k. Below, we consider fixed and random g cases separately.

1. **Fixed g**.

For a model M_j, denote by $X_k \in M_j$ that X_k is included in M_j. Then, the analytical PIP of X_k is computed as

$$\text{PIP}_k^a = \sum_{j \in J} I(X_k \in M_j) P_a(M_j|\mathbf{y}) \tag{9}$$

where $I(\cdot)$ is the indicator function and $P_a(M_j|\mathbf{y})$ is defined in (7). With sampling, you can also compute the frequency PIP based on the MCMC sample. It is computed by replacing $P_a(M_j|\mathbf{y})$ with $P_f(M_j|\mathbf{y})$, defined by (8), in the above formula.

2. **Random g**.

For a random g, the PIP of X_k is computed as follows,

$$\text{PIP}_k^f = \sum_{j \in J} I(X_k \in M_j) P_f(M_j|\mathbf{y})$$

where $P_f(M_j|\mathbf{y})$ is defined by (8).

Posterior distributions of regression coefficients

Consider a predictor X_k for $1 \leq k \leq p$. Let β_k be the regression coefficient associated with predictor X_k. The posterior distribution of β_k has a mixed structure, which consists of a continuous portion (a mixture of location-scale t-distributions) and a discrete portion (point mass at 0). Also see *Methods and formulas* of [BMA] **bmagraph coefdensity**.

1. **Fixed g**.

The analytical posterior distribution of β_k is

$$p(\beta_k|\mathbf{y}) = (1 - \text{PIP}_k^a)\, \boldsymbol{\delta}_0 + \text{PIP}_k^a\, \mathbf{f}_k$$

where $\boldsymbol{\delta}_0$ denotes the point mass at 0, and

$$\mathbf{f}_k = \sum_{j \in J} I(X_k \in M_j) p(\beta_k|M_j, \mathbf{y}) \frac{P_a(M_j|\mathbf{y})}{\text{PIP}_k^a} \tag{10}$$

In the above, $p(\beta_k|M_j, \mathbf{y})$ is a location-scale t-distribution with the model-specific location vector and scale matrix as defined, respectively, in (3) and (4). $P_a(M_j|\mathbf{y})$ is defined in (7) and PIP_k^a is defined in (9).

2. **Random g**.

There is no analytical formula for the posterior distribution of a regression coefficient. A kernel density estimator is used to approximate $\mathbf{f}_k$ in the above based on the sample of coefficients generated by the `bmacoefsample` command. See *Methods and formulas* of [BMA] **bmagraph coefdensity** for details.

Posterior means and variances of model parameters

1. **Fixed g**.

Let $\tilde{\boldsymbol{\theta}}_j$ be a $(p_j + 1) \times 1$ vector of regression coefficients with intercept $\tilde{\alpha}$ for model M_j in the centered parameterization. Let $\tilde{\boldsymbol{\theta}} = (\boldsymbol{\beta}', \tilde{\alpha})'$ be a $(p + 1) \times 1$ parameter vector in the centered parameterization and $\boldsymbol{\theta} = (\boldsymbol{\beta}', \alpha)'$ be a $(p + 1) \times 1$ parameter vector in the original, uncentered parameterization.

Then,

$$E(\tilde{\boldsymbol{\theta}}_j|\mathbf{y}, g, M_j) = \boldsymbol{\mu}_j$$

$$\text{Var}(\tilde{\boldsymbol{\theta}}_j|\mathbf{y}, g, M_j) = \boldsymbol{\Sigma}_j$$

where $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$ are defined in (3) and (4).

The analytical unconditional mean and variance for the centered parameter vector $\tilde{\boldsymbol{\theta}}$ are defined as follows,

$$E(\tilde{\boldsymbol{\theta}}|\mathbf{y}) = \sum_{j \in J} P_a(M_j|\mathbf{y}) E(\tilde{\boldsymbol{\theta}}_j|\mathbf{y}, g, M_j) = \sum_{j \in J} P_a(M_j|\mathbf{y}) \boldsymbol{\mu}_j$$

$$\text{Var}(\tilde{\boldsymbol{\theta}}|\mathbf{y}) = \sum_{j \in J} P_a(M_j|\mathbf{y}) \left(\boldsymbol{\Sigma}_j + \boldsymbol{\mu}_j \boldsymbol{\mu}_j' \right) - E(\tilde{\boldsymbol{\theta}}|\mathbf{y}) E(\tilde{\boldsymbol{\theta}}|\mathbf{y})'$$

where $P_a(M_j|\mathbf{y})$ is defined in (7).

The analytical unconditional posterior mean and variance of $\boldsymbol{\theta}$ are

$$E(\boldsymbol{\theta}|\mathbf{y}) = \mathbf{A} E(\tilde{\boldsymbol{\theta}}|\mathbf{y}) \tag{11}$$

$$\text{Var}(\boldsymbol{\theta}|\mathbf{y}) = \mathbf{A} \, \text{Var}(\tilde{\boldsymbol{\theta}}|\mathbf{y}) \mathbf{A}' \tag{12}$$

where $A = \begin{pmatrix} \mathbf{I}_p & 0 \\ -\frac{1}{n}\mathbf{1}_n'\mathbf{X} & 1 \end{pmatrix}$.

The analytical unconditional posterior mean of σ^2 is

$$E(\sigma^2|\mathbf{y}) = \sum_{j \in J} P_a(M_j|\mathbf{y}) E(\sigma^2|\mathbf{y}, g, M_j) = \sum_{j \in J} P_a(M_j|\mathbf{y}) \frac{s_{j,\delta}^2}{n - 3} \tag{13}$$

2. **Random g**.

Because the analytical form of $p(\beta_k|M_j, \mathbf{y})$ is not available with a random g, bmaregress computes Monte Carlo estimators of the posterior mean and variance of β_k's.

The frequency estimate of the unconditional mean and variance for the centered parameter vector $\tilde{\boldsymbol{\theta}}$ are defined as follows:

$$\widehat{E}(\tilde{\boldsymbol{\theta}}|\mathbf{y}) = \frac{1}{T}\sum_{t=1}^{T} E(\tilde{\boldsymbol{\theta}}_t|\mathbf{y}, g_t, m_t) = \frac{1}{T}\sum_{t=1}^{T}\boldsymbol{\mu}_t$$

$$\widehat{\text{Var}}(\tilde{\boldsymbol{\theta}}|\mathbf{y}) = \frac{1}{T}\sum_{t=1}^{T}\left(\boldsymbol{\Sigma}_t + \boldsymbol{\mu}_t\boldsymbol{\mu}_t'\right) - \widehat{E}(\tilde{\boldsymbol{\theta}}|\mathbf{y})\widehat{E}(\tilde{\boldsymbol{\theta}}|\mathbf{y})'$$

where $\boldsymbol{\mu}_t$ is defined in (3) and $\boldsymbol{\Sigma}_t$ is defined in (4), in which index j is replaced with the iteration t.

The frequency estimate of the unconditional posterior mean and variance of $\boldsymbol{\theta}$ are

$$\widehat{E}(\boldsymbol{\theta}|\mathbf{y}) = \mathbf{A}\widehat{E}(\tilde{\boldsymbol{\theta}}|\mathbf{y})$$

$$\widehat{\text{Var}}(\boldsymbol{\theta}|\mathbf{y}) = \mathbf{A}\,\widehat{\text{Var}}(\tilde{\boldsymbol{\theta}}|\mathbf{y})\mathbf{A}'$$

where $A = \begin{pmatrix} \mathbf{I}_p & 0 \\ -\frac{1}{n}\mathbf{1}_n'\mathbf{X} & 1 \end{pmatrix}$.

The unconditional posterior mean of σ^2 estimated based on the MCMC frequencies is

$$\widehat{E}(\sigma^2|\mathbf{y}) = \frac{1}{T}\sum_{t=1}^{T} E\left[\{\sigma^{(t)}\}^2|\mathbf{y}, g_t, m_t\right] = \frac{1}{T}\sum_{t=1}^{T}\frac{s_{t,\delta}^2}{n-3} \tag{14}$$

where $s_{t,\delta}^2$ is defined by (1) in *Conditional posterior distribution of model parameters* but for model m_t (with index j replaced with iteration t).

References

Atchadé, Y. F., and J. S. Rosenthal. 2005. On adaptive Markov chain Monte Carlo algorithms. *Bernoulli* 11: 815–828. https://doi.org/10.3150/bj/1130077595.

Banner, K. M., and M. D. Higgs. 2017. Considerations for assessing model averaging of regression coefficients. *Ecological Applications* 27: 78–93. https://doi.org/10.1002/eap.1419.

Bayarri, M. J., J. O. Berger, A. Forte, and G. García-Donato. 2012. Criteria for Bayesian model choice with application to variable selection. *Annals of Statistics* 40: 1550–1577. https://doi.org/10.1214/12-AOS1013.

Becker, G. S. 1968. Crime and punishment: An economic approach. *Journal of Political Economy* 76: 169–217. https://doi.org/10.1086/259394.

Brier, S. S., and S. E. Fienberg. 1980. Recent econometric modeling of crime and punishment: Support for the deterrence hypothesis? *Evaluation Review* 4: 147–191. https://doi.org/10.1177/0193841X8000400201.

Chatterjee, S., and A. S. Hadi. 2012. *Regression Analysis by Example.* 5th ed. New York: Wiley.

Draper, D. 1999. Comment [on Hoeting et al. (1999)]. *Statistical Science* 14: 405–409.

Ehrlich, I. 1973. Participation in illegitimate activities: A theoretical and empirical investigation. *Journal of Political Economy* 81: 521–565. https://doi.org/10.1086/260058.

Eicher, T. S., C. Papageorgiou, and A. E. Raftery. 2011. Default priors and predictive performance in Bayesian model averaging, with application to growth determinants. *Journal of Applied Econometrics* 26: 30–55. https://doi.org/10.1002/jae.1112.

Fernández, C., E. Ley, and M. F. J. Steel. 2001a. Benchmark priors for Bayesian model averaging. *Journal of Econometrics* 100: 381–427. https://doi.org/10.1016/S0304-4076(00)00076-2.

———. 2001b. Model uncertainty in cross-country growth regressions. *Journal of Applied Econometrics* 16: 563–576. https://doi.org/10.1002/jae.623.

Fletcher, D. 2018. *Model Averaging*. Berlin: Springer.

Foster, D. P., and E. I. George. 1994. The risk inflation criterion for multiple regression. *Annals of Statistics* 22: 1947–1975. https://doi.org/10.1214/aos/1176325766.

Fragoso, T. M., W. Bertoli, and F. Louzada. 2018. Bayesian model averaging: A systematic review and conceptual classification. *International Statistical Review* 86: 1–28. https://doi.org/10.1111/insr.12243.

Gelman, A., W. R. Gilks, and G. O. Roberts. 1997. Weak convergence and optimal scaling of random walk Metropolis algorithms. *Annals of Applied Probability* 7: 110–120. https://doi.org/10.1214/aoap/1034625254.

Geweke, J. 1989. Bayesian inference in econometric models using Monte Carlo integration. *Econometrica* 57: 1317–1339. https://doi.org/10.2307/1913710.

Hoeting, J. A., D. Madigan, A. E. Raftery, and C. T. Volinsky. 1999. Bayesian model averaging: A tutorial. *Statistical Science* 14: 382–417. https://doi.org/10.1214/ss/1009212519.

Kass, R. E., and A. E. Raftery. 1995. Bayes factors. *Journal of the American Statistical Association* 90: 773–795. https://doi.org/10.1080/01621459.1995.10476572.

Kass, R. E., and L. Wasserman. 1995. A reference Bayesian test for nested hypotheses and its relationship to the Schwarz criterion. *Journal of the American Statistical Association* 90: 928–934. https://doi.org/10.1080/01621459.1995.10476592.

Leamer, E. E. 1973. Multicollinearity: A Bayesian interpretation. *Review of Economics and Statistics* 55: 371–380. https://doi.org/10.2307/1927962.

———. 1978. *Specification Searches: Ad Hoc Inference with Nonexperimental Data*. New York: Wiley.

Ley, E., and M. F. J. Steel. 2009. On the effect of prior assumptions in Bayesian model averaging with applications to growth regression. *Journal of Applied Econometrics* 24: 651–674. https://doi.org/10.1002/jae.1057.

———. 2012. Mixtures of *g*-priors for Bayesian model averaging with economic applications. *Journal of Econometrics* 171: 251–266. https://doi.org/10.1016/j.jeconom.2012.06.009.

Liang, F., R. Paulo, G. Molina, M. A. Clyde, and J. O. Berger. 2008. Mixtures of *g* priors for Bayesian variable selection. *Journal of the American Statistical Association* 103: 410–423. https://doi.org/10.1198/016214507000001337.

Madigan, D., and J. York. 1995. Bayesian graphical models for discrete data. *Journal of Statistical Review* 63: 215–232. https://doi.org/10.2307/1403615.

Moral-Benito, E. 2015. Model averaging in economics: An overview. *Journal of Economic Surveys* 29: 46–75. https://doi.org/10.1111/joes.12044.

Porwal, A., and A. E. Raftery. 2022a. Effect of model space priors on statistical inference with model uncertainty. *New England Journal of Statistics in Data Science* 1: 149–158. https://doi.org/10.51387/22-NEJSDS14.

———. 2022b. Comparing methods for statistical inference with model uncertainty. *PNAS* 119(16): e2120737119. https://doi.org/10.1073/pnas.2120737119.

Raftery, A. E., D. Madigan, and J. A. Hoeting. 1997. Bayesian model averaging for linear regression models. *Journal of the American Statistical Association* 92: 179–191. https://doi.org/10.1080/01621459.1997.10473615.

Sala-I-Martin, X. X. 1997. I just ran two million regressions. *American Economic Review* 87: 178–183.

Steel, M. F. J. 2020. Model averaging and its use in economics. *American Economic Review* 58: 644–719. https://doi.org/10.1257/jel.20191385.

Vandaele, W. 1978. "Participation in illegitimate activities: Ehrlich revisited". In *Deterrence and Incapacitation: Estimating the Effects of Criminal Sanctions on Crime Rates*, edited by A. Blumstein, J. Cohen, and D. Nagin, 270–335. Washington, DC: National Academy of Sciences.

Zellner, A. 1986. "On assessing prior distributions and Bayesian regression analysis with *g*-prior distributions". In *Bayesian Inference and Decision Techniques: Essays in Honor of Bruno De Finetti*. Studies in Bayesian Econometrics and Statistics, edited by P. K. Goel and A. Zellner, vol. 6: 233–343. Amsterdam: North-Holland.

Zellner, A., and A. Siow. 1980. Posterior odds ratios for selected regression hypotheses. *Trabajos de estadística y de investigación operativa* 31: 585–603. https://doi.org/10.1007/BF02888369.

Also see

[BMA] **BMA postestimation** — Postestimation tools for Bayesian model averaging

[BMA] **Intro** — Introduction to Bayesian model averaging

[BMA] **BMA commands** — Introduction to commands for Bayesian model averaging

[BMA] **Glossary**

[BAYES] **bayes: regress** — Bayesian linear regression

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

[D] **splitsample** — Split data into random samples

[D] **vl** — Manage variable lists

[R] **regress** — Linear regression

[U] **20 Estimation and postestimation commands**