STATA BAYESIAN MODEL AVERAGING REFERENCE MANUAL

RELEASE 19

The suggested citation for this software is

> StataCorp. 2025. *Stata 19*. Statistical software. StataCorp LLC.

The suggested citation for this manual is

> StataCorp. 2025. *Stata 19 Bayesian Model Averaging Reference Manual*. College Station, TX: Stata Press.

www.stata.com

Contents

i

Cross-referencing the documentation

When reading this manual, you will find references to other Stata manuals, for example, [U] **27 Overview of Stata estimation commands**; [R] **regress**; and [D] **reshape**. The first example is a reference to chapter 27, *Overview of Stata estimation commands*, in the *User's Guide*; the second is a reference to the regress entry in the *Base Reference Manual*; and the third is a reference to the reshape entry in the *Data Management Reference Manual*.

All the manuals in the Stata Documentation have a shorthand notation:

[GSM]	*Getting Started with Stata for Mac*
[GSU]	*Getting Started with Stata for Unix*
[GSW]	*Getting Started with Stata for Windows*
[U]	*Stata User's Guide*
[R]	*Stata Base Reference Manual*
[ADAPT]	*Stata Adaptive Designs: Group Sequential Trials Reference Manual*
[BAYES]	*Stata Bayesian Analysis Reference Manual*
[BMA]	*Stata Bayesian Model Averaging Reference Manual*
[CAUSAL]	*Stata Causal Inference and Treatment-Effects Estimation Reference Manual*
[CM]	*Stata Choice Models Reference Manual*
[D]	*Stata Data Management Reference Manual*
[DSGE]	*Stata Dynamic Stochastic General Equilibrium Models Reference Manual*
[ERM]	*Stata Extended Regression Models Reference Manual*
[FMM]	*Stata Finite Mixture Models Reference Manual*
[FN]	*Stata Functions Reference Manual*
[G]	*Stata Graphics Reference Manual*
[H2OML]	*Machine Learning in Stata Using H2O: Ensemble Decision Trees Reference Manual*
[IRT]	*Stata Item Response Theory Reference Manual*
[LASSO]	*Stata Lasso Reference Manual*
[XT]	*Stata Longitudinal-Data/Panel-Data Reference Manual*
[META]	*Stata Meta-Analysis Reference Manual*
[ME]	*Stata Multilevel Mixed-Effects Reference Manual*
[MI]	*Stata Multiple-Imputation Reference Manual*
[MV]	*Stata Multivariate Statistics Reference Manual*
[PSS]	*Stata Power, Precision, and Sample-Size Reference Manual*
[P]	*Stata Programming Reference Manual*
[RPT]	*Stata Reporting Reference Manual*
[SP]	*Stata Spatial Autoregressive Models Reference Manual*
[SEM]	*Stata Structural Equation Modeling Reference Manual*
[SVY]	*Stata Survey Data Reference Manual*
[ST]	*Stata Survival Analysis Reference Manual*
[TABLES]	*Stata Customizable Tables and Collected Results Reference Manual*
[TS]	*Stata Time-Series Reference Manual*
[I]	*Stata Index*
[M]	*Mata Reference Manual*

Description

This entry provides a software-free introduction to Bayesian model averaging (BMA). See [BMA] **BMA commands** for a suite of commands to perform BMA. Also see [BAYES] **Intro** for an introduction to Bayesian analysis.

Remarks and examples

Remarks are presented under the following headings:

Brief motivation

Model averaging is a statistical approach that accounts for model uncertainty in your analysis. Instead of relying on just one model, model averaging averages results over multiple plausible models based on the observed data. In Bayesian model averaging (BMA), the "plausibility" of the model is described by the posterior model probability, which is determined using the fundamental Bayesian principles—the Bayes theorem—and applied universally to all data analyses.

Model averaging can be used to account for model uncertainty when estimating model parameters and predicting new observations to avoid overly optimistic conclusions. It is particularly useful in applications with several plausible models, where there is no one definitive reason to choose a particular model over the others. But even if choosing a single model is the end goal, model averaging can be beneficial. For instance, BMA provides a principled way to identify important models and predictors within the considered classes of models. Its framework allows you to learn about interrelations between different predictors in terms of their tendency to appear in a model together, separately, or independently. It can be used to evaluate the sensitivity of the final results to various assumptions about the importance of different models and predictors. And it provides optimal predictions in the log-score sense.

What is model averaging and why do we need it?

The concept of a model is central in statistics. Classical statistical inference is based on the assumption that there is an underlying data-generating model (DGM), and we can infer its characteristics from the observed data. Selecting an appropriate model for the problem at hand is the first and crucial step in performing statistical analyses. In some applications, we may have a strong theoretical or empirical

evidence about the DGM. In other applications, usually of complex and unstable nature, such as those in economics, psychology, and epidemiology, choosing a single reliable model can be difficult. In such cases, it is important to have a principled way to account for the uncertainty in the model-selection process. In practice, we are often interested in a particular property or quantity of the DGM. The classical inferential approach involves choosing a model and estimating this quantity from the observed data conditional on the chosen model. One drawback of working with a single model is that we may assign more precision to our estimates than is supported by the data (Chatfield 1995 and Draper 1995). In predictive inference, single-model approaches do not utilize all available information and may be unstable; see, for example, Piironen and Vehtari (2017).

The model averaging approach is conceptually different. Instead of choosing one model, we consider a list of candidate models. The quantity of interest is then estimated by an average across individual model estimates. Averaging is weighed by how likely each model is. In this way, model averaging accounts for the model-selection uncertainty.

The true DGM may or may not be in our list of candidate models. If it is, classical model-selection approaches may work well. Otherwise, the larger the candidate model space is, the greater the possibility of model selection to choose an incorrect model and make wrong conclusions. And the selected model may change every time the new data become available. In model selection, it may not be clear what constitutes a good candidate, given that the true model is unknown. Popular information-based criteria such as the Bayesian information criterion measure how well a model fits the data and include an additional penalty for its complexity. But a model often needs to be evaluated based on its predictive performance. Improving predictive performance motivated a variety of methods known as ensemble methods such as stacking (Wolpert 1992) and bagging (Breiman 1996). Model averaging can be viewed as an ensemble method that improves predictive performance using optimal combinations in the space of considered candidate models (Raftery and Zheng 2003).

Bayesian model averaging (BMA)

BMA (Leamer 1978) casts model averaging into a Bayesian framework. It provides a principled way to define model weights as posterior model probabilities, which is universal to all data-generating processes. BMA formulation emerges naturally as an application of a standard Bayesian predictive approach to model averaging.

In BMA, model M is a random variable with prior $P(M)$ distributed over some model space. Given the observed data D, the likelihood of M is the probability of D with respect to M, $P(D|M)$. The posterior of M is then given by the Bayes theorem

$$P(M|D) = \frac{P(D|M)P(M)}{\sum_{M^*} P(D|M^*)P(M^*)}$$

where we assume that the model space is discrete and take the sum over it in the denominator. Continuous model spaces are also possible but will not be considered here. The quantity $P(D|M)$ is known as the marginal likelihood of model M. And $P(M|D)$ is known as the posterior model probability and is a key quantity in BMA inference and prediction. Also see *Concepts of BMA*.

Let Q be any quantity of interest that is not model specific; that is, it should have the same interpretation across all models in the model space. Let Q_M be its estimator with respect to model M. The BMA estimator of Q can be written as

$$Q_{\text{BMA}} = \sum_M P(M|D)Q_M$$

The above formula follows from the fundamental BMA formula for the posterior distribution of Q over the model space,

$$g(Q|D) = \sum_M P(M|D)g(Q|D, M)$$

where $g(Q|D, M)$ is the posterior distribution of Q for model M. Then, $Q_{\text{BMA}} = E(Q|D)$ is the posterior mean of Q, and $Q_M = E(Q|D, M)$ is the posterior mean of Q for model M.

The variability of Q is described by the posterior variance of Q with respect to $g(Q|D)$,

$$\text{Var}(Q|D) = \sum_M P(M|D)\,\text{Var}(Q|D, M) + \sum_M P(M|D)\left\{E(Q|D, M) - E(Q|D)\right\}^2$$

where the second term estimates the additional uncertainty about the estimated Q across models.

In a regression context, the notion of model averaging has a more specific formulation—model uncertainty arises mainly from the uncertainty of which predictors should be included in the model.

Let Y be an outcome variable with p potential predictors (regressors or covariates) $\mathbf{x} = (X_1, X_2, \ldots, X_p)$. Let $D = \{y_i, x_{1i}, x_{2i}, \ldots, x_{pi}\}_{i=1}^n$ be a sample of observations on Y and $\mathbf{x}$. We are not sure which predictors describe Y best and consider any subset of $\mathbf{x}$ as a potential candidate set. We can enumerate all subsets and denote the jth subset by $\mathbf{x}_j$. Then M_j, defined as the model corresponding to $\mathbf{x}_j$, is an element of the discrete model space $\{M_j\}_{j=1}^{2^p}$. Two typical applications of BMA in this context are estimating regression coefficients and predicting Y from a new observation $\mathbf{x}^*$; also see *Usage of BMA* for other applications.

Let $\widehat{\boldsymbol{\beta}}_{M_j}$ be an estimate of a $p \times 1$ regression coefficient vector $\boldsymbol{\beta}$ with respect to model M_j, in which the coefficients for predictors not in the model are set to zero. Then, the BMA estimate of $\boldsymbol{\beta}$ is

$$\widehat{\boldsymbol{\beta}}_{\text{BMA}} = \sum_{j=1}^{2^p} P(M_j|D)\widehat{\boldsymbol{\beta}}_{M_j}$$

Given a new observation $\mathbf{x}^*$, a new outcome value y^* can be obtained from the BMA predictive distribution, which is as a mixture of the model-specific predictive distributions,

$$p_{\text{BMA}}(y^*|\mathbf{x}^*, D) = \sum_{j=1}^{2^p} P(M_j|D)p_j(y^*|\mathbf{x}^*, D, M_j)$$

where $p_j(y^*|\mathbf{x}^*, D, M_j)$ is the posterior predictive density of Y for model M_j. The above is a special case of the standard definition of the Bayesian posterior predictive distribution.

BMA has many appealing statistical properties, as detailed in Steel (2020). For instance, Raftery and Zheng (2003) show that BMA point estimators and predictors minimize the mean squared error weighted by the prior. Madigan and York (1995) verify that BMA estimators maximize predictive ability, and Min and Zellner (1993) find that BMA performs better than any other model-choice approach with respect to the log predictive-score.

BMA is built on Bayesian principles. Thus, it faces the same challenges as standard Bayesian analyses with respect to the specification of priors and the intensity of the computations. We discuss some of these challenges in more detail in the next section.

Concepts of BMA

Here we briefly describe some of the concepts essential to BMA. For details in the context of linear regression, see *Remarks and examples* and *Methods and formulas* of [BMA] **bmaregress**. Also see Hoeting et al. (1999), Fernández, Ley, and Steel (2001a), Moral-Benito (2015), Fragoso, Bertoli, and Louzada (2018), and Steel (2020).

BMA applies standard Bayesian principles to model averaging. Thus, all concepts of Bayesian analysis apply to BMA as well; see [BAYES] **Intro**. Compared with standard Bayesian analyses, which condition on a model, BMA views a model as random and assumes a prior distribution for it.

Model space. The extent to which BMA can properly account for model uncertainty relies on the construction of the model space. BMA results are conditional on the considered model space. If the model space does not include important candidate models, BMA will not be able to consider them and incorporate them in the results. The model space should incorporate any aspects of model uncertainty that needs to be accounted for. For instance, if one is uncertain about various functional forms of predictors, these functional forms (and possibly more) should be included in the model space. Ideally, the model space should contain the DGM, but BMA was found to provide good results even when it does not, as long as the model space is sufficiently large. In that case, BMA approximates the true DGM by a combination of models within the considered class. See Steel (2020) for a detailed discussion of the construction of the BMA model space.

Parameters of interest. When the goal of analysis is an estimation of a parameter of interest, it is important that the parameter has the same interpretation across all models. For instance, see *Interpretation of BMA regression coefficients* in *Remarks and examples* of [BMA] **bmaregress**.

Priors for models and model parameters. Specifying a prior distribution for a model parameter is an integral part of a Bayesian model specification. BMA additionally specifies a prior distribution for a model, typically, over a discrete model space. A variety of model priors and priors for model parameters are suggested in the literature, both informative and noninformative, data agnostic and data driven (Steel 2020).

In the regression context, commonly used priors, such as a Zellner's prior with a fixed g parameter for regression coefficients (Fernández, Ley, and Steel 2001a), provide exact computation of marginal likelihoods. Although computationally convenient, these priors may not always provide the best predictive performance. The application of g-priors with random g parameters (Ley and Steel 2012) allows for more flexible BMA analysis but complicates the model specification and simulation. See *Introduction to BMA for linear regression* in *Remarks and examples* of [BMA] **bmaregress** for the discussion of various priors in the context of BMA linear regression.

As with any Bayesian analysis, in the absence of strong information in the data about the DGM and model parameters, BMA results can be sensitive to the choice of priors. Sensitivity analysis is recommended to investigate the impact of priors on the results.

Estimation: Model enumeration and Markov chain Monte Carlo (MCMC) sampling. Depending on the model complexity, it may be feasible to enumerate and consider all the models in the defined space. In this case, the model space is fully explored. This is rarely feasible in practice. More commonly, MCMC sampling is used to explore the model space more efficiently by considering only more likely models given the observed data, for example, the MCMC model composition (MC3) sampling proposed by Madigan and York (1995). In addition to sampling of the model space, we may also need to use MCMC sampling for model parameters when analytical expressions for their posterior distributions are not available, which is common in practice. When MCMC sampling is used, it is important to verify the convergence of MCMC; see *Convergence of BMA* in *Remarks and examples* of [BMA] **bmaregress** in the context of linear regression.

Posterior model probability (PMP). The PMP is central to all BMA analyses. It represents the probability of a model given the observed data and model's prior. It is used as a weight in BMA estimates of parameters of interest and predictions. It is used to identify influential models. And it is used to compute the posterior inclusion probability (PIP), which is used to identify important predictors. In special cases, the PMP can be estimated exactly or analytically, in which case we refer to it as the analytical PMP. More commonly, however, it is estimated based on the MCMC sample of models, in which case we refer to it as the frequency PMP. Models with high PMPs are of interest in BMA analysis.

Posterior inclusion probability (PIP). The PIP is the probability that a predictor is included in a model computed over the model space given the observed data and the prior model probability. It measures the importance of a predictor. Because the computation of the PIP is based on the PMP, we also distinguish between the analytical PIP and frequency PIP. Predictors with high PIP values, commonly above 0.5, are considered important predictors.

Jointness. Jointness is a concept particular to BMA. Because BMA considers multiple models, it can estimate the tendency of predictors to be included jointly or exclusively across the models. Jointness means that predictors tend to be included together in many models. Such predictors are then viewed as complements, in the sense that their joint inclusion provides additional information in explaining the outcome. Disjointness means that whenever one predictor is included in a model, the other tends to be excluded. Such predictors are viewed as substitutes, meaning that only one of them is needed to explain the outcome.

Inference. In the context of BMA, the inference focuses on exploring influential models, models with high PMPs, and important predictors, predictors with high PIPs. The jointness or disjointness of predictors is often also of interest. When averaging across the model space is applicable for a parameter of interest, the parameter estimation is performed with respect to the posterior distribution over the model space. Although the inference accounts for model uncertainty, it is important to remember that it is still conditional on the explored model space.

Prediction. BMA is commonly used for prediction because of its theoretical properties and empirical performance. When the model space contains the DGM, the BMA predictive mean minimizes the expected squared error loss (Min and Zellner 1993). Madigan and Raftery (1994) compare the BMA predictive performance with that of a single model using the log predictive-score (LPS) and conclude that BMA performs at least as well. See Steel (2020) for more information.

Log predictive-score (LPS). LPS is the negative of the logarithm of the predictive density evaluated at an observation (Good 1952). It is used to assess predictive performance of a model in the context of BMA (for example, Madigan, Gavrin, and Raftery [1995] and Fernández, Ley, and Steel [2001a]). It can also be used to compare model fit.

Diagnostics. Model diagnostics are just as important for BMA as they are for a single-model analysis. Any model checks that are commonly done with one model should be performed during BMA analysis as well. Because of the many models, the application of such checks is not as straightforward. The literature recommends that the checks be performed for the model with all predictors before the estimation and for all high-PMP models after the estimation. For BMA, additional diagnostics include checking MCMC convergence and performing a sensitivity analysis to the prior choices.

Sensitivity analysis. As with any Bayesian analysis, prior sensitivity analysis is important for BMA. In the BMA context, the sensitivity analysis should be performed for both model priors and model parameter priors.

Usage of BMA

Fragoso, Bertoli, and Louzada (2018) identified several main applications of BMA across various disciplines such as "model choice", "combination of multiple models for prediction", and "combined estimation". We will refer to these simply as model choice, parameter estimation, and prediction.

BMA was motivated in the context of prediction to improve out-of-sample predictive performance of a model (for example, Hoeting et al. [1999]). BMA can be shown to produce optimal predictions with respect to the LPS (Min and Zellner 1993) by averaging predictions from multiple models and weighing them by the model's importance. The model's importance is estimated in a principled Bayesian way as a PMP and applied universally to all data-generating processes. A few applications of BMA for prediction can be found in Madigan and Raftery (1994), Raftery, Madigan, and Volinsky (1995), Volinsky et al. (1997), Hoeting et al. (1999), Tobias and Li (2004), Kaplan and Lee (2018), and Darwen (2019).

The use of BMA for model choice amounts to identifying important models and predictors. The importance of a model is based on the estimated PMP. And the importance of a predictor is based on the estimated PIP, the probability that this predictor is included in a model estimated over the considered model space. Some of the applications of model choice include Raftery, Madigan, and Hoeting (1997), Hoeting et al. (1999), Fernández, Ley, and Steel (2001b), Eicher, Papageorgiou, and Raftery (2011), Moral-Benito (2015), Arin and Braunfels (2018), and Peisker (2023).

BMA is also used to estimate a parameter common to all models. As with prediction, the BMA estimate is a weighted average of the model-specific estimates with weights defined by PMPs. For instance, see Hoeting et al. (1999), Koop (2003), Yin and Yuan (2009), Montgomery and Nyhan (2010), and Moral-Benito (2015). But be mindful when using BMA to estimate partial regression coefficients in a linear regression (Draper 1999; Banner and Higgs 2017); see *Interpretation of BMA regression coefficients* in *Remarks and examples* of [BMA] **bmaregress**.

Wasserman (2000) also shows how to use BMA to perform Bayesian variable selection.

See Fragoso, Bertoli, and Louzada (2018) for more references and discussion of the BMA usage in different research areas.

BMA versus frequentist model averaging

Frequentist model averaging (FMA) is an inferential procedure based on the so-called FMA estimator,

$$\widehat{\boldsymbol{\beta}}_{\text{FMA}} = \sum_{j=1}^{2^p} \omega_j \widehat{\boldsymbol{\beta}}_j$$

where $0 \leq \omega_j \leq 1$, $\sum_{j=1}^{2^p} \omega_j = 1$, and $\widehat{\boldsymbol{\beta}}_j$ is an estimator, usually ordinary least squares, of regression parameters for model M_j. The weights ω_j's are chosen such that $\widehat{\boldsymbol{\beta}}_{\text{FMA}}$ has certain asymptotic properties.

In contrast to BMA, where model estimators are weighted by PMPs, in FMA the weights are computed for each model independently and then normalized. The most common choice is $\omega_j \propto \exp(-0.5I_j)$ (Buckland, Burnham, and Augustin 1997), where I_j is an information criterion of the form

$$I_j = -2\log(\widehat{L}_j) + \psi_j$$

This approach includes popular choices such as the Akaike information criterion, $\psi_j = 2p_j$, and Bayesian information criterion, $\psi_j = p_j\log(n)$, where p_j is the number of predictors in the jth model. Other approaches include weights based on Mallow's criterion (Hansen 2007) and cross-validation (Hansen and Racine 2012). A more in-depth exploration of the FMA, as applied in economics in particular, can be found in Moral-Benito (2015).

Compared with FMA, BMA provides a unified and intuitive way to interpret the model's and predictor's importance by using the respective PMPs and PIPs. In fact, the PMPs, which are derived from fundamental Bayesian principles, are used as weights in all BMA computations. BMA can also handle larger model spaces more easily by using efficient MCMC sampling algorithms. Additionally, BMA benefits from several appealing statistical properties such as calibration of credible intervals and optimal prediction in the log-score sense. See Steel (2020) for details.

See De Luca and Magnus (2011) for the implementation of the weighted-average least-squares estimator in Stata.

Computational methods for BMA

For a long time, the use of BMA in practice has been hindered by the lack of computationally feasible estimation methods. Since then, a variety of specialized MCMC methods have been developed to facilitate Bayesian inference. A unique challenge of BMA is the complex nature of the posterior domain—a discrete mixture of models with continuous domains of varying dimensions.

One of the first general sampling methods for BMA was the MC3 (Madigan and York 1995), which is a stochastic method that moves through the model space by changing one predictor, or a group of predictors, at a time.

The availability of the analytical form for the marginal likelihood in linear models leads to fast and efficient MC3 sampling methods. However, analytical marginals are not available for generalized linear models and for most linear BMA models that include hyperparameters such as g-priors. Ley and Steel (2012) proposed an adaptive MC3 method applicable to the latter case. Other adaptive MCMC methods are also available (Atchadé and Rosenthal 2005).

Motivating examples

Consider the following simulated dataset. There are $n = 200$ observations and $p = 10$ predictors. Each predictor x1 through x10 is generated independently from a standard normal distribution. The outcome y is generated according to the following regression model, which we refer to as our DGM,

$$y = 0.5 + 1.2 \times x2 + 5 \times x10 + \epsilon$$

where $\epsilon \sim N(0, 1)$ is a standard normal error term.

```
. use https://www.stata-press.com/data/r19/bmaintro
(Simulated data for BMA example)

. summarize
```

Variable	Obs	Mean	Std. dev.	Min	Max
y	200	.9944997	4.925052	-13.332	13.06587
x1	200	-.0187403	.9908957	-3.217909	2.606215
x2	200	-.0159491	1.098724	-2.999594	2.566395
x3	200	.080607	1.007036	-3.016552	3.020441
x4	200	.0324701	1.004683	-2.410378	2.391406
x5	200	-.0821737	.9866885	-2.543018	2.133524
x6	200	.0232265	1.006167	-2.567606	3.840835
x7	200	-.1121034	.9450883	-3.213471	1.885638
x8	200	-.0668903	.9713769	-2.871328	2.808912
x9	200	-.1629013	.9550258	-2.647837	2.472586
x10	200	.083902	.8905923	-2.660675	2.275681

We consider three toy examples. The first example briefly introduces BMA for linear regression and compares it with standard linear regression. The second example compares the use of regression, stepwise selection, lasso, and BMA for prediction. The third example revisits these tools in a more challenging setting of $n = p$.

Examples are presented under the following headings:

> *Example 1: BMA linear regression*
> *Example 2: BMA for prediction compared with other approaches*
> *Example 3: BMA with small sample size and many predictors, $n \leq p$*

▷ Example 1: BMA linear regression

We first use `regress` to fit a standard linear regression of y on x1 through x10. We specify the predictors by using the shortcut *varlist* notation x1–x10:

```
. regress y x1-x10
```

Source	SS	df	MS		
Model	4607.24837	10	460.724837		
Residual	219.723235	189	1.1625568		
Total	4826.9716	199	24.2561387		

Number of obs	=	200
F(10, 189)	=	396.30
Prob > F	=	0.0000
R-squared	=	0.9545
Adj R-squared	=	0.9521
Root MSE	=	1.0782

y	Coefficient	Std. err.	t	P>|t|	[95% conf. interval]	
x1	.0753537	.0781737	0.96	0.336	-.0788513	.2295587
x2	1.18854	.0716658	16.58	0.000	1.047172	1.329907
x3	-.1871012	.0789484	-2.37	0.019	-.3428344	-.0313679
x4	-.0459335	.0785503	-0.58	0.559	-.2008813	.1090144
x5	.0343498	.0793095	0.43	0.665	-.1220956	.1907953
x6	-.0149194	.0767357	-0.19	0.846	-.1662879	.136449
x7	.007174	.0831239	0.09	0.931	-.1567958	.1711437
x8	-.0384917	.0810626	-0.47	0.635	-.1983953	.1214119
x9	.0968948	.0817218	1.19	0.237	-.0643093	.2580989
x10	5.13251	.0877447	58.49	0.000	4.959426	5.305595
_cons	.617996	.0791152	7.81	0.000	.4619337	.7740582

regress identifies the two true predictors x2 and x10 as "statistically significant" (with p-values less than 0.000). The estimate of the coefficient for x2 is 1.19 with a standard error of 0.072, and the 95% confidence interval (CI) is $[1.05, 1.33]$, which agrees with the true value of 1.2. The estimated coefficient for x10 is 5.13 with a standard error of 0.088, and the 95% CI is $[4.96, 5.31]$, which agrees with the true value of 5. These findings are consistent with our true DGM. regress also reports a p-value of 0.019 for x3, which is not in the DGM, with an estimated coefficient of -0.19 and a 95% CI of $[-0.34, -0.03]$. It might be tempting to use the reported p-values to infer the importance of the predictors, but p-values do not have such interpretation.

Let's now use bmaregress to perform BMA for a linear regression:

```
. bmaregress y x1-x10

Enumerating models ...
Computing model probabilities ...

Bayesian model averaging                        No. of obs        =      200
Linear regression                               No. of predictors =       10
Model enumeration                                        Groups   =       10
                                                         Always   =        0
Priors:                                         No. of models     =    1,024
  Models: Beta-binomial(1, 1)                        For CPMP >= .9 =        9
   Cons.: Noninformative                     Mean model size   =    2.479
   Coef.: Zellner's g
       g: Benchmark, g = 200                 Shrinkage, g/(1+g) =   0.9950
  sigma2: Noninformative                     Mean sigma2       =    1.272
```

y	Mean	Std. dev.	Group	PIP
x2	1.198105	.0733478	2	1
x10	5.08343	.0900953	10	1
x3	-.0352493	.0773309	3	.21123
x9	.004321	.0265725	9	.051516
x1	.0033937	.0232163	1	.046909
x4	-.0020407	.0188504	4	.039267
x5	.0005972	.0152443	5	.033015
x8	-.0005639	.0153214	8	.032742
x7	-8.23e-06	.015497	7	.032386
x6	-.0003648	.0143983	6	.032361
Always				
_cons	.5907923	.0804774	0	1

```
Note: Coefficient posterior means and std. dev. estimated from 1,024 models.
Note: Default priors are used for models and parameter g.
```

We will describe only some of the more relevant information here, but see example 1 of [BMA] **bmaregress** for details about the output of bmaregress.

bmaregress, with the default settings, considered all $2^{10} = 1,024$ possible models based on 10 predictors. Like regress, bmaregress identified the two true predictors, x2 and x10, with the estimated PIPs of 1, labeled as PIP in the table. All other predictors have much lower PIP values, and all but the PIP for x3 are below 10%. Unlike regress, we can use the PIP reported by bmaregress to describe and compare the importance of predictors. PIP genuinely represents the probability of a predictor being included in a model across the considered space of 1,024 possible models. For instance, the PIP of 0.2 for x3 is much lower than that for x2 and x10, so we can conclude that this predictor is not as important. Also, its BMA coefficient (posterior mean) of -0.035 is much closer to 0 than that from regress.

The BMA estimates of 1.2 (rounded) and 5.1 of the coefficients for x2 and x10, respectively, are close to the true values of 1.2 and 5. The respective estimated posterior standard deviations, 0.073 and 0.090, are slightly larger than those from regress. This is expected because the BMA estimates account for the uncertainty about which predictors should be included in the regression model. bmaregress does not report credible intervals by default for computational reasons, but you can obtain them as described in example 5 of [BMA] **bmaregress**. Also, with real-world observational data, we should be mindful when interpreting BMA regression coefficients; see *Interpretation of BMA regression coefficients* in *Remarks and examples* of [BMA] **bmaregress**.

Although BMA does not "select" a model, it does identify some of the influential models that contribute more to the averaged results. In this example, we can already guess which model BMA identified as the top model based on the reported PIP values by bmaregress, but let's use bmastats models to confirm:

```
. bmastats models

Computing model probabilities ...

Model summary          Number of models:
                                 Visited = 1,024
                                 Reported =     5
```

	Analytical PMP	Model size
Rank		
1	.6292	2
2	.1444	3
3	.0258	3
4	.0246	3
5	.01996	3

Variable-inclusion summary

	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
x2	x	x	x	x	x
x10	x	x	x	x	x
x3		x			
x9			x		
x1				x	
x4					x

```
Legend:
   x - estimated
```

As anticipated, the top model with a PMP, Analytical PMP, of 0.63 is the model that contains x2 and x10. The next plausible model based on our sample is the one that also includes x3, but its PMP of 0.14 is much lower.

In the above, bmaregress used the default priors. These priors are offered for convenience and should be carefully evaluated in each application. Also, sensitivity analysis should be performed to evaluate the impact of different priors on the results; see, for example, example 11 of [BMA] **bmaregress**.

In BMA, the variance of the prior for the regression coefficients is proportional to the so-called g parameter. By default, g has a fixed value of $\max(n, p^2)$, which in our example is $g = n = 200$. We can relax this by specifying a higher value for g, say, 1,000. This will reduce the shrinkage effect on the coefficients and generally produce estimates that are closer to the ordinary least-squares estimates.

Another important benefit of BMA is its ability to control model uncertainty through the model prior. If, for example, we had a prior knowledge that predictors x1 and x3 through x9 are unlikely to be related to y, we could incorporate this knowledge in our BMA model. In the following specification, we use the mprior() option to specify the binomial model prior with the inclusion probability of 0.1 for x1 and x3 through x9 and the inclusion probability of 0.5 for x2 and x10.

```
. bmaregress y x1-x10, mprior(binomial x2 x10 0.5 x1 x3-x9 0.1)
Enumerating models ...
Computing model probabilities ...
Bayesian model averaging                   No. of obs         =       200
Linear regression                          No. of predictors  =        10
Model enumeration                                     Groups  =        10
                                                      Always  =         0
Priors:                                    No. of models      =     1,024
  Models: Binomial, IP varies                 For CPMP >= .9  =         2
   Cons.: Noninformative                   Mean model size    =     2.129
   Coef.: Zellner's g
       g: Benchmark, g = 200               Shrinkage, g/(1+g) =    0.9950
  sigma2: Noninformative                   Mean sigma2        =     1.276
```

y	Mean	Std. dev.	Group	PIP
x2	1.200944	.0730381	2	1
x10	5.080663	.0899736	10	1
x3	−.0106068	.0452704	3	.064039
x9	.0009677	.0126993	9	.012195
x1	.0008208	.0115323	1	.01149
Always				
_cons	.5884159	.0803504	0	1

```
Note: Coefficient posterior means and std. dev. estimated from 1,024 models.
Note: Default prior is used for parameter g.
Note: 5 predictors with PIP less than .01 not shown.
```

The effect of this model prior is that the posterior inclusion probability of predictors x1 and x3 through x9 is now less than 8%. There is also a slight improvement in the estimates of the intercept and the coefficient for x2.

The inclusion of prior assumptions supported by science and empirical work in a model is part of standard Bayesian analysis. With such priors, the BMA framework has the potential to provide a more reliable inference than the classical regression approach in the situations where the data have limited information about the model and its parameters.

◁

▷ Example 2: BMA for prediction compared with other approaches

In this example, we compute and compare predictions for the bmaintro dataset by using the following methods: linear regression, regress (see [R] **regress**); stepwise selection with linear regression, the stepwise prefix (see [R] **stepwise**); linear lasso variable selection, lasso linear (see [LASSO] **lasso**); and BMA linear regression, bmaregress (see [BMA] **bmaregress**).

To compare predictive performance of the models, we split our dataset into two equal samples: one for "training" the model (used for fitting) and the other for "testing" the model (used for prediction). We store the resulting sample identifier in the sample variable. And we specify a random-number seed for reproducibility.

```
. splitsample, generate(sample) nsplit(2) rseed(50)
```

Next, we fit each of the four commands using the training data, if sample == 1, and compute predictions using the test data, if sample == 2.

We start with regress to fit a linear regression and predict to obtain the linear predictor for y, which we store in the yreg variable.

```
. regress y x1-x10 if sample == 1
```

Source	SS	df	MS		Number of obs	=	100
					F(10, 89)	=	199.77
Model	2353.4317	10	235.34317		Prob > F	=	0.0000
Residual	104.84695	89	1.17805562		R-squared	=	0.9573
					Adj R-squared	=	0.9526
Total	2458.27865	99	24.8310975		Root MSE	=	1.0854

y	Coefficient	Std. err.	t	P>|t|	[95% conf. interval]	
x1	.2278093	.1115858	2.04	0.044	.0060906	.449528
x2	1.040423	.1084559	9.59	0.000	.824924	1.255923
x3	-.2557993	.1140321	-2.24	0.027	-.4823787	-.0292199
x4	-.0182061	.1175268	-0.15	0.877	-.2517293	.2153171
x5	.0389276	.1187846	0.33	0.744	-.1970948	.27495
x6	.0120724	.1107333	0.11	0.913	-.2079523	.2320971
x7	.0792028	.1378848	0.57	0.567	-.1947713	.3531768
x8	-.0841665	.1259057	-0.67	0.506	-.3343384	.1660054
x9	.0039031	.1181302	0.03	0.974	-.2308191	.2386254
x10	5.281029	.1298317	40.68	0.000	5.023056	5.539002
_cons	.5726978	.1175907	4.87	0.000	.3390475	.8063481

```
. predict yreg if sample == 2
(option xb assumed; fitted values)
(100 missing values generated)
```

Next, we use stepwise to perform stepwise backward selection with the significance level of 0.05 for the removal of a predictor from the model. And we use predict to obtain the linear predictor from the selected model and store it in the ysw variable.

```
. stepwise, pr(.05): regress y x1-x10 if sample == 1

Wald test, begin with full model:
p = 0.9737 >= 0.0500, removing x9
p = 0.9134 >= 0.0500, removing x6
p = 0.8862 >= 0.0500, removing x4
p = 0.7456 >= 0.0500, removing x5
p = 0.5099 >= 0.0500, removing x8
p = 0.5102 >= 0.0500, removing x7
```

Source	SS	df	MS		Number of obs	=	100
					F(4, 95)	=	527.09
Model	2352.28746	4	588.071866		Prob > F	=	0.0000
Residual	105.991183	95	1.11569666		R-squared	=	0.9569
					Adj R-squared	=	0.9551
Total	2458.27865	99	24.8310975		Root MSE	=	1.0563

y	Coefficient	Std. err.	t	P>|t|	[95% conf. interval]	
x1	.2143505	.1062623	2.02	0.046	.0033932	.4253078
x2	1.038816	.1028192	10.10	0.000	.8346945	1.242938
x3	-.2465552	.1087814	-2.27	0.026	-.4625137	-.0305968
x10	5.285204	.1231611	42.91	0.000	5.040698	5.52971
_cons	.5527609	.1064491	5.19	0.000	.3414327	.7640891

```
. predict ysw if sample == 2
(option xb assumed; fitted values)
(100 missing values generated)
```

We then use lasso for the linear model followed by lassocoef to see the coefficient estimates from the selected model and by predict to compute and store the penalized linear predictor in the ylasso variable.

```
. lasso linear y x1-x10 if sample == 1, rseed(18) nolog
```

Lasso linear model   No. of obs   =   100
                                                No. of covariates =   10
Selection: Cross-validation   No. of CV folds   =   10

ID	Description	lambda	No. of nonzero coef.	Out-of-sample R-squared	CV mean prediction error
1	first lambda	4.697569	0	-0.0072	24.75885
43	lambda before	.0943851	4	0.9506	1.213251
* 44	selected lambda	.0860002	4	0.9507	1.211054
45	lambda after	.0783602	4	0.9507	1.211522
48	last lambda	.0592766	5	0.9503	1.220737

* lambda selected by cross-validation.

```
. lassocoef, display(coef) nolegend
```

	active
x1	.1167579
x2	1.051272
x3	-.1659852
x10	4.53756
_cons	0

```
. predict ylasso if sample == 2
(options xb penalized assumed; linear prediction with penalized coefficients)
```

Finally, we use bmaregress to fit a BMA linear regression followed by bmapredict to compute the posterior predictive mean and store it in the ybma variable.

```
. bmaregress y x1-x10 if sample == 1

Enumerating models ...
Computing model probabilities ...

Bayesian model averaging                     No. of obs          =      100
Linear regression                            No. of predictors   =       10
Model enumeration                                    Groups      =       10
                                                     Always      =        0
Priors:                                      No. of models       =    1,024
  Models: Beta-binomial(1, 1)                    For CPMP >= .9   =       17
   Cons.: Noninformative                     Mean model size     =    2.804
   Coef.: Zellner's g
       g: Benchmark, g = 100                 Shrinkage, g/(1+g)  = 0.9901
  sigma2: Noninformative                     Mean sigma2         =    1.412
```

y	Mean	Std. dev.	Group	PIP
x10	5.18159	.1381456	10	1
x2	1.068169	.115504	2	1
x3	−.0676021	.1264303	3	.27554
x1	.0439351	.1014456	1	.20554
x8	−.0043739	.0369172	8	.05923
x9	−.0020804	.0305354	9	.054026
x7	.0022291	.0354666	7	.053837
x5	.0017863	.0301671	5	.053101
x6	.0004583	.0266441	6	.051342
x4	−.0000354	.0281472	4	.051285

Always				
_cons	.5575281	.1202808	0	1

```
Note: Coefficient posterior means and std. dev. estimated from 1,024 models.
Note: Default priors are used for models and parameter g.
. bmapredict ybma if sample == 2, mean
note: computing analytical posterior predictive means.
```

We now compute the mean squared error for each of the four predictions:

```
. generate mse_y     = (y-yreg)^2
(100 missing values generated)
. generate mse_sw    = (y-ysw)^2
(100 missing values generated)
. generate mse_lasso = (y-ylasso)^2
(100 missing values generated)
. generate mse_bma   = (y-ybma)^2
(100 missing values generated)

. summarize mse*
```

Variable	Obs	Mean	Std. dev.	Min	Max
mse_y	100	1.315471	1.544705	.0006445	8.494073
mse_sw	100	1.295875	1.57022	.0000219	8.754056
mse_lasso	100	1.246921	1.507352	.0003377	7.452369
mse_bma	100	1.174436	1.375909	.0002316	5.69697

The BMA prediction has the lowest mean squared error. Of course, a proper comparison of the techniques requires a carefully designed simulation study.

◁

> ▷ Example 3: BMA with small sample size and many predictors, $n \leq p$

Let's now consider a case when the number of observations is too small relative to the number of predictors.

```
. use https://www.stata-press.com/data/r19/bmaintrosmall, clear
(Simulated data for BMA example, small sample)

. notes list y

y:
  1.  y = .5 + 1.2*x2 + 5*x10 + rnormal()

. summarize
    Variable |        Obs        Mean    Std. dev.       Min        Max

           y |         10    .0976614    4.145433   -5.263075   6.823442
          x1 |         10   -.2640087    1.147843   -2.680089   1.069156
          x2 |         10   -.5486203    1.202882   -2.306713   1.269136
          x3 |         10    .4727975    1.193019   -.9573489   3.020441
          x4 |         10   -.0216079    .8695972    -1.82316   1.216567

          x5 |         10    .2634739    .9095448    -1.28917   1.439632
          x6 |         10     .091497     1.36508   -2.567606   1.809207
          x7 |         10    .3522653    1.033754   -1.115946   1.885638
          x8 |         10   -.1419826    .4697729   -.8331077    .6677282
          x9 |         10   -.0343085    1.213427   -2.035336   1.647427

         x10 |         10   -.0635723    .7551339   -1.023638    1.19934
```

In our toy example, we have only 10 observations, which is too small to make any reliable inferential conclusions. But we use it here for demonstration purposes to avoid dealing with too many variables. In practice, one can imagine datasets with, say, 100 observations and more than 100 potential predictors of which only a few are important in explaining the outcome, and we would like to investigate which ones. The analysis below can be easily adapted to datasets with more observations and variables.

Considering that the number of predictors in our dataset equals the sample size, we expect the traditional linear regression analysis and stepwise selection to fail. And they do.

```
. regress y x1-x10
note: x10 omitted because of collinearity.
      Source |        SS           df       MS      Number of obs   =        10
 ------------+----------------------------------   F(9, 0)         =         .
       Model |  154.661546         9  17.1846162   Prob > F        =         .
    Residual |           0         0          .    R-squared       =    1.0000
 ------------+----------------------------------   Adj R-squared   =         .
       Total |  154.661546         9  17.1846162   Root MSE        =         0

           y | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
 ------------+----------------------------------------------------------------
          x1 |  -4.475056          .         .       .           .          .
          x2 |   2.618239          .         .       .           .          .
          x3 |   -3.52965          .         .       .           .          .
          x4 |  -3.814989          .         .       .           .          .
          x5 |  -.1365321          .         .       .           .          .
          x6 |   1.262926          .         .       .           .          .
          x7 |   1.092976          .         .       .           .          .
          x8 |  -2.792013          .         .       .           .          .
          x9 |  -.7586842          .         .       .           .          .
         x10 |          0  (omitted)
       _cons |   1.051957          .         .       .           .          .
```

Because of insufficient sample size, `regress` arbitrarily omits one of the highest collinear predictors from the model because of collinearity. This happens to be one of the important predictors, `x10`. Also, as expected, `regress` fails to produce standard errors and p-values for the coefficients.

`stepwise` is not designed for $n \leq p$ and errors out.

We run `lasso linear` and compute predictions as before in example 2, except we use the same sample for fitting and prediction. Because of the small sample size, checking the out-of-sample (predictive) performance of the models is not feasible. Instead, we compare their in-sample performance, also known as model fit.

```
. lasso linear y x1-x10, rseed(18) nolog
Lasso linear model                              No. of obs        =         10
                                                No. of covariates =         10
Selection: Cross-validation                     No. of CV folds   =         10
```

			No. of nonzero	Out-of- sample	CV mean prediction
ID	Description	lambda	coef.	R-squared	error
1	first lambda	3.539901	0	-0.2057	18.64734
55	lambda before	.2871323	3	0.8906	1.692064
* 56	selected lambda	.2740817	3	0.8907	1.690632
57	lambda after	.2616242	3	0.8907	1.691032
60	last lambda	.2275474	3	0.8897	1.70528

```
* lambda selected by cross-validation.
. lassocoef, display(coef) nolegend
```

	active
x2	1.257819
x3	-.124988
x10	3.152851
_cons	0

```
. predict ylasso
(options xb penalized assumed; linear prediction with penalized coefficients)
```

The penalized coefficient of 3.15 for `x10` is not as close to the true value of 5. However, when the goal of the analysis is the optimal prediction, the actual coefficient estimates are of limited interest. And, in the context of `lasso`, it would not be appropriate to use these penalized coefficient estimates for inference anyway; see, for instance, [LASSO] **dsregress** instead.

We now fit BMA linear regression and compute predictions by using `bmaregress` and `bmapredict`, respectively.

```
. bmaregress y x1-x10

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

Bayesian model averaging			No. of obs	=	10
Linear regression			No. of predictors	=	10
Model enumeration			Groups	=	10
			Always	=	0
Priors:			No. of models	=	1,024
Models: Beta-binomial(1, 1)			For CPMP >= .9	=	47
Cons.: Noninformative			Mean model size	=	2.967
Coef.: Zellner's g					
g: Benchmark, g = 100			Shrinkage, g/(1+g)	=	0.9901
sigma2: Noninformative			Mean sigma2	=	0.916

y	Mean	Std. dev.	Group	PIP
x10	4.785368	.7709731	10	.99683
x2	1.353152	.5137089	2	.94675
x3	−.1178808	.4227608	3	.18263
x1	.0877212	.5042626	1	.17811
x6	.0642453	.2037918	6	.15993
x8	−.1180912	.7904259	8	.12465
x9	.0469446	.2233004	9	.11361
x7	−.0404475	.2257238	7	.10327
x4	−.0364019	.4581988	4	.099553
x5	−.0046	.0954065	5	.062103

Always				
_cons	1.216777	.399357	0	1

```
Note: Coefficient posterior means and std. dev. estimated from 1,024 models.
Note: Default priors are used for models and parameter g.
. bmapredict ybma, mean
note: computing analytical posterior predictive means.
```

`bmaregress` still identifies the two important predictors, but the PIP estimates are now smaller—0.997 and 0.947 for x10 and x2, respectively—compared with the values of 1 from example 2. This is expected given such a small sample size. In fact, a PIP as low as 0.5 would still qualify the predictor as important. The posterior mean estimates of the coefficients, 4.79 and 1.35, are reasonably close to their true values, 5 and 1.2, especially considering the small sample.

```
. generate mse_lasso = (y-ylasso)^2
. generate mse_bma    = (y-ybma)^2
. summarize mse*
```

Variable	Obs	Mean	Std. dev.	Min	Max
mse_lasso	10	.5736865	.7505244	.007729	2.541895
mse_bma	10	.3219035	.3677712	.0037272	1.186045

According to the smaller mean squared error, BMA produces predictions that are closer to the observed values than lasso in this example. And, unlike lasso, BMA can produce credible intervals for the predictions; see [BMA] **bmapredict**.

◁

It is difficult to generalize the conclusions based on these simple examples to other more complex situations, because we only looked at one dataset and one realization of the DGM. A proper simulation study is needed to make more general conclusions. But our limited findings appear to agree with some of the results reported in the literature.

Brief background and literature review

The initial development of the concept of model averaging was driven by the application problems, which have not been considered by mainstream statisticians. Barnard (1963) was one of the first to use a combination of models. His research was in quality-control methods with application to airline data. An early work by Bates and Granger (1969) introduced the idea of model combinations to problems of forecasting and influenced a string of follow-up articles, such as Newbold and Granger (1974) and Winkler and Makridakis (1983). During the 1970s, the development of model averaging took place mostly in economics research.

In statistical research, model averaging was also motivated by problems of prediction. Roberts (1965) viewed marginal distributions, either prior or posterior, as predictive distributions suitable to answer questions about model selection, interpretation, and validation. He suggested combining two models based on two different elicited priors. His idea was generalized by Leamer (1978), who was particularly interested in the uncertainty involved in model selection. Despite this early work, it took another two decades of the theoretical work for BMA to become a principled statistical method (Draper 1995; Kass and Wasserman 1995; and George 2014). Meanwhile, the developments in Bayesian computation, such as MCMC sampling methods, allowed researchers to effectively apply BMA in practice (Madigan and York 1995; Raftery 1996; Raftery, Madigan, and Volinsky 1995; and Hoeting et al. 1999). Madigan and Raftery (1994) showed the optimal predictive performance of BMA for high-dimensional contingency tables in comparison with model-selection methods. Clyde (1999) investigated prior specification and model search strategies in BMA.

The popularity of BMA in various scientific disciplines grew substantially. Fragoso, Bertoli, and Louzada (2018) provide a systematic review of published articles from 1996 to 2014. An in-depth survey of model-averaging application to problems of ecology is presented in Dormann et al. (2018). For application of BMA in political science, see Adams, Bishin, and Dow (2004) and Montgomery and Nyhan (2010).

The use of model-averaging methods in economic research remains strong. The application of BMA to problems of empirical microeconomics, with emphasis on big-data problems, is discussed in Koop (2017). A general overview of the use of model averaging in economics is given by Steel (2020). Among the questions in economic research, BMA has been traditionally applied to determining the growth factors driving economic processes (Brock and Durlauf 2001; Fernández, Ley, and Steel 2001b; Lenkoski, Eicher, and Raftery 2014; and Eicher and Newiak 2013). BMA is also a popular approach in policy and decision-making evaluation (Brock, Durlauf, and West 2003). The benefit of BMA as a tool for dealing with uncertainty in economic research is well documented in Marinacci (2015).

A survey of statistical methods accounting for model uncertainty demonstrates the advantage of BMA over other popular model-selection methodologies (Porwal and Raftery 2022). For comparison of BMA with other predictive methodologies, see Yao et al. (2018) and Piironen and Vehtari (2017).

References

Adams, J., B. G. Bishin, and J. K. Dow. 2004. Representation in congressional campaigns: Evidence for discounting/directional voting in U.S. Senate elections. *Journal of Politics* 66: 348–373. https://doi.org/10.1111/j.1468-2508.2004.00155.x.

Arin, K. P., and E. Braunfels. 2018. The resource curse revisited: A Bayesian model averaging approach. *Energy Economics* 70: 170–178. https://doi.org/10.1016/j.eneco.2017.12.033.

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.

Barnard, G. A. 1963. New methods of quality control. *Journal of the Royal Statistical Society*, A ser., 126: 255–258. https://doi.org/10.2307/2982365.

Bates, J. M., and C. W. J. Granger. 1969. The combination of forecasts. *Operational Research* 20: 451–468. https://doi.org/10.2307/3008764.

Breiman, L. 1996. Bagging predictors. *Machine Learning* 24: 123–140. https://doi.org/10.1007/BF00058655.

Brock, W. A., and S. N. Durlauf. 2001. What have we learned from a decade of empirical research on growth? Growth empirics and reality. *World Bank Economic Review* 15: 229–272. https://doi.org/10.1093/wber/15.2.229.

Brock, W. A., S. N. Durlauf, and K. D. West. 2003. Policy evaluation in uncertain economic environments. *Brookings Papers on Economic Activity* 1: 235–322. https://doi.org/10.1353/eca.2003.0013.

Buckland, S. T., K. P. Burnham, and N. H. Augustin. 1997. Model selection: An integral part of inference. *Biometrics* 53: 603–618. https://doi.org/10.2307/2533961.

Chatfield, C. 1995. Model uncertainty, data mining and statistical inference. *Journal of the Royal Statistical Society*, A ser., 158: 419–466. https://doi.org/10.2307/2983440.

Clyde, M. A. 1999. "Bayesian model averaging and model search strategies". In *Bayesian Statistics: Proceedings of the Sixth Valencia International Meeting*, edited by J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, vol. 6: 157–185. Oxford: Clarendon Press.

Darwen, P. J. 2019. Bayesian model averaging for river flow prediction. *Applied Intelligence* 49: 103–111. https://doi.org/10.1007/s10489-018-1232-0.

De Luca, G., and J. R. Magnus. 2011. Bayesian model averaging and weighted-average least squares: Equivariance, stability, and numerical issues. *Stata Journal* 11: 518–544.

Dormann, C. F., J. M. Calabrese, G. Guillera-Arroita, E. Matechou, V. Bahn, K. Bartoń, C. M. Beale, S. Ciuti, J. Elith, K. Gerstner, J. Guelat, P. Keil, J. J. Lahoz-Monfort, L. J. Pollock, B. Reineking, D. R. Roberts, B. Schröder, W. Thuiller, D. I. Warton, B. A. Wintle, S. N. Wood, R. O. Wüest, and F. Hartig. 2018. Model averaging in ecology: A review of Bayesian, information-theoretic, and tactical approaches for predictive inference. *Ecological Monographs* 88: 485–504. https://doi.org/10.1002/ecm.1309.

Draper, D. 1995. Assessment and propagation of model uncertainty. *Journal of the Royal Statistical Society*, B ser., 57: 45–70. https://doi.org/10.1111/j.2517-6161.1995.tb02015.x.

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

Eicher, T. S., and M. Newiak. 2013. Intellectual property rights as development determinants. *Canadian Journal of Economics* 46: 4–22. https://doi.org/10.1111/caje.12000.

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.

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.

George, E. I. 2014. "Bayesian model selection". In *Wiley StatsRef: Statistics Reference Online*, edited by N. Balakrishnan, T. Colton, B. Everitt, W. Piegorsch, F. Ruggeri, and J. L. Teugels. New York: Wiley. https://doi.org/10.1002/9781118445112.stat00228.

Good, I. J. 1952. Rational decisions. *Journal of the Royal Statistical Society*, B ser., 14: 107–114. https://doi.org/10.1111/j.2517-6161.1952.tb00104.x.

Hansen, B. E. 2007. Least squares model averaging. *Econometrica* 75: 1175–1189. https://doi.org/10.1111/j.1468-0262.2007.00785.x.

Hansen, B. E., and J. S. Racine. 2012. Jackknife model averaging. *Journal of Econometrics* 167: 38–46. https://doi.org/10.1016/j.jeconom.2011.06.019.

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.

Kaplan, D., and C. Lee. 2018. Optimizing prediction using Bayesian model averaging: Examples using large-scale educational assessments. *Evaluation Review* 42: 423–457. https://doi.org/10.1177/0193841X18761421.

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.

Koop, G. 2003. *Bayesian Econometrics.* Chichester, UK: Wiley.

———. 2017. Bayesian methods for empirical macroeconomics with big data. *Review of Economic Analysis* 9: 33–56. https://doi.org/10.15353/rea.v9i1.1434.

Leamer, E. E. 1978. *Specification Searches: Ad Hoc Inference with Nonexperimental Data.* New York: Wiley.

Lenkoski, A., T. S. Eicher, and A. E. Raftery. 2014. Two-stage Bayesian model averaging in endogenous variable models. *Econometric Reviews* 33: 122–151. https://doi.org/10.1080/07474938.2013.807150.

Ley, E., and M. F. J. Steel. 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.

Madigan, D., J. Gavrin, and A. E. Raftery. 1995. Eliciting prior information to enhance the predictive performance of Bayesian graphical models. *Communications in Statistics—Theory and Methods* 24: 2271–2292. https://doi.org/10.1080/03610929508831616.

Madigan, D., and A. E. Raftery. 1994. Model selection and accounting for model uncertainty in graphical models using Occam's window. *Journal of the American Statistical Association* 89: 1535–1546. https://doi.org/10.2307/2291017.

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.

Marinacci, M. 2015. Model uncertainty. *Journal of the European Economic Association* 13: 1022–1100. https://doi.org/10.1111/jeea.12164.

Min, C., and A. Zellner. 1993. Bayesian and non-Bayesian methods for combining models and forecasts with applications to forecasting international growth rates. *Journal of Econometrics* 56: 89–118. https://doi.org/10.1016/0304-4076(93)90102-B.

Montgomery, J. M., and B. Nyhan. 2010. Bayesian model averaging: Theoretical developments and practical applications. *Political Analysis* 18: 245–270. https://doi.org/10.1093/pan/mpq001.

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

Newbold, P., and C. W. J. Granger. 1974. Experience with forecasting univariate time series and the combination of forecasts. *Journal of the Royal Statistical Society*, A ser., 137: 131–165. https://doi.org/10.2307/2344546.

Peisker, J. 2023. Context matters: The drivers of environmental concern in European regions. *Global Environmental Change* 79: 102636. https://doi.org/10.1016/j.gloenvcha.2023.102636.

Piironen, J., and A. Vehtari. 2017. Comparison of Bayesian predictive methods for model selection. *Statistics and Computing* 27: 711–735. https://doi.org/10.1007/s11222-016-9649-y.

Porwal, A., and A. E. Raftery. 2022. Comparing methods for statistical inference with model uncertainty. *PNAS* 119(16): e2120737119. https://doi.org/10.1073/pnas.2120737119.

Raftery, A. E. 1996. "Hypothesis testing and model selection". In *Markov Chain Monte Carlo in Practice*, edited by W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, 163–187. Boca Raton, FL: Chapman and Hall.

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.

Raftery, A. E., D. Madigan, and C. T. Volinsky. 1995. "Accounting for model uncertainty in survival analysis improves predictive performance". In *Bayesian Statistics: Proceedings of the Fifth Valencia International Meeting, June 5–9, 1994*, edited by J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, vol. 5: 323–349. Oxford: Clarendon Press.

Raftery, A. E., and Y. Zheng. 2003. Discussion: Performance of Bayesian model averaging. *Journal of the American Statistical Association* 98: 931–938. https://doi.org/10.1198/016214503000000891.

Roberts, H. V. 1965. Probabilistic prediction. *Journal of the American Statistical Association* 60: 50–62. https://doi.org/10.2307/2283136.

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.

Tobias, J. L., and M. Li. 2004. Returns to schooling and Bayesian model averaging: A union of two literatures. *Journal of Economic Surveys* 18: 153–180. https://doi.org/10.1111/j.0950-0804.2004.00003.x.

Volinsky, C. T., D. Madigan, A. E. Raftery, and R. A. Kronmal. 1997. Bayesian model averaging in proportional hazards models: Assessing the risk of a stroke. *Journal of the Royal Statistical Society*, C ser., 46: 433–448. https://doi.org/10.1111/1467-9876.00082.

Wasserman, L. 2000. Bayesian model selection and model averaging. *Journal of Mathematical Psychology* 44: 92–107. https://doi.org/10.1006/jmps.1999.1278.

Winkler, R. L., and S. Makridakis. 1983. The combination for forecats. *Journal of the Royal Statistical Society*, A ser., 146: 150–157. https://doi.org/10.2307/2982011.

Wolpert, D. H. 1992. Stacked generalization. *Neural Networks* 5: 241–259. https://doi.org/10.1016/S0893-6080(05)80023-1.

Yao, Y., A. Vehtari, D. Simpson, and A. Gelman. 2018. Using stacking to average Bayesian predictive distributions (with dicscussion). *Bayesian Analysis* 13: 917–1007. https://doi.org/10.1214/17-BA10914.

Yin, G., and Y. Yuan. 2009. Bayesian model averaging continual reassessment method in phase I clinical trials. *Journal of the American Statistical Association* 104: 954–968. https://doi.org/10.1198/jasa.2009.ap08425.

Also see

BMA commands	— Introduction to commands for Bayesian model averaging

Description

This entry describes a suite of commands to perform Bayesian model averaging (BMA). BMA is a statistical procedure that performs inference and computes predictions by combining results from multiple models according to Bayesian principles. It accounts for model uncertainty and thus can provide more reliable inference and prediction than traditional methods that choose one model. See *Brief motivation* in *Remarks and examples* of [BMA] **Intro** for a quick overview of BMA.

Setup

splitsample	split data into random samples for training, validation, and prediction
vl	manage large variable lists conveniently

Estimation

bmaregress	BMA linear regression
bmacoefsample	Posterior samples of regression coefficients

Graphical commands

bmagraph	Graphical summaries
bmagraph pmp	Model-probability plots
bmagraph varmap	Variable-inclusion maps
bmagraph msize	Model-size distribution plots
bmagraph coefdensity	Coefficient density plots

Postestimation statistics

bmastats	Posterior summaries
bmastats msize	Model-size summary
bmastats models	Posterior model and variable-inclusion summaries
bmastats pip	Posterior inclusion probabilities for predictors
bmastats jointness	Jointness measures for predictors
bmastats lps	Log predictive-score

Predictions

bmapredict	BMA predictions

Remarks and examples

Here we provide a brief overview of the BMA commands implemented in Stata. See [BMA] **Intro** for an introduction to the BMA methodology and *Usage of BMA*, in particular, for various applications of BMA.

The `bmaregress` estimation command implements BMA for linear regression. In the regression context, the BMA model space consists of 2^p regression models formed by all possible combinations of the inclusion or exclusion of each of the p predictors. The model is considered a discrete random parameter with a prior distribution over the model space. You can choose from a variety of prior model distributions. The model parameters—the regression coefficients, intercept, and error variance—are also random, as they are in standard Bayesian analysis. The intercept and error variance are assumed to have noninformative priors, and the regression coefficients are assumed to have a Zellner's g-prior. The g parameter controls the shrinkage of coefficients toward zero and can be fixed or random. `bmaregress` supports many fixed choices and hyperpriors for this parameter.

`bmaregress` explores the model space either by enumerating all possible regression models or by sampling from the posterior model distribution. The sampling is based on specialized Markov chain Monte Carlo (MCMC) algorithms. When sampling is used, you must check convergence of the results before continuing with inference and prediction. This can be done visually with the `bmagraph pmp` command; [BMA] **bmagraph pmp**.

`bmaregress` also offers various modeling options such as always including certain predictors and keeping groups of predictors together in the models. It supports factor variables and offers several ways of handling main effects and interactions during model building. See [BMA] **bmaregress** for details about the command.

`bmaregress` reports posterior summaries (over the model space) for regression coefficients as well as posterior inclusion probabilities (PIPs) for the predictors. The PIP measures the importance of a predictor in explaining the outcome over the model space. You can use `bmastats pip` to report PIPs for a subset of predictors; [BMA] **bmastats pip**.

After `bmaregress`, you can use postestimation commands described in [BMA] **bmastats** and [BMA] **bmagraph** for inference about models and predictors.

`bmastats models` summarizes models with high posterior model probabilities, PMPs, and the predictors they include; see [BMA] **bmastats models**. `bmagraph pmp` and `bmagraph varmap` are its graphical counterparts; see [BMA] **bmagraph pmp** and [BMA] **bmagraph varmap**.

`bmastats msize` and `bmagraph msize` provide the descriptive and graphical summaries, respectively, for the posterior model-size distribution. These are useful to explore the complexity of the BMA model; see [BMA] **bmastats msize** and [BMA] **bmagraph msize**.

The concept of jointness is of particular interest in BMA. It describes the tendency of a pair of variables to be included in the models together, separately, or independently. You can use the `bmastats jointness` command to compute various measures of jointness; see [BMA] **bmastats jointness**.

The `bmaregress` command computes and reports the posterior means and standard deviations for regression coefficients. To compute other posterior summaries such as credible intervals, you need a posterior sample of regression coefficients. You can use the `bmacoefsample` command to generate this sample of coefficients (and other model parameters); see [BMA] **bmacoefsample**. Once the sample is available, many standard Bayesian postestimation commands can be used to summarize this sample such as `bayesstats summary`. See [BMA] **BMA postestimation** for the full list.

The posterior distribution of the regression coefficients can be visualized by `bmagraph coefdensity`. With a fixed g, you can use this command directly after `bmaregress`. With a random g, an MCMC sample of model parameters must be simulated first by using `bmacoefsample`. See [BMA] **bmagraph coefdensity** for details.

Another important application of BMA is prediction. You can use bmapredict to compute various predictions after bmaregress; see [BMA] **bmapredict**. As with some other postestimation commands, certain predictions rely on an MCMC sample of model parameters. For those, you need to run bmacoefsample first. If the sample is available, you can also use the bayespredict command to compute more complicated predictions; [BAYES] **bayespredict**.

Finally, you can use the bmastats lps command to compare model fit and predictive performance of BMA models by using the log predictive-score (LPS). The fit of models is typically compared by computing LPS for the estimation sample. Predictive performance is evaluated by computing LPS for the out-of-sample observations, which were not used during estimation. See [BMA] **bmastats lps** for details.

For examples of BMA commands, see the following.

Getting started. *Motivating examples* in *Remarks and examples* of [BMA] **Intro** provides a quick introduction to and motivation for the use of bmaregress. *Getting started examples* in *Remarks and examples* of [BMA] **bmaregress** provides a tour of various BMA analysis in Stata.

Model choice and inference. *BMA analysis of cross-country economic growth data* in *Remarks and examples* of [BMA] **bmaregress** demonstrates the use of BMA for inference and model choice, including the investigation of jointness of predictors.

Prediction. The use of BMA for prediction is described in *BMA predictive performance for the USA crime rate data* in *Remarks and examples* of [BMA] **bmaregress** and example 1 of [BMA] **bmapredict**. The evaluation of a model's predictive performance using the LPS is demonstrated in *Remarks and examples* in [BMA] **bmastats lps** with the application of predicting the systolic blood pressure.

Acknowledgments

Reference

De Luca, G., and J. R. Magnus. 2011. Bayesian model averaging and weighted-average least squares: Equivariance, stability, and numerical issues. *Stata Journal* 11: 518–544.

Also see

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

[BMA] **Glossary**

[BAYES] **Bayesian commands** — Introduction to commands for Bayesian analysis

---

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, *varname*, or one of in–out groups of predictors, (*varlist*). *varname* is always its own group; that is, (*varname*) 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)
<u>gprior</u>(*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>thinning</u>(*#*)	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>clevel</u>(*#*)	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>inputorder</u>	display results in input order; default is descending order of PIP
<u>recentered</u>	show coefficients for recentered predictors
pipcutoff(*#*)	lowest PIP value for displaying results; default is pipcutoff(.01)
<u>notable</u>	suppress estimation table
<u>nohead</u>er	suppress output header
<u>title</u>(*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
betabinomial	same as betabinomial 1 1; the default
betabinomial $\#_1$ $\#_2$	beta-binomial prior using shape parameters of beta distribution
betabinomial #	beta-binomial prior using expected model size #
uniform	uniform prior
binomial #	binomial prior with the same inclusion probability # for all in−out terms
binomial	same as binomial 0.5
binomial *inoutterm* # [...]	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* # [...].

　　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.

Reporting

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 need 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      Storage   Display    Value
    name         type    format    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 ...
```


Variable-inclusion map
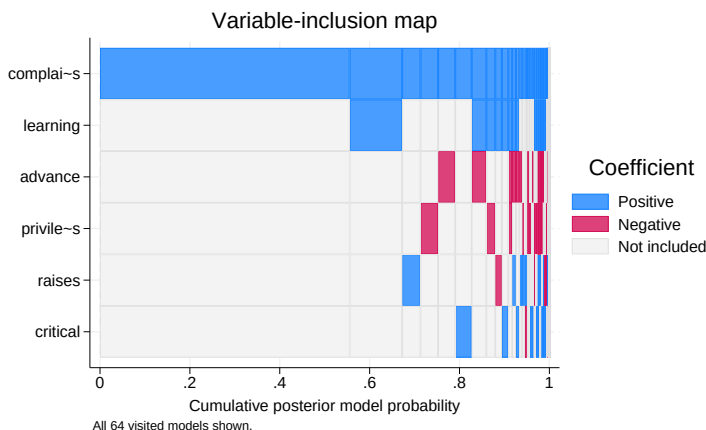
By default, the first top 100 models are plotted. In our example, this means that all 64 models are shown on the graph. We can see that the top model contains only `complaints` with a PMP or, equivalently, CPMP of roughly 0.56 (the width of the first bar). The next highest-PMP model contains both `complaints` 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.
```
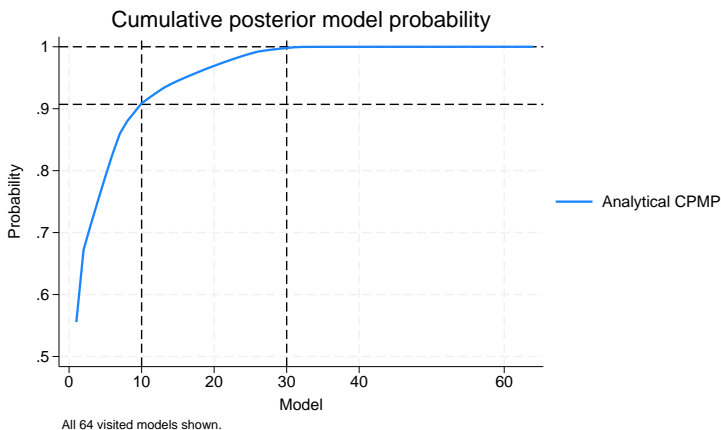

Again, the default top 100 models and thus all 64 models in our example are shown on the plot. We can see that the CPMP of about 0.9 is reached with 10 models and of 1 with roughly 30 models, so the rest of the visited models have essentially 0 PMPs and do not contribute to the averaged results.

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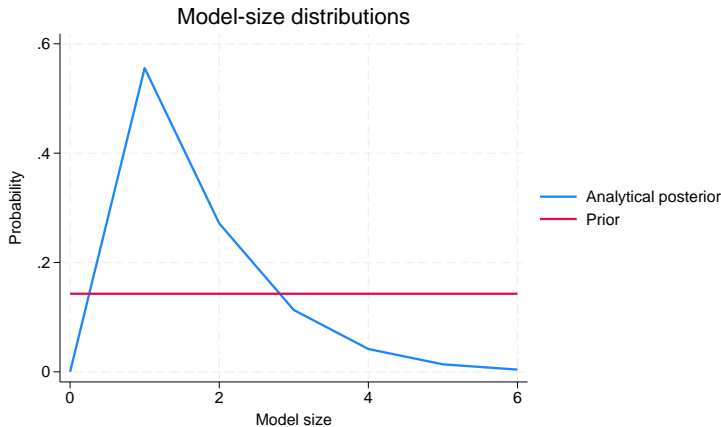
The default beta-binomial prior is uniform over the model size. Compared with the prior, the posterior prior is skewed to the left, favoring models with fewer predictors. In fact, the mode of the distribution corresponds to the model with only one predictor, which we know is complaints. Thus, the data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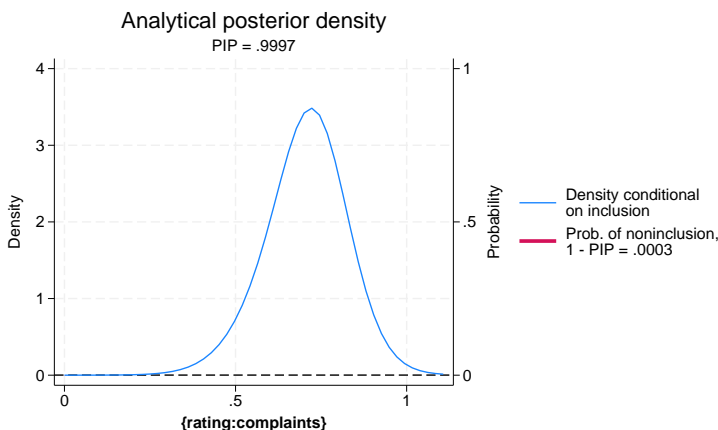

The posterior density of a regression coefficient is a mixture of a point mass at zero, which represents the probability of not being included in the model and a continuous density conditional on being included. For coefficient {complaints} (or using its full name {rating:complaints}), the probability 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}
```
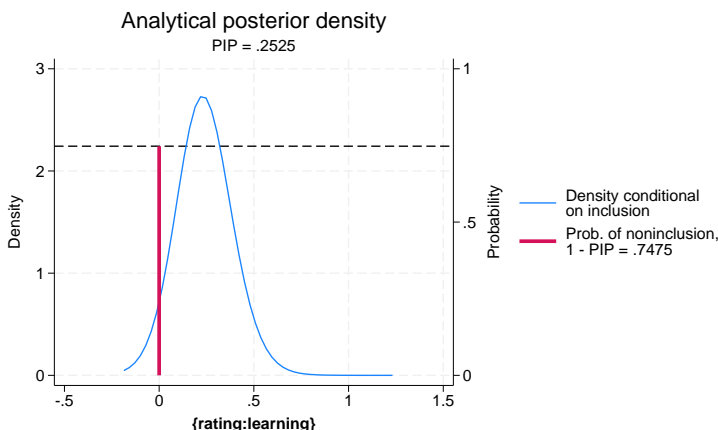

The probability of inclusion for {learning} is about 0.25, so we see the red vertical line with the horizontal reference line at a noninclusion probability of roughly $1 - 0.25 = 0.75$. Conditional on the inclusion, the continuous density has its mass between roughly -0.2 and 0.7, which includes zero.

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

	Mean	Std. dev.	MCSE	Median	Equal-tailed [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 regression 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
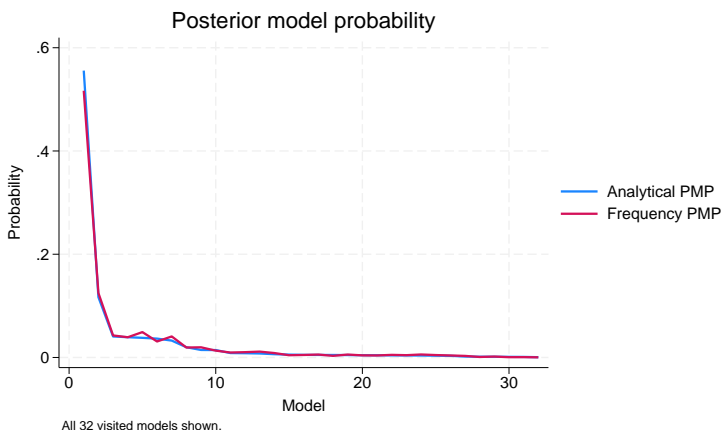
We can also explore the CPMPs by specifying the `cumulative` option. The analytical and sampling cumulative probabilities overlap closely too.

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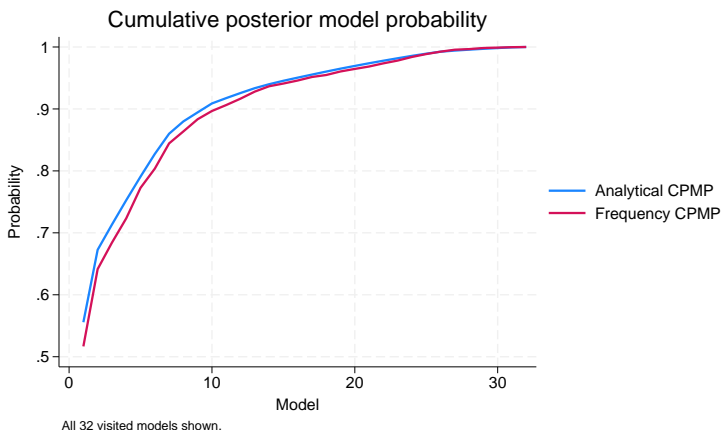
The cumulative graph also shows the contributions of the top PMP models. In our example, the top 6 models contribute to about 80% of the total probability mass on the model space.

◁

▷ 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.
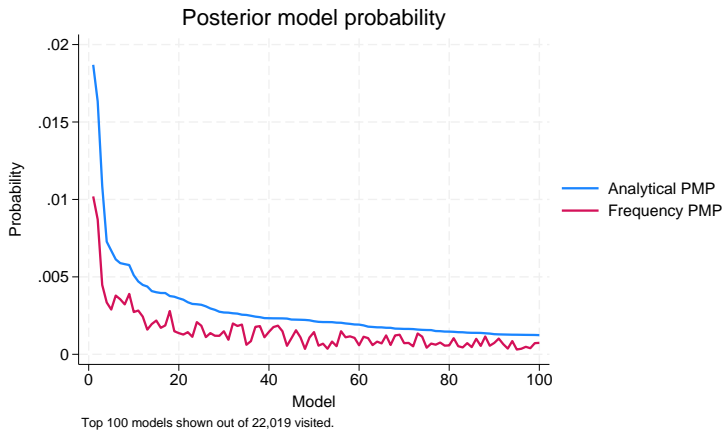
For the first 100 models, the frequency PMPs are below the analytical PMPs because we explored only a small fraction of all possible models, and the PMP distribution has a long and heavy tail. With the increase of the MCMC sample size, the differences will diminish. We do not have a reason to suspect 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 mini-
mum 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
```
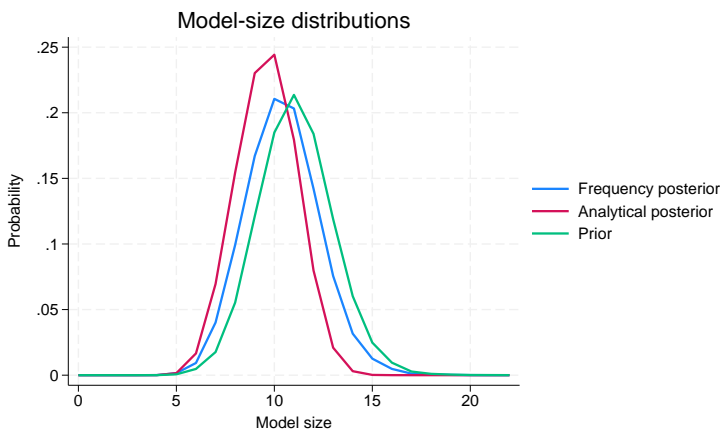

The model-size distributions are fairly similar with the posterior one shifted to the left, favoring slightly smaller models than what was assumed a priori.

◁

▷ Example 19: Coefficient summaries

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

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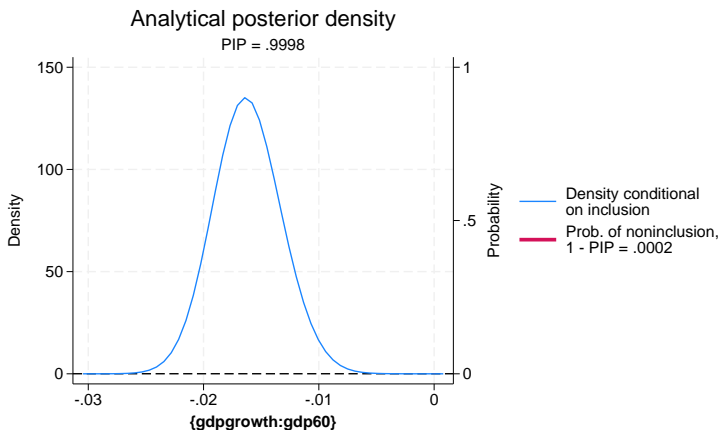
For coefficient {gdp60} (or using its full name {gdpgrowth:gdp60}), the probability of noninclusion is very low, 0.0002, so the red line that represents it is not even visible on the graph. Thus the posterior density of {gdp60} is essentially a continuous normal-like density, centered at about -0.015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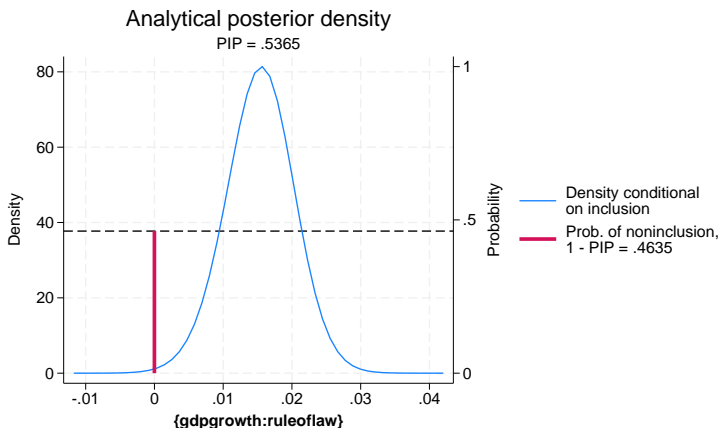

The probability of noninclusion for {ruleoflaw} is about 0.46, so we see the red vertical line meets the horizontal reference line at roughly 0.46. Conditional on the inclusion, the continuous density is centered around 0.015 with most of its mass between 0 and 0.03.

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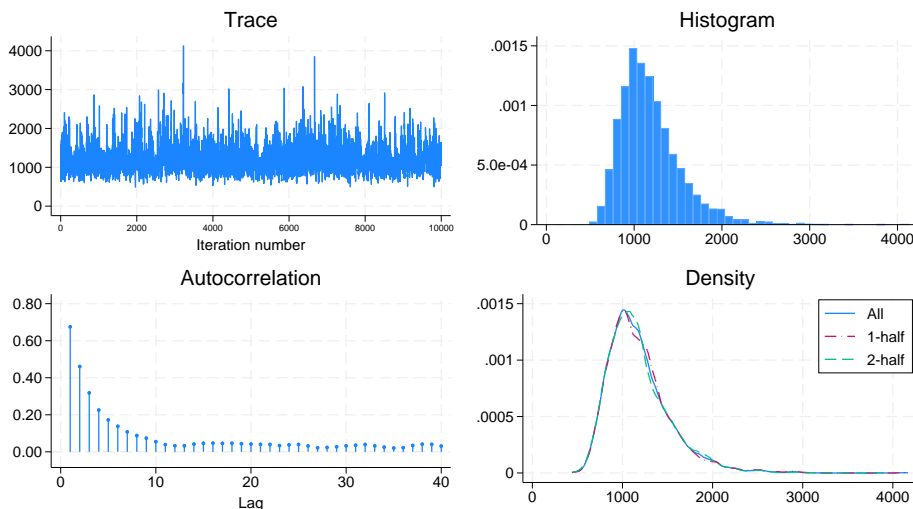
So the convergence for g does not imply convergence over the model space. To improve the latter, we specify a larger initial value for g, 1,000, which is closer to the prior mean of g, and increase the MCMC sample size to 40,000.

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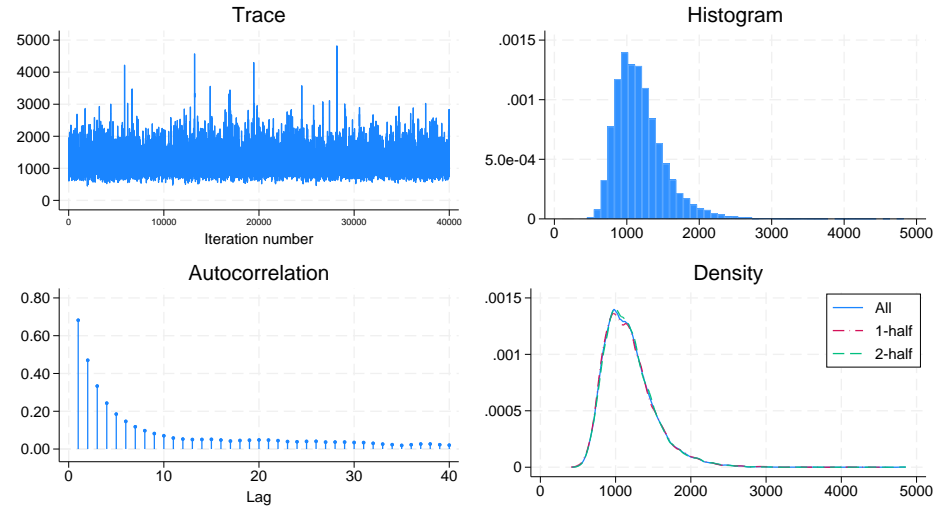
We obtain results similar to those for a fixed g in example 17. The estimated shrinkage posterior mean of 0.999 matches closely the fixed value of 0.994. Thus, the chosen prior for g does not seem to have much of an effect on the results.

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 enumera-
                                tion
  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 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 $p_{inc,k}$ be the prior probability of inclusion for the kth predictor. Assume that the inclusion of each predictor is independent; then the binomial prior specifies

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

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

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

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

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

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

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

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

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

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

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

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 γ.

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

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

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

3. Jump to the model γ^* with probability

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

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

Random g parameter

In this case, both g and γ are random, and we construct a Markov chain based on the following conditional statements:

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

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

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

For sampling the model γ, 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 $\alpha_{\mathrm{opt}} = 0.44$ (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 $\mathrm{LN}(\mu, \sigma)$ be the lognormal distribution with mean μ and standard deviation σ. Its density is denoted by $f_{\mathrm{LN}}(\cdot; \mu, \sigma)$.

Modified MC3 sampler for (g, γ).

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

Let $\{g^{(t)}, \gamma^{(t)}\}$ be the current state of the chain at iteration t, where $\gamma^{(t)} = (\gamma_1^t, \gamma_2^t, \ldots, \gamma_p^t)$. 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 $\{g^{(0)}, \gamma^{(0)}\}$ 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 $g^* \sim \text{LN}(g^{(t)}, \sigma^{(t)})$.

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

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

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

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

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,

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

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

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

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

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

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

Description

bmacoefsample simulates regression coefficients and other model parameters from their posterior distributions after bmaregress.

Quick start

Simulate regression coefficients and other model parameters from their posterior distributions after bmaregress, and use bayesstats summary to display posterior summaries for {y:}, all 12 regression coefficients, and the intercept, including their 90% highest posterior density (HPD) credible intervals (CrIs)

```
bmaregress y x1-x12, saving(bmamodelmcmc)
bmacoefsample
bayesstats summary {y:}, clevel(90) hpd
```

Same as above, but specify a random-number seed and the size of the simulated dataset, and save the simulated dataset in file bmacoefmcmc.dta

```
bmacoefsample, rseed(1234) mcmcsize(5000) saving(bmacoefmcmc)
```

Simulate results without saving first, but then later specify option saving() to save the previously simulated results in file bmacoefmcmc1.dta, and save a new set of results in bmacoefmcmc2.dta

```
bmacoefsample, rseed(1234)
bmacoefsample, saving(bmacoefmcmc1)
bmacoefsample, saving(bmacoefmcmc2)
```

Same as above, but save only one set of results, use option simulate to resimulate the results, and save the new ones in bmacoefmcmc1, replacing already existing file

```
bmacoefsample, rseed(1234)
bmacoefsample, saving(bmacoefmcmc1, replace) simulate
```

Menu

Statistics > Bayesian model averaging > Posterior samples of coefficients

Syntax

> bmacoefsample [, *coefsimopts*]

coefsimopts	Description
Simulation	
<u>mcmcsize</u>(#)	Markov chain Monte Carlo (MCMC) sample size; default is mcmcsize(10000) or as specified with bmaregress
rseed(#)	random-number seed
Reporting	
<u>saving</u>(*filename*[, replace])	save simulation results to *filename*.dta
[no]dots	suppress or display dots every 1,000 iterations and iteration numbers every 5,000 iterations; default is dots
dots(#[, every(#)])	display dots as simulation is performed
<u>simu</u>late	simulate new results instead of saving existing ones

simulate does not appear in the dialog box.

Options

> ┌─── Simulation ───

mcmcsize(#) specifies the target MCMC sample size. The default is mcmcsize(10000) or as specified in bmaregress's mcmcsize() option.

rseed(#) sets the random-number seed. This option can be used to reproduce results. Option rseed(#) is equivalent to typing set seed # prior to calling the bmacoefsample command; see [R] **set seed**.

> ┌─── Reporting ──

saving(*filename*[, replace]) saves simulation results in *filename*.dta. The replace option specifies to overwrite *filename*.dta if it exists. If the saving() option is not specified, bmacoefsample saves simulation results in a temporary file for later access. This temporary file will be overwritten every time bmacoefsample or bmaregress is run and will be erased if the current estimation results are cleared. The simulation dataset has the same structure as described in option saving() of [BAYES] **bayesmh**. The simulation results include posterior samples of regression coefficients, the intercept, and the error variance.

If you first run bmacoefsample without option saving(), you can rerun it with this option later to save the current simulation results in a file without having to redo a potentially time-consuming simulation. If, for some reason, you need to simulate a new sample in this case, you should specify option simulate in addition to saving().

dots, nodots, dots(#), and dots(#, every(#)) specify to display or suppress dots during simulation. dots, the default, displays dots every 1,000 iterations and iteration numbers every 5,000 iterations; it is a synonym for dots(1000, every(5000)). dots(#) displays a dot every # iterations. If dots(..., every(#)) is specified, then an iteration number is displayed every #th iteration instead of a dot. dots(, every(#)) is equivalent to dots(1, every(#)).

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

`simulate` is used with option `saving()` to simulate a new set of results and save them in a file instead of saving the previously simulated results. This option is implied with `bmacoefsample` in all but one case—when option `saving()` is used and simulation results from a previous run of `bmacoefsample` have not been saved in a file already.

Remarks and examples

Bayesian model averaging (BMA) simulation consists of two steps: 1) simulation of a model space and, optionally, 2) simulation of model parameters, which include regression coefficients, the intercept, and the error variance. The first, main step is performed by `bmaregress`, during which a posterior sample of models is obtained. In addition, a posterior sample of parameter g of a Zellner's g-prior, which affects the simulation of the model space, is obtained. Because `bmaregress` reports only posterior means and standard deviations of model parameters, which are available analytically, it does not automatically simulate posterior samples of model parameters to save time. But these samples may be needed if we want to obtain CrIs, functions of regression coefficients, or certain predictions.

The `bmacoefsample` command simulates model parameters of a BMA linear regression fit by `bmaregress` from their posterior distributions. The simulated data include posterior samples of regression coefficients, the intercept, and the error variance. For convenience, the data also include the posterior sample of the g parameter simulated by `bmaregress`. To use `bmacoefsample` after `bmaregress`, you must first save the BMA model simulation results by using the `saving()` option with `bmaregress` either during estimation or on replay.

`bmacoefsample` differs from a standard Stata postestimation command—it is a hybrid between an estimation and a postestimation command. Unlike other Stata postestimation commands, `bmacoefsample` actually modifies or, rather, augments the estimation results stored by `bmaregress`. But it is still a postestimation command because it can run only after `bmaregress` is run. You can think of it as a "replay option" of the `bmaregress` command, which continues the simulation process started by `bmaregress`. But it is more convenient as a standalone command.

In addition to storing results in `e()`, `bmacoefsample` has its own concept of "replay". Because it performs a potentially time-consuming simulation, it allows you to save your simulation results later, if you did not do so the first time you ran the command. That is,

```
. bmaregress ..., saving(bmamodelmcmc)
. bmacoefsample
```

produces a simulation sample of model parameters but saves them temporarily. If you run `bmacoefsample` or `bmaregress` again, these results will be lost. You can use `bmacoefsample`'s `saving()` option to save the current model parameter simulation results permanently in a file:

```
. bmacoefsample, saving(mycoef)
```

Once the current results are permanently saved, the next run of `bmacoefsample` will automatically produce a new sample. In fact, if you use any option with `bmacoefsample` other than `saving()`, it will automatically trigger the generation of a new sample. You can also force the `bmacoefsample` command to resample and save a new set of simulation results by specifying the `simulate` option together with `saving()`:

```
. bmacoefsample
. bmacoefsample, saving(mycoef) simulate
```

For a fixed g, the samples generated by bmaregress and bmacoefsample are independent. For a random g, these MCMC samples are dependent.

Inference based on MCMC simulations depends on the variation of the MCMC sample. Specifically, the size of the MCMC sample, that is the number of draws from parameters' posterior distributions, affects the precision of the parameter estimates. The larger the MCMC sample size, the more precise the estimates. The MCMC sample size is specified in the mcmcsize() option. The default is 10,000 or the value specified in bmaregress's mcmcsize() option.

bmacoefsample is a stochastic command, so you need to specify the rseed() option to reproduce results.

After bmacoefsample, you can use the following standard Bayesian postestimation commands to explore posterior summaries of model parameters: [BAYES] **bayesstats summary**, [BAYES] **bayesstats ess**, [BAYES] **bayesstats ppvalues**, [BAYES] **bayesgraph**, and [BAYES] **bayespredict**. You can also use the estimates command, except estimates table, estimates stats, and estimates selected; see [BMA] **BMA postestimation**.

For examples of how to use this command, see, for instance, example 5, example 16, and example 19 of [BMA] **bmaregress**.

Stored results

In addition to the estimation results stored by bmaregress, bmacoefsample stores the following in e():

Scalars
   e(mcmcsize2)   MCMC sample size

Macros
   e(cmd2)   bmacoefsample
   e(filename)   name of the file with simulation results
   e(parnames)   names of model parameters
   e(postvars)   variable names corresponding to model parameters in e(parnames)
   e(scparams)   scalar model parameters
   e(pareqmap)   model parameters in display order
   e(rngstate2)   random-number state at the time of simulation

Methods and formulas

Methods and formulas are presented under the following headings:

        Enumerated model space
        Simulated model space

Conditional on a model M and parameter g, the posterior distribution of the error variance σ^2 is an inverse-gamma distribution, and the posterior distribution of regression coefficients and the intercept is a multivariate location-scale t-distribution; see *Conditional posterior distribution of model parameters* in *Methods and formulas* of [BMA] **bmaregress**.

bmacoefsample simulates model parameters from their posterior distributions in two ways, which depend on whether bmaregress enumerated or sampled the model space.

Enumerated model space

When all models in the model space are visited by `bmaregress`, the posterior model distribution is known, and PMPs can be computed exactly. Also, Zellner's g parameter is fixed with model enumeration. In this case, `bmacoefsample` generates an independent sample of size T for model parameters as follows.

For $t = 1, 2, \ldots, T$:

1. Draw a model M_t from the model space according to the PMP distribution, which is described in *Posterior model probability* in *Methods and formulas* of [BMA] **bmaregress**.

2. Draw model parameters according to

$$\sigma_t^2 | g, M_t \;\sim\; \text{InverseGamma}\left(\frac{n-1}{2}, \frac{s_{t,\delta}^2}{2}\right)$$

$$\boldsymbol{\theta}_{t,p_t} | \sigma_t^2, g, M_t \;\sim\; N_{p_t+1}(\boldsymbol{\mu}_t, \sigma_t^2 \boldsymbol{\Sigma}_t)$$

where n is the sample size and $s_{t,\delta}^2$, $\boldsymbol{\mu}_t$, and $\boldsymbol{\Sigma}_t$ are computed according to (1), (3), and (4) in *Conditional posterior distribution of model parameters* in *Methods and formulas* of [BMA] **bmaregress** (with index j substituted for t, that is, conditional on model M_t instead of M_j). $\boldsymbol{\theta}_{t,p_t} = (\boldsymbol{\beta}'_{t,p_t}, \alpha)'$ is a $(p_t + 1) \times 1$ vector of the p_t regression coefficients included in model M_t and the intercept. Regression coefficients that are not included in M_t are assigned 0s. Then a $(p + 1) \times 1$ vector $\boldsymbol{\theta}_t$ includes all p regression coefficients and the intercept from iteration t.

The result is a sample $\{\boldsymbol{\theta}_t, \sigma_t^2\}_{t=1}^T$ from the BMA posterior distribution of model parameters.

Simulated model space

For a random g, the exact posterior model distribution is not available, and `bmaregress` provides an MCMC sample (m_t, g_t), for $t = 1, 2, \ldots, T$, from the BMA posterior distribution of models and parameter g. `bmacoefsample` command reuses this sample to simulate model parameters.

For each t, regression coefficients that are not included in m_t are assigned 0s. The remaining p_t regression coefficients and the intercept, $\boldsymbol{\theta}_{t,p_t}$, and variance σ_t^2 are drawn according to

$$\sigma_t^2 | g_t, m_t \;\sim\; \text{InverseGamma}\left(\frac{n-1}{2}, \frac{s_{t,\delta}^2}{2}\right)$$

$$\boldsymbol{\theta}_{t,p_t} | \sigma_t^2, g_t, m_t \;\sim\; N_{p_t+1}(\boldsymbol{\mu}_t, \sigma_t^2 \boldsymbol{\Sigma}_t)$$

where n is the sample size and $s_{t,\delta}^2$, $\boldsymbol{\mu}_t$, and $\boldsymbol{\Sigma}_t$ are computed according to (1), (3), and (4) in *Conditional posterior distribution of model parameters* in *Methods and formulas* of [BMA] **bmaregress** (with index j substituted for t, that is, conditional on model m_t instead of M_j).

Let $\boldsymbol{\theta}_t$ be a $(p+1) \times 1$ vector that includes all p regression coefficients and the intercept from iteration t. Then the result is a sample $\{\boldsymbol{\theta}_t, \sigma_t^2\}_{t=1}^T$ from the BMA posterior distribution of model parameters.

Also see

Description

The following Bayesian model averaging (BMA) postestimation commands are available after bmaregress:

Command	Description
bmacoefsample	posterior samples of regression coefficients
bmagraph	graphical summaries
bmagraph pmp	model-probability plots
bmagraph msize	model-size distribution plots
bmagraph varmap	variable-inclusion maps
bmagraph coefdensity	coefficient posterior density plots
bmastats	summaries for models and predictors
bmastats models	posterior model and variable-inclusion summaries
bmastats msize	model-size summary
bmastats pip	posterior inclusion probabilities (PIPs) for predictors
bmastats jointness	jointness measures for predictors
bmastats lps	log predictive-score
bmapredict	BMA predictions
* estimates	cataloging estimation results

* estimates table, estimates stats, and estimates selected are not appropriate with bmaregress estimation results.

The following standard Bayesian postestimation commands are of particular interest after bmaregress followed by bmacoefsample (see [BMA] **bmacoefsample**):

Command	Description
bayesgraph	graphical summaries and convergence diagnostics
bayesstats summary	Bayesian summary statistics for model parameters and their functions
bayesstats ess	effective sample sizes and related statistics
bayesstats ppvalues	Bayesian predictive p-values
bayespredict	Bayesian predictions
* estimates	cataloging estimation results

* estimates table, estimates stats, and estimates selected are not appropriate with bmacoefsample estimation results.

Remarks and examples

BMA postestimation includes both BMA-specific postestimation tools and some of the standard Bayesian postestimation tools. BMA-specific postestimation is focused on BMA convergence, model exploration, and the importance of predictors based on how frequently they are included in various models. The standard Bayesian postestimation is used for exploring posterior samples of model parameters and regression coefficients, in particular.

After you fit a BMA model by using the `bmaregress` command, the first step of a BMA postestimation analysis, and any Bayesian analysis in general, is to check convergence. The `bmagraph pmp` command provides a visual way to assess BMA convergence by plotting the analytical posterior model probability (PMP) (or its estimate) versus the frequency PMP, estimated from a Markov chain Monte Carlo (MCMC) model sample. If a model converged, the analytical and frequency PMP will be close. See [BMA] **bmagraph pmp** and *Convergence of BMA* in *Remarks and examples* of [BMA] **bmaregress** for details. The PMP graph is also useful to explore the models with high PMP.

Once BMA convergence is established, you can explore the BMA results in more detail. You can use the `bmastats models` command to examine models with high PMP and models that include certain predictors. The `bmagraph pmp` and `bmagraph varmap` commands are essentially graphical counterparts of `bmastats models`. The latter produces a variable-inclusion map, a map that shows each model and all predictors included in that model with color-coded bars that represent the signs of the corresponding coefficients.

The model-size summaries as provided by `bmastats msize` are useful to measure the complexity of the visited models—small-sized models tend to have a few strong predictors whereas larger-sized models tend to have many weaker predictors. In addition to summaries, you can use the `bmagraph msize` command to explore the entire model-size distribution.

The `bmastats pip` command reports PIPs for predictors. In the context of BMA, PIP is used to measure the importance of one predictor relative to the others. And you can also examine the tendency of pairs of predictors to be included together, separately, or independently by computing various jointness measures using `bmastats jointness`.

The `bmagraph coefdensity` command can be used to plot posterior distributions of regression coefficients. These distributions are available analytically for models with a fixed g. But to fully explore the regression coefficients, you will often need to have an MCMC sample from their posterior distribution. You can use the `bmacoefsample` command to generate this sample. In fact, some of the postestimation commands require that a sample like this is available. (`bmaregress` does not generate this sample automatically to save time, because it is not always needed in a BMA analysis.)

The `bmastats lps` command computes log predictive-scores, which can be used to compare goodness of fit and predictive performance of models; see [BMA] **bmastats lps** for details.

Finally, you can use `bmapredict` to obtain various predictions after a BMA model, both in and out of sample; see [BMA] **bmapredict**.

Once an MCMC sample of regression coefficients or, more generally, model parameters is available, you can use some of the standard Bayesian postestimation tools as described in [BAYES] **Bayesian postestimation**. For instance, you can use the `bayesstats summary` command to explore posterior summaries, including credible intervals, for regression coefficients; see [BAYES] **bayesstats summary**. And you can use the `bayespredict` command to obtain more complicated BMA predictions; see [BAYES] **bayespredict**.

You can use `estimates store` and `estimates save` (see [R] **estimates**) to store or save estimation results after `bmaregress` and `bmacoefsample`. The `estimates` commands require that you first use the `saving()` option with both BMA commands to save their MCMC simulation results. If you use one of the `estimates` commands after `bmaregress`, you will store BMA estimation results and the MCMC model simulation results. If you do this after `bmacoefsample`, in addition to the `bmaregress` estimation results, you will store additional estimation results from `bmacoefsample` and the MCMC model-parameter simulation results.

Also see

[BMA] **bmaregress** — Bayesian model averaging for linear regression

[BMA] **bmacoefsample** — Posterior samples of regression coefficients

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

[BMA] **Glossary**

[BAYES] **Bayesian postestimation** — Postestimation tools after Bayesian estimation

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

Description

The following graphical postestimation commands are available with bmagraph after bmaregress:

Command	Description
bmagraph pmp	model-probability plots
bmagraph msize	model-size distribution plots
bmagraph varmap	variable-inclusion maps
bmagraph coefdensity	coefficient posterior density plots

Remarks and examples

See [BMA] **BMA postestimation** for a short introduction to Bayesian model averaging (BMA) postestimation.

The bmagraph pmp command is used for checking BMA convergence and for exploring models with high posterior model probability; see [BMA] **bmagraph pmp**.

The bmagraph msize command plots the prior and posterior model-size distributions and is useful for examining model complexity; see [BMA] **bmagraph msize**.

The bmagraph varmap command produces a variable-inclusion map, a map that shows each model and all predictors included in that model with color-coded bars that represent the signs of the corresponding coefficients. See [BMA] **bmagraph varmap**.

The bmagraph coefdensity command plots posterior distributions of regression coefficients; see [BMA] **bmagraph coefdensity**.

Also see

[BMA] **bmaregress** — Bayesian model averaging for linear regression

[BMA] **bmacoefsample** — Posterior samples of regression coefficients

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

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

[BMA] **Glossary**

Description

bmagraph coefdensity provides posterior density plots for regression coefficients after the bmaregress command. This command is useful for exploratory analysis of regression coefficients. For a given predictor, bmagraph coefdensity shows the posterior probability of that predictor not being included in a model and, separately, the smooth density of its coefficient conditional on the predictor being included.

Quick start

Plot the analytical posterior density of the coefficient for predictor x after a Bayesian model averaging (BMA) regression model with a fixed g

    bmagraph coefdensity {x}

Same as above, but plot the MCMC-sample posterior density instead of the analytical one; bmacoefsample must be run first

    bmaregress, saving(bmamodelmcmc)
    bmacoefsample
    bmagraph coefdensity {x}, mcmcsample

Plot posterior densities of coefficients for predictors x1 and x2, and customize the look of the vertical probability-mass line at zero for the second graph

    bmagraph coefdensity {x1 x2}, przeroline2opts(lcolor(green))

Same as above, but do not show the lines for probability mass at zero

    bmagraph coefdensity {x1 x2}, noprzerolines

Customize the kernel function used and the line pattern for the graph for {x1}

    bmagraph coefdensity {x1 x2}, dens1opts(kernel(gaussian) lpattern(dot))

Plot densities for all coefficients on one graph in two rows

    bmagraph coefdensity _all, combine(rows(2))

Plot the MCMC-sample posterior density of the coefficient for predictor x after a BMA regression model with a random g

    bmaregress, saving(bmamodelmcmc2)
    bmacoefsample
    bmagraph coefdensity {x}

Equivalent to the above bmagraph coefdensity

    bmagraph coefdensity {x}, mcmcsample

Menu

Statistics > Bayesian model averaging > Coefficient densities

Syntax

Density plot for one coefficient

> bmagraph <u>coefdens</u>ity {*coef*} [, *singleopts*]

Density plot for multiple coefficients

> bmagraph <u>coefdens</u>ity *coefspec* [, *multiopts*]

Density plot for all coefficients

> bmagraph <u>coefdens</u>ity _all [, *multiopts*]

coef is an unabbreviated name of the variable or a factor level used in the bmaregress model. *coefspec*
may be a regression coefficient {*coef*}, a list of coefficients {*coef1*}, {*coef2*}, etc., or, equivalently,
{*coef1 coef2 ...*}. _all refers to all regression coefficients.

singleopts	Description
Main	
<u>analytic</u>	plot analytical posterior density; default for fixed g
<u>mcmcsample</u>	plot MCMC-sample posterior density; only choice for random g
<u>name</u>(*name*, ...)	specify name of graph
<u>saving</u>(*filename*, ...)	save graph in file
[no]<u>przeroline</u>	plot or hide vertical line for probability mass at zero
<u>przerolineopt</u>(*line_options*)	affect rendition of vertical line for probability mass at zero
Density options	
densopts	density plot options
Y axis, X axis, Titles, Legend, Overall	
twoway_options	any options other than by(), name(), and saving() documented in [G-3] *twoway_options*

multiopts	Description
Main	
analytic	plot analytical posterior density; default for fixed g
mcmcsample	plot MCMC-sample posterior density; only choice for random g
name(*namespec*, ...)	specify names of graphs
saving(*filespec*, ...)	save graphs in files
[no]przerolines	plot or hide vertical lines for probability mass at zero
przerolineopts(*line_options*)	affect rendition of all vertical lines for probability mass at zero
przeroline#opts(*line_options*)	affect rendition of #th vertical line for probability mass at zero
combine[(*grcombineopts*)]	display plots on one graph
Density options	
dens#opts(*densopts*)	density options for #th plot
densopts	density options for all plots
Y axis, X axis, Titles, Legend, Overall	
twoway_options	any options other than by(), name(), and saving() documented in [G-3] *twoway_options*

Options

> ⌐ Main ⌐

analytic specifies that the density be computed analytically. This is the default for models with a fixed g and is not available for models with a random g. This option may not be specified together with option mcmcsample.

mcmcsample specifies that the density be estimated from a Markov chain Monte Carlo (MCMC) sample of coefficients. This option uses an MCMC sample generated by the bmacoefsample command and may not be specified together with option analytic. This is the only choice for models with a random g; that is, option mcmcsample is implied with a random g. With a fixed g, if an MCMC sample is not available, option mcmcsample is not allowed. You must use bmacoefsample to generate an MCMC sample to produce coefficient density graphs based on sample estimates.

name(*namespec*[, replace]) specifies the name of the graph or multiple graphs. See [G-3] *name_option* for a single graph. If multiple graphs are produced, then the argument of name() is either a list of names or *stub*, in which case graphs are named *stub*1, *stub*2, and so on. With multiple graphs, if name() is not specified, name(Graph__#, replace) is assumed; thus, the produced graphs may be replaced by subsequent bmagraph commands.

The replace suboption causes existing graphs with the specified name or names to be replaced.

saving(*filespec*[, replace]) specifies the filename or filenames to use to save the graph or multiple graphs to disk. See [G-3] *saving_option* for a single graph. If multiple graphs are produced, then the argument of saving() is either a list of filenames or *stub*, in which case graphs are saved with filenames *stub*1, *stub*2, and so on.

The replace suboption specifies that the file (or files) may be replaced if it already exists.

przerolines and noprzerolines show or hide the vertical lines for probability mass at zero. By default, the lines are shown.

przerolineopts(*line_options*) and przeroline#opts(*line_options*) affect the rendition of the vertical lines for probability mass at zero; see [G-3] **line_options**. przerolineopts() controls the look of all vertical lines but may be overridden for specific lines by using the respective options przeroline#opts().

combine[(*grcombineopts*)] specifies the display of all plots of coefficients as subgraphs on one graph. By default, a separate graph is produced for each plot when multiple coefficients are specified.

grcombineopts is any of the options documented in [G-2] **graph combine**.

_____ Density options _____

dens#opts(*densopts*) specifies density options for the #th density plot.

densopts specify options for the (kernel) density plot; see the options documented in [G-2] **graph twoway kdensity**, except options horizontal and range(*varname*). When option analytic is assumed, the density functions are known; thus, the kernel density estimation is not needed. In that case, *densopts* include only n() and *cline_options*, described in [G-2] **graph twoway kdensity**.

_____ Y axis, X axis, Titles, Legend, Overall _____

twoway_options are any of the options documented in [G-3] **twoway_options**, excluding by(), name(), or saving(). These include, for instance, options for titling the graph (see [G-3] **title_options**).

Remarks and examples

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.

A posterior density for a regression coefficient in a BMA linear regression is a mixture of a discrete and continuous components. A discrete component corresponds to the posterior probability of noninclusion, the probability that the corresponding predictor is not included in a model after observing the data, which is equivalent to the posterior probability that the coefficient is equal to zero. This component is represented by a red vertical line with a probability mass at zero equal to one minus the posterior inclusion probability (PIP). A continuous component corresponds to the continuous density conditional on the predictor being included or, equivalently, on the coefficient not being equal to zero.

For a BMA linear regression with a fixed g, the posterior densities of regression coefficients are known and can be plotted at any prespecified x-axis points; see *Posterior distributions of regression coefficients* in *Methods and formulas* of [BMA] **bmaregress**. We refer to these densities as analytical posterior densities. Alternatively, if an MCMC sample of regression coefficients, as produced by bmacoefsample, is available, the posterior densities can be estimated from this sample by using kernel density estimation (see [R] **kdensity**) when you specify the mcmcsample option. We refer to these posterior densities as MCMC-sample posterior densities. With a random g, this is the only option because analytical posterior densities are not available. That is, with a random g, you must run the bmacoefsample first before you can run bmagraph coefdensity.

▷ Example 1: Posterior density plots for BMA linear regression using enumeration

Recall the performance dataset (Chatterjee and Hadi 2012, sec. 3.3) analyzed in example 1 of [BMA] **bmaregress**, where the employees' satisfaction with their supervisors, rating, is modeled by six potential predictors.

We fit a BMA linear regression for the rating outcome using all six other variables as potential predictors. By default, the model space is fully explored using enumeration.

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

. 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.
```

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.

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—here the regression coefficients.

Let us inspect the analytical marginal posterior density of the coefficient for `complaints`.

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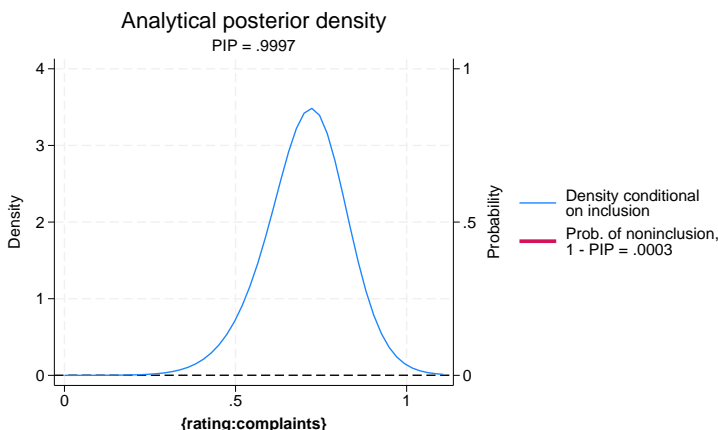

Analytical posterior density
PIP = .9997

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. Thus, the posterior density of {complaints} is essentially a continuous density, with a mean of roughly 0.7 and with a slightly heavier left tail. Most of the mass of the distribution is between roughly 0.025 and 1.1, away from 0.

Let's plot the analytical marginal posterior density for the {learning} coefficient and request a cyan color for the density line:

```
. bmagraph coefdensity {learning}, lcolor(cyan)
```


Analytical posterior density
PIP = .2525
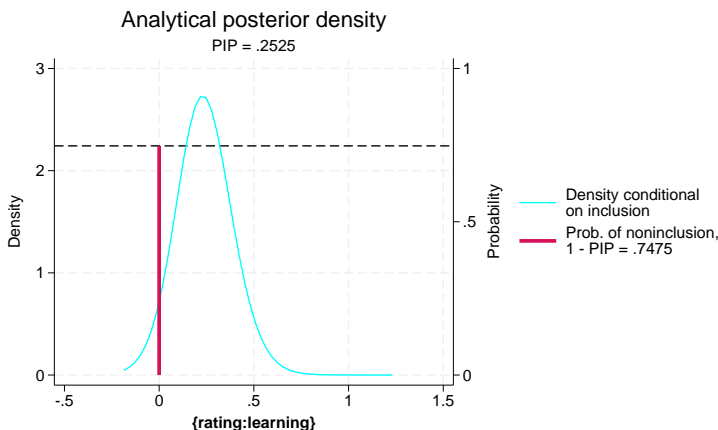
The probability of a noninclusion for {learning} is about 0.75, so we see the red vertical line with the horizontal reference line at roughly 0.75. The look of the vertical line can be controlled via the `przerolineopts()` option. Conditional on the inclusion, the continuous density has its mass between roughly -0.2 and 0.7, which includes 0.

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

Instead of showing the density of each coefficient separately, we can plot the densities of all coefficients by specifying _all and request they be plotted on the same graph using the combine() option.

. bmagraph coefdensity _all, combine(rows(2))

Analytical posterior density
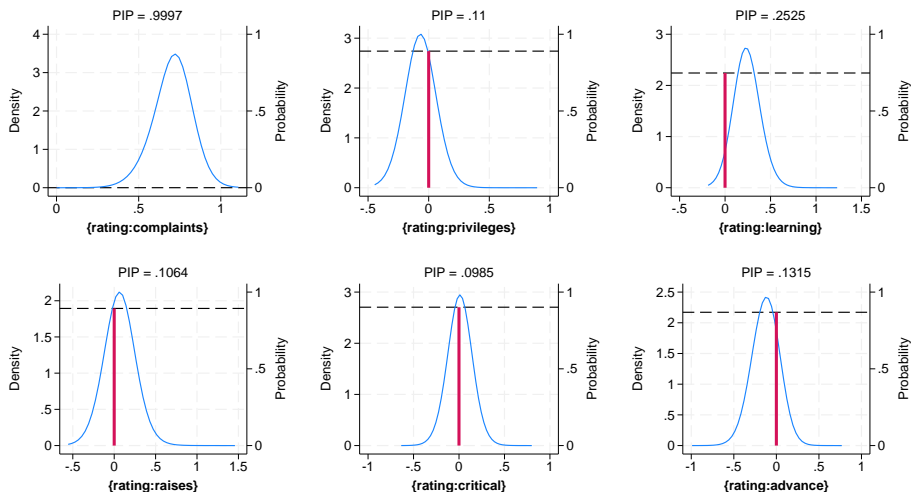
The rows(2) suboption of combine() specifies that plots be organized in two rows.

◁

▷ Example 2: Posterior density plots for BMA linear regression using sampling

We refit the same BMA model as in example 1 but now use the sampling option to request the MCMC model composition (MC3) algorithm instead of the default model enumeration.

```
. 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.
```

Although the MC3 sampling is used for the model-space exploration, we can still use bmagraph coefdensity directly after bmaregress to plot the analytical posterior densities of regression coefficients. This is because we are still assuming a fixed g parameter in our model.

Let's compare the analytical and MCMC-sample posterior densities. To obtain the sample estimates, we must first simulate an MCMC sample from the posterior distribution of regression coefficients by using bmacoefsample. To use bmacoefsample, we first need to save the simulation results produced by bmaregress.

```
. bmaregress, saving(bmaex2, replace)
note: file bmaex2.dta not found; file saved.

. bmacoefsample, rseed(18)

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

To request the density estimate based on the MCMC sample, we specify the `mcmcsample` option. We plot the MCMC-sample posterior density estimate for the `{learning}` coefficient.

```
. bmagraph coefdensity {learning}, mcmcsample
```
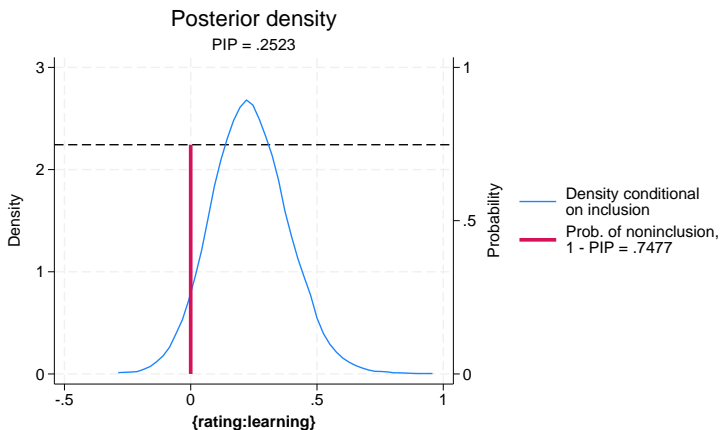

To more easily compare the plot above with the one in example 1, we rescale the x axis to have the same range.

```
. bmagraph coefdensity {learning}, mcmcsample xscale(range(-.5 1.5))
```
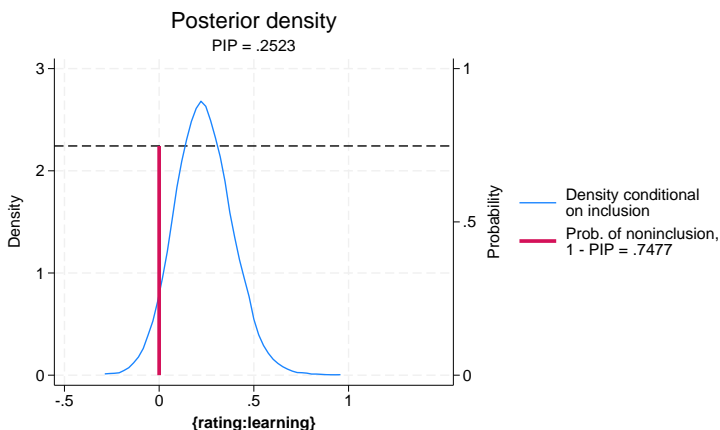

The plot is similar to that in example 1.

The marginal density of the `learning` coefficient can also be estimated using the `bayesgraph kdensity` command (see [BAYES] **bayesgraph**). This command, however, incorporates the mass at zero and shows the distribution mixture as one smooth curve, which may not be of practical use in the context of BMA models.

```
. bayesgraph kdensity {learning}
```


Density of **rating:learning**
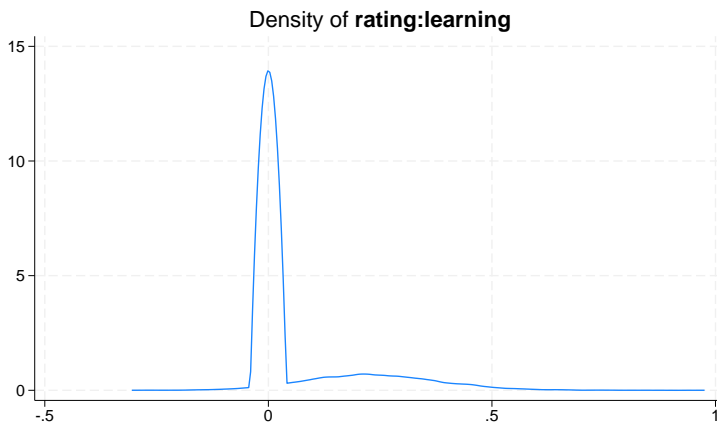
Notice how the density curve shrinks toward 0 given the relatively large probability mass $(1 - \text{PIP} = 0.7477)$ at that point.

For comparison with the density plots in example 1, we will plot MCMC-sample densities for all coefficients in our model. To demonstrate, we also use dens6opts(lcolor(cyan)) to specify a cyan color for the density on the sixth plot corresponding to {rating:advance}.

```
. bmagraph coefdensity _all, mcmcsample combine(rows(2)) dens6opts(lcolor(cyan))
```


Posterior density
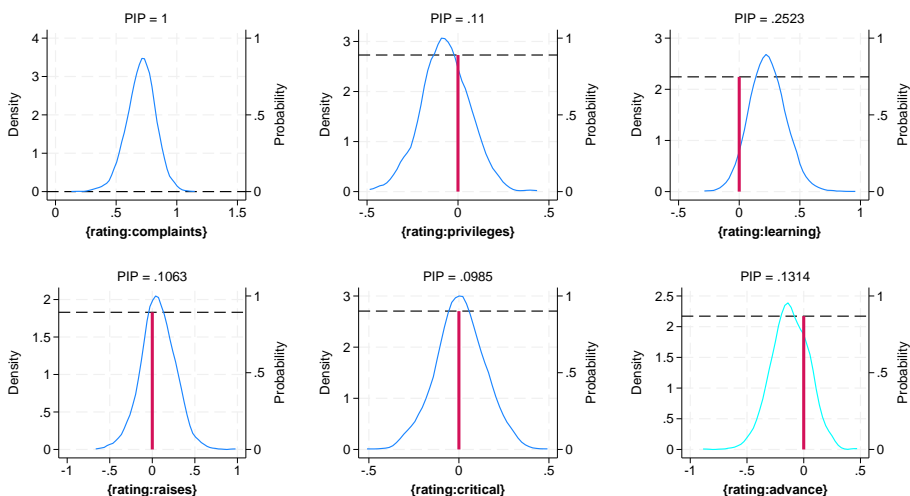
If adequate exploration of the model space by bmaregress occurred and a sufficiently large MCMC sample is generated by bmacoefsample, the analytical and MCMC-sample posterior densities are expected to be similar. This is the case in our examples (after we adjust for the scales of the x axes).

With a random g, the analytical posterior densities are not available, and the MCMC sampling is needed to approximate these distributions. In that case, the bmacoefsample command must be run first to obtain posterior samples of regression coefficients, as we showed above. bmagraph coefdensity can then be used to plot an MCMC-sample posterior density, except we do not need to specify the mcmcsample option—it is implied with a random g.

◁

Methods and formulas

Continuing with the notation in *Methods and formulas* of [BMA] **bmaregress**, we 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),

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

where $\boldsymbol{\delta}_0$ denotes the point mass at 0 and $\mathbf{f}_k$ is defined by (10) in *Posterior distributions of regression coefficients* in *Methods and formulas* of [BMA] **bmaregress** for models with a fixed g (option analytic). When the mcmcsample option is assumed, either by default for models with a random g or if specified for models with a fixed g, $\mathbf{f}_k$ is computed using the kdensity (see [R] **kdensity**) command based on the coefficient sample generated by bmacoefsample.

The vertical line for the probability mass at 0 corresponds to $1 - \text{PIP}_k$, where PIP_k is the PIP of X_k computed as

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

and J indexes a subspace of the model space visited by bmaregress. Also see *Posterior inclusion probability* in *Methods and formulas* of [BMA] **bmaregress**.

In the above formula, for models with a fixed g parameter, $P(M_j|\mathbf{y})$ is either $P_a(M_j|\mathbf{y})$ (option analytic) or $P_f(M_j|\mathbf{y})$ (option mcmcsample). For models with a random g parameter, $P(M_j|\mathbf{y})$ is $P_f(M_j|\mathbf{y})$. These quantities are defined, respectively, by (7) and (8) in *Posterior model probability* in *Methods and formulas* of [BMA] **bmaregress**.

Reference

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

Also see

Description

bmagraph msize provides a graphical summary for the posterior and prior model-size distributions after the bmaregress command.

Quick start

Plot posterior and prior model-size distributions after fitting a Bayesian model averaging (BMA) linear regression

> bmagraph msize

Same as above, but plot the posterior model-size distribution only

> bmagraph msize, noprior

Menu

Statistics > Bayesian model averaging > Model-size distributions

Syntax

> bmagraph msize [, *options*]

options	Description
Main	
<u>const</u>ant	include constant term in model-size computations; default is no constant
noprior	suppress prior model-size distribution shown by default
priorlineopts(*cline_options*)	affect rendition of prior line
anlineopts(*cline_options*)	affect rendition of analytical posterior line; ignored with random *g*
freqlineopts(*cline_options*)	affect rendition of frequency posterior line; ignored with model enumeration
postlineopts(*cline_options*)	affect rendition of all posterior lines
Line options	
cline_options	affect rendition of all plotted lines
Y axis, X axis, Titles, Legend, Overall	
twoway_options	any options other than by() documented in [G-3] *twoway_options*

Options

constant specifies that the constant term be included in model-size computations. By default, the constant term is not included.

noprior specifies that the prior model-size distribution not be shown on the plot. It is shown by default.

priorlineopts(*cline_options*) affects the rendition of the prior line; see [G-3] *cline_options*.

anlineopts(*cline_options*) affects the rendition of the analytical posterior line; see [G-3] *cline_options*. This option is ignored for BMA models with a random g.

freqlineopts(*cline_options*) affects the rendition of the frequency posterior line; see [G-3] *cline_options*. The frequency model-size distribution is plotted whenever sampling is used. This option is ignored with model enumeration.

postlineopts(*cline_options*) affects the rendition of the analytical and frequency posterior lines; see [G-3] *cline_options*.

cline_options affects the rendition of all plotted lines; see [G-3] *cline_options*.

twoway_options are any of the options documented in [G-3] *twoway_options*, excluding by(). These include options for titling the graph (see [G-3] *title_options*) and for saving the graph to disk (see [G-3] *saving_option*).

Remarks and examples

See *Remarks and examples* in [BMA] **bmastats msize** for a general discussion of a BMA model size.

A model-size distribution is used to explore model complexity. The prior model-size distribution describes our a priori assumption about model size. The posterior model-size distribution describes the effect of the data on the BMA model. For this purpose, we compare the posterior and prior model-size distributions. If the posterior is skewed to the left with respect to the prior, then the data favor smaller models than assumed by the prior. If the posterior is skewed to the right, then the data favor larger models than assumed by the prior.

For the prior model-size distribution, bmagraph msize always uses analytical computation. In the case of sampling, it is conditional on the visited models. For the posterior model-size distribution, it plots the analytical distribution for a fixed g and the MCMC frequency-based or simply frequency distribution for a random g and whenever sampling is used.

▷ Example 1: Model-size distributions for BMA models using enumeration

We consider the performance dataset (Chatterjee and Hadi 2012, sec. 3.3) analyzed in example 1 of [BMA] **bmastats msize**. In that example, we explored only a few summaries of the model-size distributions. Here we describe the entire distributions.

The rating variable is regressed on all predictors from complaints to advance. By default, because of the small number of predictors, six, the model space is explored fully by using enumeration.

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

. 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.
```

There is a total of $2^6 = 64$ models in the fully explored model space. Let's use `bmagraph msize` to draw the posterior and prior model-size distributions.

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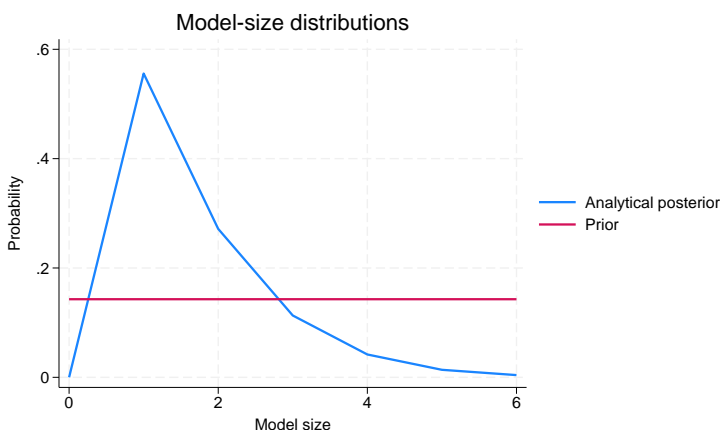

The reported model size does not include the constant, so its range is between 0 and 6. You may include the constant by specifying the `constant` option. By default, an uninformative uniform prior is assumed for the model size. The posterior model-size distribution is skewed to the left. Its mode is 1, so the posterior favors smaller models.

A frequency-based posterior estimate of the model-size distribution is not available in this example because there is no MCMC sample with model enumeration.

◁

▷ Example 2: Model-size distributions for BMA models using MCMC model composition (MC3) sampling

Continuing with example 1, we fit the same BMA model, but this time we use the MC3 sampling algorithm by specifying the sampling option. We also specify the rseed() option for reproducibility.

```
. 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*.

Instead of enumerating models (fully exploring the space), bmaregress explored only half the model space. It visited 32 out of the total 64 models. We inspect the effect of this on the prior and posterior model-size distributions.

```
. bmagraph msize
```
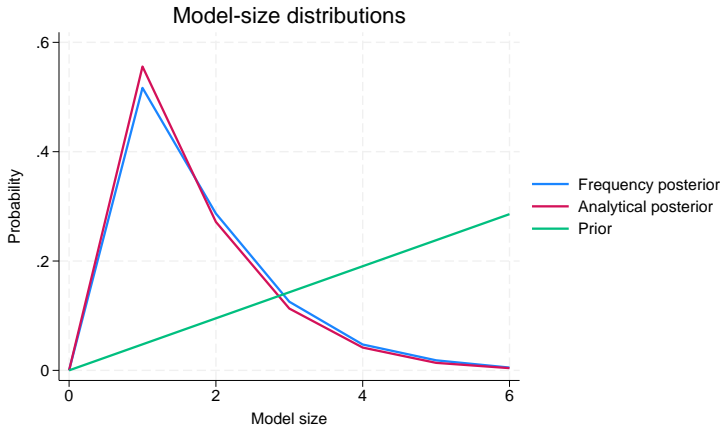

Although we used the same model prior as in example 1, the prior model-size distribution looks different. This is because our explored model space now contains 32 models instead of all 64, and the prior model-size distribution is conditional on the visited models.

The model of size 1, the one that includes `complaints`, has the highest posterior probability of about 0.56.

With a fixed g when we fit a BMA model using MC3 sampling, in addition to the analytical model-size distribution, the frequency posterior model-size distribution is available. Provided that the model-space sampling converges, the analytical and frequency distributions should be close. In our example, the analytical and frequency model-size distributions are nearly identical.

◁

▷ Example 3: Model-size distributions for BMA models with random g

Both example 1 and example 2 used a fixed g. Let's explore the case of a random g. (An in-depth coverage of the effects of the g-prior on model complexity can be found in, for example, Ley and Steel [2012].)

To demonstrate, we will use a robust prior for g.

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

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

bmaregress now uses MC3 sampling for the models and adaptive Metropolis–Hastings sampling for g.

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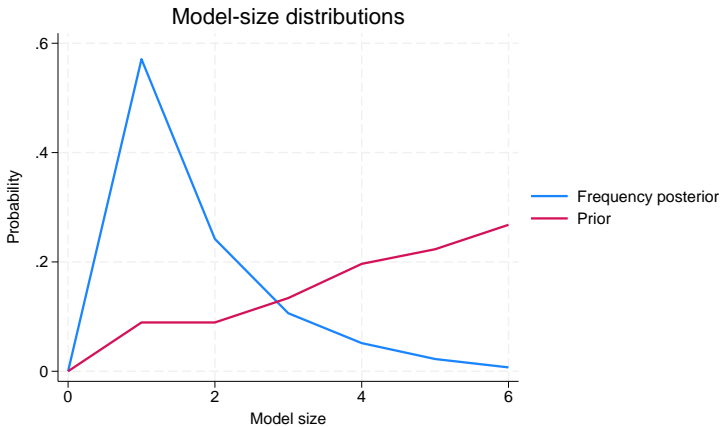

The analytical posterior model-size distribution is not available with a random g. The frequency posterior model-size distribution is similar to that in example 2 for fixed $g = 36$. Particularly, the null model was not visited by the MC3 sampler, and models of size 1 and 2 have the highest posterior probabilities (both above 0.2), but the mode of the posterior distribution here is 1.

Methods and formulas

See *Methods and formulas* in [BMA] **bmastats msize**.

References

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

Ley, E., and M. F. J. Steel. 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.

Also see

[BMA] **bmastats msize** — Model-size summary after BMA regression

[BMA] **bmagraph** — Graphical summary for models and predictors after BMA regression

[BMA] **bmaregress** — Bayesian model averaging for linear regression

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

[BMA] **Glossary**

Description

bmagraph pmp provides graphical summary information for the models visited by the bmaregress command, including their posterior model probabilities (PMPs) and cumulative PMPs (CPMPs). This command is useful for checking Markov chain Monte Carlo (MCMC) convergence of a Bayesian model averaging (BMA) model.

Quick start

Plot PMPs for the default top 100 models with highest PMPs

    bmagraph pmp

Same as above, but request CPMPs

    bmagraph pmp, cumulative

Same as above, but suppress frequency PMPs, and show only the top 20 models with highest PMPs

    bmagraph pmp, cumulative nofreqline top(20)

Menu

Statistics > Bayesian model averaging > Model-probability plots

Syntax

> bmagraph pmp [, *options*]

options	Description
Main	
<u>top</u>(*#*)	show top *#* models with highest PMPs; default is top(100)
<u>cumulative</u>	display CPMPs instead of PMPs for plotted models
<u>pmpcutoff</u>(*#*)	do not show models with PMP less than *#*; default is *# = 0*
<u>maxmodels</u>(*#*)	plot results for the first *#* models; default is maxmodels(1000)
<u>anlineopts</u>(*cline_options*)	affect rendition of analytical-PMP or harmonic-PMP line
[no]freqline	show or suppress frequency-PMP line
<u>freqlineopts</u>(*cline_options*)	affect rendition of frequency-PMP line
Line options	
cline_options	affect rendition of all plotted lines
Y axis, X axis, Titles, Legend, Overall	
twoway_options	any options other than by() documented in [G-3] *twoway_options*
all	plot all models

all does not appear in the dialog box.

Options

> **Main**

top(*#*) specifies that the top *#* models with highest PMPs be shown. By default, the top 100 are shown. The maximum number of models plotted is controlled by option maxmodels(). Use top() in combination with maxmodels(), if you need to plot more than 1,000 models.

cumulative specifies that CPMPs be reported for the models instead of the default PMPs.

pmpcutoff(*#*) specifies that models with PMPs less than *#* should not be shown. The default is pmpcutoff(0). This option is useful when there are many models with small PMPs.

maxmodels(*#*) specifies the maximum number of models to be plotted. The default is maxmodels(1000). When this option is specified together with option top($#_1$), the number of plotted models is the minimum between *#* and $#_1$.

anlineopts(*cline_options*) affects the rendition of the analytical-PMP line; see [G-3] *cline_options*. With a random *g*, this option affects the harmonic-PMP line, where the harmonic-mean estimator is used to approximate the analytical PMPs.

freqline and nofreqline show or suppress the frequency-PMP line. By default, the frequency-PMP line is shown when it is available. Frequency-PMP line is not available with model enumeration, so nofreqline is implied and freqline is ignored, if specified, in that case.

freqlineopts(*cline_options*) affects the rendition of the frequency-PMP line when it is available; see [G-3] *cline_options*. With model enumeration, this option is ignored if specified.

cline_options affects the rendition of all plotted lines; see [G-3] *cline_options*.

Y axis, X axis, Titles, Legend, Overall

twoway_options are any of the options documented in [G-3] *twoway_options*, excluding by(). These include options for titling the graph (see [G-3] *title_options*) and for saving the graph to disk (see [G-3] *saving_option*).

The following option is available with bmagraph pmp but is not shown in the dialog box:

all specifies that all models, up to maxmodels(), be shown on the plot. This option may be useful when the number of visited models is small. This option may not be specified together with top().

Remarks and examples

The bmagraph pmp command is used to visualize the posterior distribution of the explored model space and to check for sampling convergence when the model space is explored using the MC3 algorithm. The analytical and MCMC frequency-based or simply frequency posterior probability distributions should be close when the model space is sufficiently explored.

When the model space is explored fully by using enumeration, the frequency posterior probabilities coincide with the analytical ones, and there is no need to check for convergence. The frequency-PMP line is not plotted in this case. With a random g, analytical-PMP line is not available and a harmonic-PMP line is displayed instead. This line represents a harmonic-mean estimate of the analytical PMPs.

▷ Example 1: Analytical and frequency PMPs

Recall the performance dataset (Chatterjee and Hadi 2012, sec. 3.3) analyzed in example 1 of [BMA] **bmaregress**, where the employee satisfaction with their supervisors, rating, is modeled by six potential predictors.

Let's fit a BMA linear regression to these data using bmaregress and explore PMPs using bmagraph pmp. With a few predictors, bmaregress explores a full space of all possible models. In this case, we do not need to check for sampling convergence because no sampling is performed. But we can still use bmagraph pmp to explore PMPs.

To demonstrate, let's use sampling instead of the default model enumeration in our example by specifying the sampling option with bmaregress. We use the notable option to suppress the output table because we are not interested in the regression coefficients but in the models explored by bmaregress.

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

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

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				

We use `bmagraph pmp` to display both analytical and frequency PMPs of the 32 visited models.

```
. bmagraph pmp
```


All 32 visited models shown.
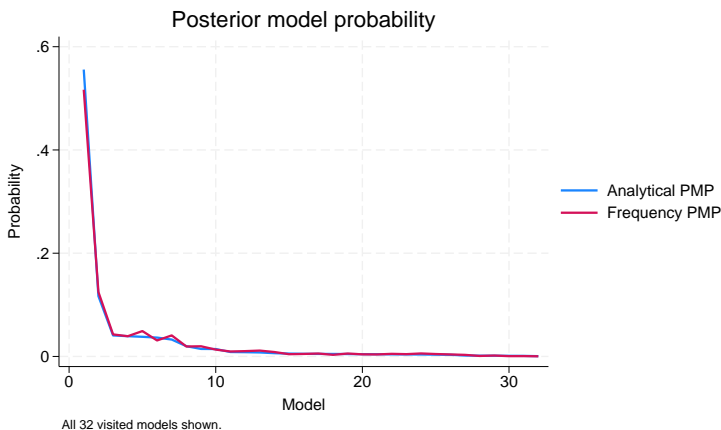
The two lines are nearly identical, which is a strong indication of convergence.

Instead of PMPs, we can use the `cumulative` option to plot CPMPs:

. bmagraph pmp, cumulative

Cumulative posterior model probability

All 32 visited models shown.
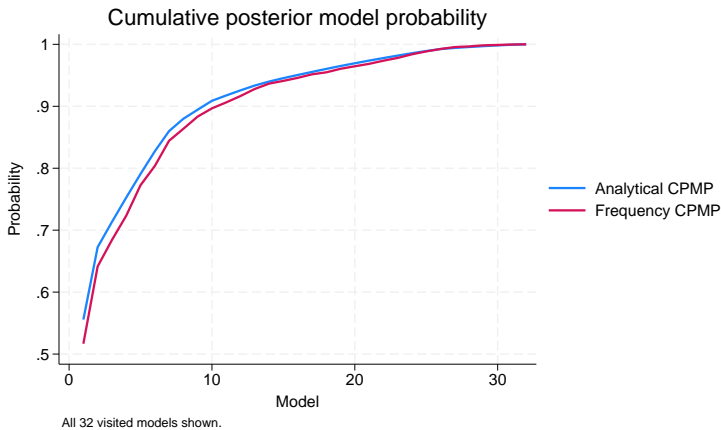
The plot shows that the first 10 models account for about 90% of the total probability mass.

◁

▷ Example 2: Harmonic and frequency PMPs

The BMA model in example 1 assumed a fixed parameter g that controls the shrinkage of regression coefficients toward zero. We can instead specify a prior distribution, a hyperprior, for g. For instance, we use a robust hyperprior below.

```
. bmaregress rating complaints-advance, gprior(robust) rseed(18) notable
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
```

With a random g, there is no analytical formula for computing PMPs. In this case, MCMC convergence is evaluated by comparing the frequency PMP to the harmonic-mean approximation of the analytical PMP. To demonstrate, let's also use the `top(20)` option to plot the top 20 models with the highest PMPs.

```
. bmagraph pmp, top(20)
```


Top 20 models shown out of 34 visited.
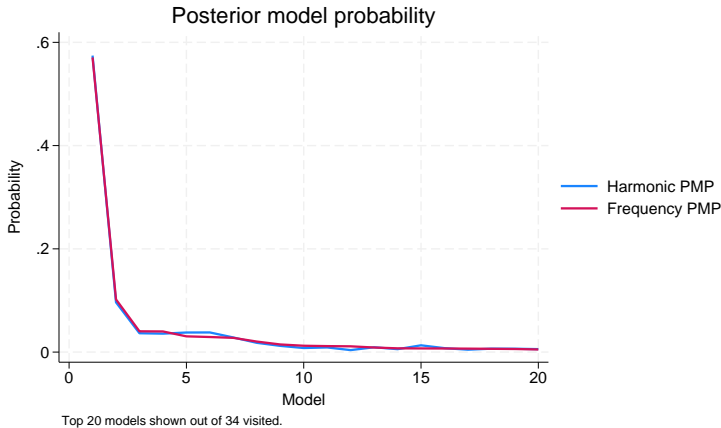
The above plot should be interpreted with extra caution because the assessment of the MCMC convergence here is subject to the accuracy of the harmonic-mean approximation of the analytical PMP, in addition to the inherent sampling variability. In our example, there is no concern for the accuracy of the approximation or nonconvergence.

◁

Methods and formulas

For methods and formulas of PMPs, see *Posterior model probability* in *Methods and formulas* of [BMA] **bmaregress**.

The CPMP for a model j is a cumulative sum of the first jth highest PMPs. The analytical CPMP is a cumulative sum of analytical PMPs, and the frequency CPMP is a cumulative sum of frequency PMPs.

Reference

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

Also see

[BMA] **bmastats models** — Model and variable-inclusion summaries after BMA regression

[BMA] **bmagraph** — Graphical summary for models and predictors after BMA regression

[BMA] **bmaregress** — Bayesian model averaging for linear regression

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

[BMA] **Glossary**

Description

bmagraph varmap produces a variable-inclusion map after the bmaregress command for top models ranked by the highest posterior model probability (PMP), which include predictors with posterior inclusion probability (PIP) above 0.01.

Quick start

Inclusion map for predictors included in the top 100 models ranked by highest PMP

```
bmagraph varmap
```

Inclusion map for predictors with PIP of at least 0.2 and for top 10 models

```
bmagraph varmap, top(10) pipcutoff(0.2)
```

Same as above, but force all model bars to have equal widths, instead of widths being proportional to the model's PMP

```
bmagraph varmap, top(10) pipcutoff(0.2) equalwidths
```

Menu

Statistics > Bayesian model averaging > Variable-inclusion map

Syntax

> bmagraph varmap [, *options*]

options	Description
Main	
<u>top</u>(#)	show top # models with highest PMPs; default is top(100)
<u>pipc</u>utoff(#)	do not show predictors with PIP less than #; default is # = 0.01
<u>maxm</u>odels(#)	plot results for the first # models; default is maxmodels(1000)
<u>equalw</u>idths	request that equal-width model bars are displayed; default is widths proportional to PMP
<u>pmpn</u>ormalized	show normalized PMP
<u>poscoef</u>opts(*rbaropts*)	control the look of bars for positive coefficients
<u>negcoef</u>opts(*rbaropts*)	control the look of bars for negative coefficients
<u>zerocoef</u>opts(*rbaropts*)	control the look of bars for zero (not-included) coefficients
Range bar options	
rbaropts	control the look of all bars
Y axis, X axis, Titles, Legend, Overall	
twoway_options	any options other than by() documented in [G-3] *twoway_options*
<u>all</u>	plot all models

all does not appear in the dialog box.

Options

> ┌─────┐
> │ Main │
> └─────┘──

top(#) specifies that the top # models with the highest PMPs be shown. By default, the top 100 are shown. The maximum number of models plotted is controlled by option maxmodels(). Use top() in combination with maxmodels(), if you need to plot more than 1,000 models.

pipcutoff(#) specifies that predictors with PIPs less than # not be shown. The default is pipcutoff(0.01). This option is useful when there are many predictors with small PIPs.

maxmodels(#) specifies the maximum number of models to be plotted. The default is maxmodels(1000). When this option is specified together with option top($#_1$), the number of plotted models is the minimum between # and $#_1$.

equalwidths specifies that all plotted model bars have equal widths. By default, the bar widths are proportional to PMPs. This option may not be specified together with pmpnormalized.

pmpnormalized specifies that the model bar widths correspond to the normalized PMPs of the plotted models instead of PMPs. After normalization, the plotted range of cumulative PMPs (CPMP) will be between 0 and 1. This option may not be specified together with option equalwidths.

poscoefopts(*rbaropts*), negcoefopts(*rbaropts*), and zerocoefopts(*rbaropts*) control the look of bars for the positive, negative, and zero (not-included) coefficients, respectively. *rbaropts* are any options as defined in [G-3] *barlook_options*, and barwidth(#), mwidth, and msize(*markersizestyle*) as defined in [G-2] **graph twoway rbar**.

rbaropts control the look of all bars. *rbaropts* are any options as defined in [G-3] *barlook_options*, and barwidth(#), mwidth, and msize(*markersizestyle*) as defined in [G-2] **graph twoway rbar**.

Y axis, X axis, Titles, Legend, Overall

twoway_options are any of the options documented in [G-3] *twoway_options*, excluding by(). These include options for titling the graph (see [G-3] *title_options*) and for saving the graph to disk (see [G-3] *saving_option*).

The following option is available with bmagraph varmap but is not shown in the dialog box:

all specifies that all models, up to maxmodels(), be shown on the graph. This option may be useful when the number of visited models is small. This option may not be specified together with top().

Remarks and examples

In the context of BMA, a variable-inclusion map is a graphical representation of which variables or predictors are included in individual regression models. The plotted models are ranked by the highest PMP. Each predictor (with a PIP above 0.01) within a model is represented by a colored bar: blue if a predictor is included in a model with a positive coefficient, red if a coefficient is negative, and gray if a predictor is not included in the model. This map provides a graphical summary about the included predictors and their probabilities of being included across models and the signs of their corresponding regression coefficients. We can also evaluate the complexity of models from this map based on the number of included predictors.

▷ Example 1: Variable-inclusion map after BMA regression

Recall the performance dataset (Chatterjee and Hadi 2012, sec. 3.3) analyzed in example 1 of [BMA] **bmaregress**, where the employees' satisfaction with their supervisors, rating, is modeled by six potential predictors.

Let's fit a linear BMA regression to these data by using `bmaregress` and plot a variable-inclusion map by using `bmagraph varmap`.

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

. 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.
```

From the output, the model space is fully explored through enumeration. There is a total of $2^6 = 64$ models in the full model space. For a detailed interpretation of the above output, see example 1 of [BMA] **bmaregress**.

We use the `bmagraph varmap` command to construct a detailed variable-inclusion map for the fitted BMA model.

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


Variable-inclusion map

All 64 visited models shown.
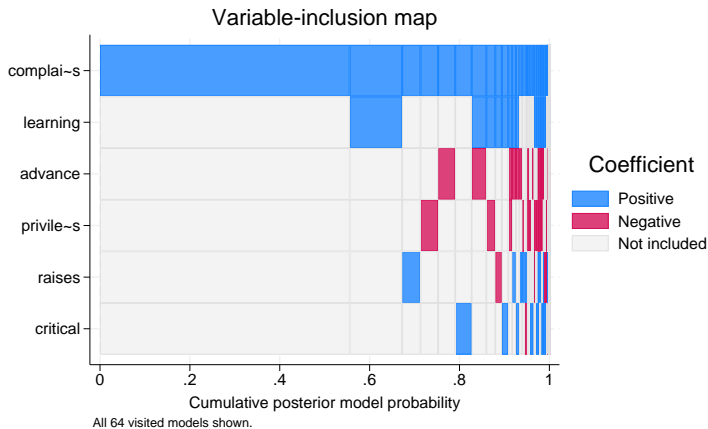
The plot displays all 64 models, which is less than the default 100. The models are ordered by their PMPs (from highest to lowest), and their CPMPs are displayed on the x axis. On the y axis, all six predictors are shown. Each pair of a model and a predictor is represented by a bar with a width that is proportional to the model's PMP. The bar is blue if a predictor is included in the model with a positive coefficient. It is red if a predictor is included in the model with a negative coefficient. It is gray if a predictor is not included in the model. In our example, because all models are included in the map, the plotted range of the CPMP x axis is between 0 and 1.

The first column corresponds to the model with the highest PMP of about 0.56 (the width of the bar), which includes only one predictor, `complaints`. This is not surprising because, from the output of `bmaregress`, predictor `complaints` has by far the highest PIP of 0.9997 of all predictors. We also see this on the map—the entire row for `complaints` has blue bars across almost all models.

The next highest-ranking model includes both `complaints` and `learning`, the predictor with the next highest PIP of 0.25. Also, it seems that the coefficients of these two predictors are always positive (blue bars) whenever they are included in a model. In other words, the two predictors appear to have a positive association with `rating`.

◁

▷ Example 2: Customizing the variable-inclusion map

The number of visited regression models can be large, and plotting all of them may become impractical. In such cases, we may use the `top(#)` option to show only the top # models with the highest PMPs. For illustration purposes, we will use `top(5)` to replicate the variable-inclusion summary table in example 1 of [BMA] **bmastats models**.

```
. bmagraph varmap, top(5)
Computing model probabilities ...
```


Variable-inclusion map

Top 5 models shown out of 64 visited.
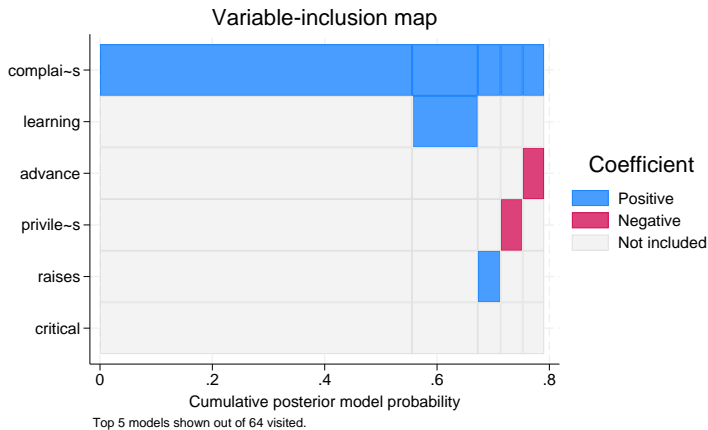
Because we are showing only 5 out of 64 models, the maximum range for the CPMP of the plotted models is less than 80%. The width of each model bar corresponds to the model's PMP value. You may specify the `pmpnormalized` option to rescale CPMP to span the range from 0 to 1.

As we mentioned earlier, variable `complaints` appears to be included in all models. However, the models with small PMPs may be represented by bars with widths too small to show the fill color. For those models, it may not be possible to determine from the map whether the predictor was included in the model. In that case, we may request that all bar widths be equal instead of being proportional to PMPs by specifying the `equalwidths` option.

```
. bmagraph varmap, equalwidths
Computing model probabilities ...
```


Variable-inclusion map

All 64 visited models shown.
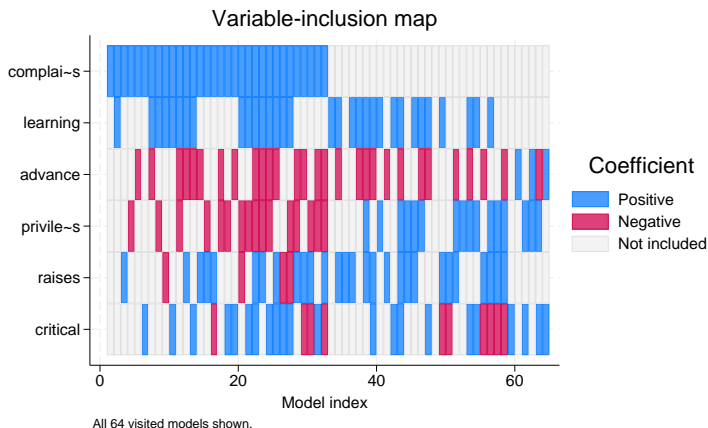
The model index (from 1 to 64) is now displayed on the x axis instead of the CPMP. It turns out that `complaints` is included only in the first 32 models with the highest PMPs with all 32 coefficients being positive.

For a BMA with many predictors and models, it may become necessary to restrict the list of the plotted predictors to those having PIP above a certain threshold to improve the readability of the graph. Below, we use the `pipcutoff()` option to limit the plotted predictors to only those that have PIP above 0.16.

```
. bmagraph varmap, equalwidths pipcutoff(.16)
Computing model probabilities ...
```


Variable-inclusion map

All 64 visited models shown. 4 predictors with PIP less than .16 not shown.
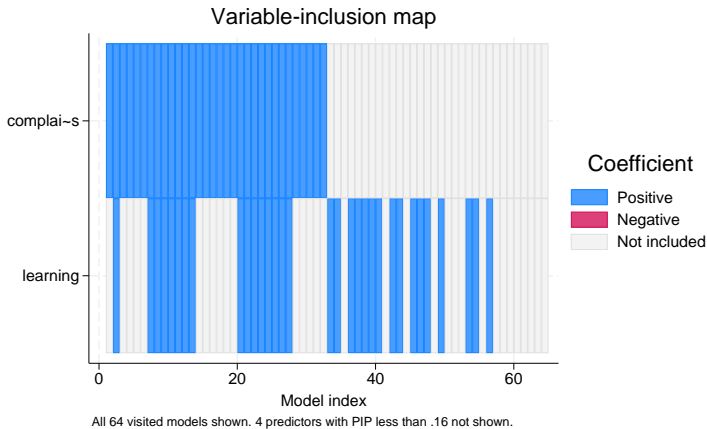
Based on the PIP column of the `bmaregress` output in example 1, we expect four predictors (`advance`, `privileges`, `raises`, and `critical`) to be excluded from the plot because their PIP < 0.16. `bmagraph varmap` also produces a note following the command specification to inform you about the number of predictors omitted from the plot.

◁

Methods and formulas

The widths of the bars corresponding to each model are proportional to PMPs, unless the `equalwidths` option is specified. For methods and formulas of PMPs, see *Posterior model probability* in *Methods and formulas* of [BMA] **bmaregress**.

The CPMP for a model j is a cumulative sum of the first jth highest PMPs.

Reference

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

Also see

[BMA] **bmastats models** — Model and variable-inclusion summaries after BMA regression

[BMA] **bmagraph** — Graphical summary for models and predictors after BMA regression

[BMA] **bmaregress** — Bayesian model averaging for linear regression

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

[BMA] **Glossary**

Description

bmapredict computes Bayesian model averaging (BMA) predictions using current estimation results produced by bmaregress. Certain predictions such as credible intervals (CrIs) require that you also run bmacoefsample with the saving() option after bmaregress.

bmareps generates a random subset of Markov chain Monte Carlo (MCMC) replicates of simulated outcomes from the entire MCMC sample and stores them as new variables in the current dataset. This command is useful for checking model fit. bmareps requires that you first save the MCMC sample of model parameters by using the saving() option with the bmacoefsample command.

Quick start

Compute and save posterior predictive means in a variable pmean1 after a BMA regression with a fixed g

```
bmaregress y x1–x12
bmapredict pmean1, mean
```

Compute and save posterior predictive means in a variable pmean2 after a BMA regression with a random g; requires that bmacoefsample be run after bmaregress

```
bmaregress y x1–x12, gprior(robust) saving(bmamodel1)
bmacoefsample, saving(bmacoef1)
bmapredict pmean2, mean
```

Compute 95% CrIs of simulated outcome, and save them in variables cri_l and cri_u; requires that bmacoefsample be run after bmaregress with a fixed or random g

```
bmaregress y x1–x12, saving(bmamodel2)
bmacoefsample, saving(bmacoef2)
bmapredict cri_l cri_u, cri
```

Same as above, but simulate an entire predictive outcome dataset, and save it in bmaypred.dta

```
bmapredict, saving(bmaypred)
```

Equivalent to the above, but use bayespredict and save results in bayesypred.dta

```
bayespredict {_ysim}, saving(bayesypred)
```

Write your own program yfuncprog to compute a more complicated function of a simulated outcome, and use bayespredict to simulate its posterior distribution

```
bayespredict (yfunc: @yfuncprog {_ysim}), saving(bayesyfuncpred)
```

Generate three replicates from the posterior predictive distribution, save them as predy1, predy2, and predy3, and specify random-number seeds for reproducibility

```
bmaregress y x1–x12, saving(bmamodel3) rseed(123)
bmacoefsample, saving(bmacoef3) rseed(982)
bmareps predy*, nreps(3) rseed(23672)
```

Generate log predictive-scores (always available and do not require bmacoefsample)

> bmapredict lps, lps

Menu

Statistics > Bayesian model averaging > Predictions

Syntax

Syntax is presented under the following headings:

Compute analytical posterior predictive summaries (available only with fixed g)

Analytical posterior mean

> bmapredict [*type*] *newvar* [*if*] [*in*], mean [<u>analytic</u>]

Analytical posterior standard deviation

> bmapredict [*type*] *newvar* [*if*] [*in*], std [<u>analytic</u>]

Analytical posterior predictive means and standard deviations are available after bmaregress only for models with a fixed g, in which case the analytic option is implied.

Compute MCMC-sample posterior predictive summaries (available after bmacoefsample)

Posterior mean of simulated outcome

> bmapredict [*type*] *newvar* [*if*] [*in*], mean [<u>mcmcsample</u> *meanopts simopts*]

Posterior median or posterior standard deviation of simulated outcome

> bmapredict [*type*] *newvar* [*if*] [*in*], median|std [<u>mcmcsample</u> *simopts*]

Credible intervals of simulated outcome

> bmapredict [*type*] *newvar_l* *newvar_u* [*if*] [*in*], cri [<u>mcmcsample</u> *criopts simopts*]

MCMC-sample posterior predictive summaries are available only after bmacoefsample with the saving() option is used after bmaregress. For models with a random g, option mcmcsample is implied for mean and std. This option is always implied for median and cri.

Compute predictions of simulated outcome (available after bmacoefsample)

Simulate outcome and save in a file

> bmapredict [*if*] [*in*], <u>sav</u>ing(*filename*[, replace]) [*simopts*]

Also see *Syntax* in [BAYES] **bayespredict**, particularly for predictions of functions of a simulated outcome.

Predictions for a simulated outcome are available only after bmacoefsample with the saving() option is used after bmaregress.

Generate a subset of MCMC replicates of simulated outcome (available after bmacoefsample)

> bmareps [*type*] *newrepspec* [*if*] [*in*], nreps(#) [*simopts*]

newrepspec is *newvar* with nreps(1) for a single replicate and *stub** with nreps(#), where # is greater than 1, for multiple replicates.

Replicates of a simulated outcome are available only after bmacoefsample with the saving() option is used after bmaregress.

Compute log predictive-scores

> bmapredict [*type*] *newvar* [*if*] [*in*], lps

simopts	Description
Simulation	
rseed(#)	random-number seed
[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

meanopts	Description
Main	
mcse(*newvar*)	create *newvar* containing Monte Carlo standard errors (MCSEs)
Advanced	
batch(#)	specify length of block for batch-means calculations; default is batch(0)
corrlag(#)	specify maximum autocorrelation lag; default varies
corrtol(#)	specify autocorrelation tolerance; default is corrtol(0.01)

criopts	Description
Main	
<u>cl</u>evel(#)	set CrI level; default is clevel(95)
hpd	calculate highest posterior density (HPD) CrIs instead of the default equal-tailed CrIs

Options

Options are presented under the following headings:

Options for analytical posterior predictive summaries

┌─ Main ───

mean with analytic calculates analytical posterior predictive means of an outcome and stores them as a new variable in the current dataset. For models with fixed g, option analytic is implied.

std with analytic calculates analytical posterior predictive standard deviations of an outcome and stores them as a new variable in the current dataset. For models with fixed g, option analytic is implied.

analytic specifies to use analytical expressions to calculate posterior predictive means or standard deviations. This option is implied for models with fixed g, and it is not available with random g. Analytical computations are available directly after bmaregress.

Options for MCMC-sample posterior predictive summaries

┌─ Main ───

mean with mcmcsample calculates MCMC-sample posterior predictive means of a simulated outcome and stores them as a new variable in the current dataset. For models with random g, option mcmcsample is implied.

median calculates posterior medians of a simulated outcome and stores them as a new variable in the current dataset. Option mcmcsample is always implied with median.

std with mcmcsample calculates MCMC-sample posterior predictive standard deviation of a simulated outcome and stores them as a new variable in the current dataset. For models with random g, option mcmcsample is implied.

cri calculates CrIs for a simulated outcome and stores the corresponding lower and upper bounds in two new variables in the current dataset. Option mcmcsample is always implied with cri.

mcmcsample specifies to estimate posterior predictive means or standard deviations from the MCMC sample of model parameters instead of using analytical expressions. This option is the only choice for models with random g and thus is implied with random g.

mcse(*newvar*) is for use in a combination with option mean. It adds *newvar* of storage type *type* containing MCSEs for the posterior means of a simulated outcome.

clevel(#) specifies the credible level, as a percentage, for equal-tailed and HPD CrIs. The default is clevel(95) or as set by [BAYES] **set clevel**. This option requires that cri also be specified.

hpd calculates the HPD CrIs instead of the default equal-tailed CrIs. This option requires that cri also be specified.

> Simulation

rseed(#) sets the random-number seed. This option can be used to reproduce results. rseed(#) is
   equivalent to typing set seed # prior to calling bmapredict; see [R] **set seed**.

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(#)).

> Advanced

The advanced options are available only with option mean.

batch(#) specifies the length of the block for calculating batch means and an MCSE using batch means.
   The default is batch(0), which means no batch calculations. When batch() is not specified, the
   MCSE is computed using effective sample sizes instead of batch means. batch() may not be combined
   with corrlag() or corrtol().

corrlag(#) specifies the maximum autocorrelation lag used for calculating effective sample sizes. The
   default is min$\{500, \text{mcmcsize}()/2\}$. The total autocorrelation is computed as the sum of all lag-k
   autocorrelation values for k from 0 to either corrlag() or the index at which the autocorrelation
   becomes less than corrtol() if the latter is less than corrlag(). Options corrlag() and batch()
   may not be combined.

corrtol(#) specifies the autocorrelation tolerance used for calculating effective sample sizes. The
   default is corrtol(0.01). For a given model parameter, if the absolute value of the lag-k autocorre-
   lation is less than corrtol(), then all autocorrelation lags beyond the kth lag are discarded. Options
   corrtol() and batch() may not be combined.

Options for predictions of simulated outcome

> Main

saving(*filename*[, replace]) saves the simulated outcome in *filename*.dta. It also saves auxiliary
   estimation results in *filename*.ster, which is accessible by specifying estimates use *filename*. The
   replace option specifies to overwrite *filename*.dta and *filename*.ster if they exist.

> Simulation

rseed(#) sets the random-number seed. This option can be used to reproduce results. rseed(#) is
   equivalent to typing set seed # prior to calling bmapredict; see [R] **set seed**.

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(#)).

Options for bmareps

___Main___

nreps(#) specifies the number of MCMC replicates of simulated outcome to be drawn at random from
the entire sample of MCMC replicates. # must be an integer between 1 and the MCMC sample size,
inclusively. The generated replicates are stored as new variables in the current dataset. For a single
replicate, nreps(1), you specify one new variable name. For multiple replicates, you specify a *stub**,
in which case the replicates will be stored in variables *stub*1, *stub*2, ..., *stub*R, where R is the number
of replicates specified in nreps().

___Simulation___

rseed(#) sets the random-number seed. This option can be used to reproduce results. rseed(#) is
equivalent to typing set seed # prior to calling bmareps; see [R] **set seed**.

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(#)).

Option for log predictive-scores

___Main___

lps calculates log predictive-scores and stores them as a new variable in the current dataset.

Remarks and examples

Bayesian predictions include a simulated outcome, which is a sample from the posterior predictive
distribution of the fitted Bayesian model, and their functions; see [BAYES] **bayespredict** for details. In
the context of BMA, this posterior predictive distribution also accounts for model uncertainty. Often of
interest are summaries of this distribution such as means, medians, standard deviations, and CrIs.

bmapredict simulates an outcome from its BMA posterior predictive distribution and, optionally,
saves it in a dataset specified in the saving() option. It also computes BMA posterior predictive sum-
maries and stores them as new variables in the current dataset when you specify one of options mean,
median, std, or cri.

For BMA models with a fixed g, you can compute posterior predictive means and standard deviations
analytically or, if an MCMC sample of model parameters is available, from the simulated MCMC predic-
tive sample. The analytical computations are always the default whenever they are available; that is, the
analytic option is implied with the mean and std options with a fixed g. For BMA models with a ran-
dom g, analytical computations are not available, and thus the MCMC-sample computations are the only
choice—the mcmcsample option is implied with a random g with mean and std. For other summaries
such as posterior predictive medians and CrIs (options median and cri), only MCMC-sample estimates
are available regardless of whether parameter g is fixed or random; that is, the mcmcsample option is
always implied.

MCMC-sample computations require that you run the `bmacoefsample` command with the `saving()` option following `bmaregress` to generate and save a posterior sample of model parameters, which will be used to produce an MCMC sample of a simulated outcome from its posterior predictive distribution; see *Methods and formulas*.

Once a posterior MCMC sample of model parameters is available (by running `bmacoefsample`), you can also use `bayespredict`, one of the standard Bayesian postestimation commands, to obtain BMA predictions. This command is useful, for instance, if you need to compute any functions of the simulated outcome; see [BAYES] **bayespredict**. In fact, typing

```
. bmapredict, saving(bmaysim)
```

is equivalent to typing

```
. bayespredict {_ysim}, saving(bmaysim)
```

`bmareps` generates a random subset of MCMC replicates of a simulated outcome from the entire MCMC sample and stores them as new variables in the current dataset. This command is useful for checking model fit. It is similar to the `bayesreps` command as described in [BAYES] **bayespredict**.

You can also use the `lps` option to generate log predictive-scores. This can also be done by using `bmastats lps, generate()` and is provided with `bmapredict` for completeness; see [BMA] **bmastats lps**.

`bmapredict` (as well as `bmareps`) is a stochastic command; use the `rseed()` option for reproducibility. Remember that to have a complete reproducibility, you will also need to specify the `rseed()` option with `bmacoefsample` and `bmaregress`, which are also stochastic commands. If you do not need to reproduce each individual step, a better approach is to set a random-number seed once at the beginning of your analysis by using `set seed` (see [R] **set seed**).

▷ Example 1: Prediction for BMA linear regression with fixed g

Consider the Ames housing dataset (De Cock 2011), `ameshouses`, also used in a Kaggle competition, which describes residential houses sold in Ames, Iowa, between 2006 and 2010. It contains about 80 housing (and related) characteristics such as home size, amenities, and location. This dataset is often used for building predictive models for home sale price, `saleprice`. We will use BMA to model home sale price and evaluate its predictive performance. Here we will use just a few predictors to demonstrate some of the `bmapredict` features.

We split the dataset into two subsamples in the ratio of 2 to 1. The first subsample will be used for fitting the models and the second for testing. We use the log-transformed sale price, `lsaleprice`, as our outcome variable. And we generate a new variable, `age`, to record the age of home in years at the time of sale.

```
. use https://www.stata-press.com/data/r19/ameshouses
(Ames house data)
. splitsample, generate(sample) nsplit(2) split(2 1) rseed(18)
. generate double lsaleprice = log(saleprice)
. generate age = yrsold - yearbuilt + 1
```

We fit a BMA linear model for `lsaleprice` and use just a few of the available predictors for demonstration purposes. The type of sale, `saletype`, is included as a factor variable. The overall home quality and condition are included as ordinal predictors. We use the default specifications for model and *g* priors. The total number of predictors, 29, makes model enumeration impractical, so the MCMC model composition (MC3) sampling is used instead. And we specify the `rseed()` option for reproducibility.

```
. bmaregress lsaleprice i.saletype overallqual overallcond age
> paveddrive grlivarea totalbsmtsf garagearea garagecars
> bedroom kitchenabvgr kitchenqual bsmtqual exterqual extercond
> fullbath halfbath fireplaces wooddecksf lotarea
> if sample == 1, rseed(18)
Burn-in ...
Simulation ...
Computing model probabilities ...
```

Bayesian model averaging			No. of obs	=	973
Linear regression			No. of predictors	=	27
MC3 sampling			Groups	=	27
			Always	=	0
			No. of models	=	454
			For CPMP >= .9	=	85
Priors:			Mean model size	=	11.435
Models: Beta-binomial(1, 1)			Burn-in	=	2,500
Cons.: Noninformative			MCMC sample size	=	10,000
Coef.: Zellner's g			Acceptance rate	=	0.1410
g: Benchmark, g = 973			Shrinkage, g/(1+g)	=	0.9990
sigma2: Noninformative			Mean sigma2	=	0.027

Sampling correlation = 0.9690

lsaleprice	Mean	Std. dev.	Group	PIP
overallqual	.084876	.0069513	9	1
overallcond	.0624909	.0053748	10	1
age	-.0031076	.0002871	11	1
grlivarea	.0001866	.0000233	13	1
garagecars	.091476	.0099425	16	1
fireplaces	.0570484	.0100975	25	1
wooddecksf	.0002054	.0000468	26	1
totalbsmtsf	.0000683	.0000162	14	.99776
kitchenqual	-.0276135	.0105654	19	.94116
lotarea	1.86e-06	8.67e-07	27	.89429
bedroomabvgr	.0138705	.013693	17	.56959
saletype				
New	.0184764	.0285989	6	.34325
bsmtqual	-.0047449	.0080652	20	.30144
saletype				
CWD	-.0148399	.0723153	1	.056743
paveddrive	.0020311	.0098775	12	.056586
extercond	.000576	.0029977	22	.051934
fullbath	.0005671	.0040245	23	.034588
saletype				
WD	-.0004454	.0052115	8	.032251
kitchenabvgr	-.0007089	.0061078	18	.022655
saletype				
Oth	.0021639	.0228432	7	.022015
halfbath	.0001984	.002301	24	.021049
exterqual	-.0001345	.0020265	21	.020406

saletype				
Con	.0018756	.0258891	2	.017179
ConLI	−.0011395	.0225588	4	.015548
garagearea	5.36e−08	6.54e−06	15	.014047
saletype				
ConLD	.0005622	.0100442	3	.013859
Always				
_cons	10.74662	.0676172	0	1

Note: Coefficient posterior means and std. dev. estimated from 454 models.
Note: Default priors are used for models and parameter *g*.
Note: 1 predictor with PIP less than .01 not shown.

The bmaregress commands visited a total of 454 models with an average size of about 11. There are 11 predictors with posterior inclusion probability (PIP) greater than 0.5. Whether the home is new and the height of the basement (bsmtqual) appear to impact the home prices, too, with respective PIPs of 0.34 and 0.3. The other predictors have PIPs below 10%.

Let's use the bmapredict command to compute the posterior mean predictions for lsaleprice on the test subsample, then compute the mean squared errors of these predictions.

```
. bmapredict pmean1 if sample == 2, mean
note: computing analytical posterior predictive means.
. generate double sqerr1 = (lsaleprice-pmean1)^2
(973 missing values generated)
. summarize sqerr1
```

Variable	Obs	Mean	Std. dev.	Min	Max
sqerr1	487	.0177806	.0354926	1.24e−07	.5608268

The estimated mean squared error for the model is 0.018.

Parameter *g* is fixed in our example, so bmapredict computed the analytical posterior predictive mean. For comparison, we can compute one based on an MCMC sample. To do this, we first need to simulate a posterior sample of model parameters by using bmacoefsample. To save time, we use a smaller MCMC sample of 1,000.

```
. bmaregress, saving(hprices_bmareg1)
note: file hprices_bmareg1.dta saved.
. bmacoefsample, saving(hprices_sample1) rseed(18) mcmcsize(1000) nodots
Simulation ...
file hprices_sample1.dta saved.
```

We then specify the `mcmcsample` option with `bmapredict` to compute the MCMC-sample posterior predictive mean.

```
. bmapredict pmean1_s if sample == 2, mean mcmcsample rseed(18)
note: computing posterior predictive means using simulation.
Computing predictions ...
. summarize pmean1 pmean1_s
```

Variable	Obs	Mean	Std. dev.	Min	Max
pmean1	487	12.03638	.3525371	11.10996	12.97869
pmean1_s	487	12.03606	.3524122	11.11674	12.98349

The two predicted means are very similar, as is expected provided the MCMC sample size is sufficiently large and the MCMC converged.

Now that we have an MCMC sample of model parameters, we can, for example, generate predictive CrIs for the test subsample and check their coverage.

```
. bmapredict cril1 criu1 if sample == 2, cri rseed(18)
note: computing credible intervals using simulation.
Computing predictions ...
. generate cover1 = lsaleprice < criu1 & lsaleprice > cril1 if sample == 2
(973 missing values generated)
. summarize cover1
```

Variable	Obs	Mean	Std. dev.	Min	Max
cover1	487	.9774127	.1487364	0	1

The coverage of the simulated 95% equal-tailed predictive CrIs is 98%, slightly larger than expected.

◁

▷ Example 2: Prediction for BMA linear regression with random g

The usage of `bmapredict` after a BMA linear regression with random g is slightly different. To illustrate the prediction workflow, let's fit the same BMA regression as in example 1 but use the random `hyperg(3)` prior for parameter g here.

```
. bmaregress lsaleprice i.saletype overallqual overallcond age
> paveddrive grlivarea totalbsmtsf garagearea garagecars
> bedroom kitchenabvgr kitchenqual bsmtqual exterqual extercond
> fullbath halfbath fireplaces wooddecksf lotarea
> if sample == 1, gprior(hyperg 3) rseed(18)
Burn-in ...
Simulation ...
Computing model probabilities ...
```

Bayesian model averaging		No. of obs	=	973
Linear regression		No. of predictors	=	27
MC3 and adaptive MH sampling		Groups	=	27
		Always	=	0
		No. of models	=	678
		For CPMP >= .9	=	319
Priors:		Mean model size	=	12.622
Models: Beta-binomial(1, 1)		Burn-in	=	2,500
Cons.: Noninformative		MCMC sample size	=	10,000
Coef.: Zellner's g		Acceptance rate	=	0.5261
g: Hyper-g(3)				
sigma2: Noninformative		Mean sigma2	=	0.027

Sampling correlation = 0.8885

lsaleprice	Mean	Std. dev.	Group	PIP
overallqual	.084121	.0070143	9	1
overallcond	.0626002	.0054331	10	1
age	−.0030242	.000306	11	1
grlivarea	.0001802	.0000232	13	1
garagecars	.0906061	.0105574	16	1
fireplaces	.0576947	.0101318	25	1
wooddecksf	.0002048	.000047	26	1
totalbsmtsf	.0000691	.0000165	14	.9968
kitchenqual	−.0278064	.0102489	19	.9515
lotarea	1.89e−06	8.37e−07	27	.9103
bedroomabvgr	.0182072	.013439	17	.7243
bsmtqual	−.0076919	.0091544	20	.4843
saletype				
New	.022116	.0305646	6	.4058
paveddrive	.0069869	.0175817	12	.1869
saletype				
CWD	−.0424225	.1177667	1	.161
extercond	.0011747	.0041903	22	.1077
garagearea	7.44e−07	.0000155	15	.0755
kitchenabvgr	−.0024381	.0112473	18	.0752
saletype				
ConLD	.0036524	.0242701	3	.0719
fullbath	.0010832	.0055586	23	.0715
saletype				
ConLw	.000148	.0221823	5	.0701
Con	.0078374	.0522285	2	.0688
WD	−.0002908	.0080338	8	.0687
halfbath	.0005647	.0038594	24	.0544
saletype				
ConLI	−.0038273	.0413487	4	.0521
Oth	.0039455	.0312492	7	.0437
exterqual	−.0003791	.0032394	21	.0416
Always				
_cons	10.75005	.0707137	0	1

Note: Coefficient posterior means and std. dev. estimated from 678 models.
Note: Default prior is used for models.

	Mean	Std. dev.	MCSE	Median	Equal-tailed [95% cred. interval]	
g	425.3529	206.9134	6.15565	375.5357	179.5198	963.0143
Shrinkage	.9971914	.0011648	.000033	.9973442	.9944604	.9989627

The number of visited models is larger, 678 versus 454, as is the posterior mean model size, 13 versus 11.

Unlike for BMA models with fixed g, the mean and standard deviations cannot be computed analytically with random g. All posterior predictive summaries now need to be estimated from an MCMC predictive sample of lsaleprice. As in example 1 for MCMC-sample mean and CrIs, we first need to run bmacoefsample to obtain a posterior sample of model parameters.

```
. bmaregress, saving(hprices_bmareg2)
note: file hprices_bmareg2.dta saved.
. bmacoefsample, saving(hprices_sample2) rseed(18) mcmcsize(1000) nodots
Simulation ...
file hprices_sample2.dta saved.
```

We compute posterior mean predictions and the mean squared error for these predictions. Because our BMA model uses random g, the mcmcsample option is automatically implied.

```
. bmapredict pmean2 if sample == 2, mean rseed(18)
note: computing posterior predictive means using simulation; option mcmcsample
      implied.
Computing predictions ...
. generate double sqerr2 = (lsaleprice-pmean2)^2
(973 missing values generated)
. summarize sqerr2
```

Variable	Obs	Mean	Std. dev.	Min	Max
sqerr2	487	.0180237	.0358707	7.16e-08	.5625565

The estimated mean squared error for this model is 0.018.

We can also compute 95% equal-tailed predictive CrIs and their coverage in this example.

```
. bmapredict cril2 criu2 if sample == 2, cri rseed(18)
note: computing credible intervals using simulation.
Computing predictions ...
. generate cover2 = lsaleprice < criu2 & lsaleprice > cril2 if sample == 2
(973 missing values generated)
. summarize cover2
```

Variable	Obs	Mean	Std. dev.	Min	Max
cover2	487	.9794661	.1419635	0	1

The estimated coverage is 98%.

The predictive performances of the two models, with fixed and random g, are similar in terms of the mean squared error and CrI coverage.

◁

▷ Example 3: Predictions by using bayespredict

After generating a sample of model parameters by using bmacoefsample, you can also use the more advanced functionality of the bayespredict command to compute BMA predictions. For instance, let's generate a posterior sample of mean squared prediction errors, which are functions of a simulated outcome. There is no automatic way to compute this statistic by using any of the available options of bmapredict or bayespredict, so we need to write a program that computes it.

```
. program psqerr2
  1.            args mse ysim
  2.            local touse $BAYESPR_touse
  3.            local y $BAYESPR_extravars
  4.            tempvar v
  5.            generate double `v' = (`y'-`ysim')^2 if `touse'
  6.            summarize `v', meanonly
  7.            scalar `mse' = r(mean)
  8. end
```

The `psqerr2` program has two arguments: `mse` to store the mean squared error value and `ysim` that will contain the simulated outcome generated by `bayespredict` for each MCMC iteration. To compute the squared errors, we will need the actual outcome values, labeled as `y` in the program, which will be passed to the program as an extra variable. The program then uses the observed outcome values and the simulated values to compute the squared errors, using the proper estimation sample `touse`. Finally, it saves the mean of the squared errors in the scalar `mse`.

Next, we call `bayespredict` by using its syntax for user-defined programs; see [BAYES] **bayespredict**. We provide the name of our program and the vector of simulations of the outcome, {_ysim}. We also provide the name of the observed outcome variable, `lsaleprice`, in the `extravars()` option. And we save our simulated squared errors in the `bmapred2` dataset for later use.

```
. bayespredict (sqerr2:@psqerr2 {_ysim}, extravars(lsaleprice)) if sample == 2,
> saving(bmapred2) rseed(18)

Computing predictions ...

file bmapred2.dta saved.
file bmapred2.ster saved.
```

We can use the `bayesstats summary` command (see [BAYES] **bayesstats summary**) to compute, for instance, the posterior mean of the mean squared error estimates.

```
. bayesstats summary {sqerr2} using bmapred2
```

Posterior summary statistics   MCMC sample size =   1,000

					Equal-tailed	
	Mean	Std. dev.	MCSE	Median	[95% cred. interval]	
sqerr2	.0461	.0030704	.000097	.0460156	.0402083	.0525146

The posterior mean is about 0.046. It is, as expected, larger than the mean squared error calculated in example 2, where we used the predicted mean outcome to compute the errors. In practice, there is no reason to compute the mean squared error via simulation, as we showed in this example. We did this purely to demonstrate a more advanced usage of `bayespredict`.

◁

Methods and formulas

Methods and formulas are presented under the following headings:

> *BMA predictions for the linear model*
> *Analytic predictive mean and standard deviation for fixed g*
> *Simulating outcome from its posterior predictive distribution*

BMA predictions for the linear model

We consider predictions for a BMA linear model, which is fit by the bmaregress command. Consider a $(p + 1) \times 1$ vector of regression coefficients and the intercept in the original metric, $\boldsymbol{\theta} = (\boldsymbol{\beta}', \alpha)'$ and error variance σ^2.

Let $\mathbf{X}$ be a $n \times p$ matrix of predictor values used to fit the BMA model, $\mathbf{X}^*$ be a $q \times p$ matrix of new predictor values, $\mathbf{y}^*$ be a vector of posterior predictions of size q, and $\mathbf{X_1^*} = [\mathbf{X}^* \ \mathbf{1}_q]$, where $\mathbf{1}_q$ is a $q \times 1$ vector of ones. In some cases, certain statistics of $\mathbf{y}^*$, such as mean and standard deviation, can be computed directly; others require simulations from the predictive distribution. We consider these two cases separately.

Analytic predictive mean and standard deviation for fixed g

With a fixed g, the posterior predictive mean and variance of $\mathbf{y}^*$ can be computed analytically from the posterior means of $\boldsymbol{\theta} = (\boldsymbol{\beta}', \alpha)'$ and σ^2 and posterior variance of $\boldsymbol{\theta}$ as follows,

$$E(\mathbf{y}^*|\mathbf{y}, \mathbf{X}, \mathbf{X_1^*}) = \mathbf{X_1^*} E(\boldsymbol{\theta}|\mathbf{y})$$

$$\text{Var}(\mathbf{y}^*|\mathbf{y}, \mathbf{X}, \mathbf{X_1^*}) = E(\sigma^2|\mathbf{y})\mathbf{I}_q + \mathbf{X_1^*} \text{Var}(\boldsymbol{\theta}|\mathbf{y})\mathbf{X_1^*}'$$

where $E(\boldsymbol{\theta}|\mathbf{y})$, $E(\sigma^2|\mathbf{y})$, and $\text{Var}(\boldsymbol{\theta}|\mathbf{y})$ are given, respectively, by (11), (13), and (12) in *Methods and formulas* of [BMA] **bmaregress**.

The posterior mean $E(\mathbf{y}^*|\mathbf{y}, \mathbf{X}, \mathbf{X_1^*})$ is computed with options mean and analytic. The posterior standard deviation of $\mathbf{y}^*$ is the square root of the diagonal of $\text{Var}(\mathbf{y}^*|\mathbf{y}, \mathbf{X}, \mathbf{X_1^*})$. It is computed with options std and analytic.

Simulating outcome from its posterior predictive distribution

The posterior predictive distribution of $\mathbf{y}^*$ can be estimated via MCMC simulations. Suppose we have a sample $\{\boldsymbol{\beta}_t, \alpha_t, \sigma_t^2\}_{t=1}^T$ of regression coefficients, intercepts, and error variances from a BMA linear model. Such posterior samples are generated by the bmacoefsample command (see [BMA] **bmacoefsample**). Here T is the MCMC sample size as stored in e(mcmcsize2) by bmacoefsample.

For each t, an error vector $\boldsymbol{\epsilon}_t^*$ and outcome $\mathbf{y}_t^*$ are simulated according to

$$\boldsymbol{\epsilon}_t^*|\boldsymbol{\beta}_t, \sigma_t^2, \mathbf{X} \ \sim \ N_q(\mathbf{0}, \sigma_t^2 \mathbf{I}_q)$$
$$\mathbf{y}_t^*|\mathbf{X}, \mathbf{X_1^*} \ = \ \alpha_t + \mathbf{X}^* \boldsymbol{\beta}_t + \boldsymbol{\epsilon}_t^*$$

The result is a predictive sample $\{\mathbf{y}_t^*\}_{t=1}^T$, which is conditionally independent given $\mathbf{X}^*$ and $\mathbf{X}$.

If you use bmapredict, saving(*filename*) on the same dataset used to fit the model, the simulated outcome $\{\tilde{\mathbf{y}}\}_{t=1}^T$ of size $n \times T$ will be saved in *filename*.dta with the observed data matrix $\mathbf{X}$ being used in place of $\mathbf{X}^*$ in the above. This is sometimes called a replication sample. Posterior summaries corresponding to options mean mcmcsample, median, std mcmcsample, and cri are computed based on the simulated outcome $\{\tilde{\mathbf{y}}\}_{t=1}^T$ as described in *Methods and formulas* of [BAYES] **bayespredict**.

bmareps generates a subset of replicates $\{\tilde{\mathbf{y}}\}_{t=1}^{T_{\text{reps}}}$, where T_{reps} is determined by the nreps(#) option.

For the computation of log predictive-scores (option `lps`), see *Methods and formulas* of [BMA] **bmastats lps**.

Reference

De Cock, D. 2011. Ames, Iowa: Alternative to the Boston housing data as an end of semester regression project. *Journal of Statistics Education* 19(3). https://doi.org/10.1080/10691898.2011.11889627.

Also see

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

[BMA] **bmaregress** — Bayesian model averaging for linear regression

[BMA] **bmacoefsample** — Posterior samples of regression coefficients

[BMA] **Glossary**

[BAYES] **bayespredict** — Bayesian predictions

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

Description

The following `bmastats` subcommands are available after `bmaregress`:

Command	Description
bmastats models	posterior model and variable-inclusion summaries
bmastats msize	model-size summary
bmastats pip	posterior inclusion probabilities (PIPs) for predictors
bmastats jointness	jointness measures for predictors
bmastats lps	log predictive-score

Remarks and examples

See [BMA] **BMA postestimation** for a short introduction to Bayesian model averaging (BMA) postestimation.

The `bmastats models` command reports models with high posterior model probability and their included predictors. You can also use it to explore cumulative posterior model probability. See [BMA] **bmastats models**.

The `bmastats msize` command provides model-size summaries for the prior and posterior model sizes; see [BMA] **bmastats msize**.

The `bmastats pip` command reports PIP for predictors. In the context of BMA, PIP is used to measure the importance of one predictor relative to the others. You can use this command to report PIP for specific predictors. See [BMA] **bmastats pip**.

The `bmastats jointness` command provides various jointness measures, which examine the tendency of pairs of predictors to be included together, separately, or independently in the models; see [BMA] **bmastats jointness**.

The `bmastats lps` command computes log predictive-scores, which can be used to compare model goodness of fit and model predictive performance; see [BMA] **bmastats lps**.

Also see

[BMA] **bmaregress** — Bayesian model averaging for linear regression

[BMA] **bmacoefsample** — Posterior samples of regression coefficients

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

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

[BMA] **Glossary**

Description

bmastats jointness reports various jointness measures between pairs of predictors after the bmaregress command. These jointness measures assess the degree of inclusion dependency between two predictors across the models visited by bmaregress.

Quick start

Compute all available jointness measures for predictors x1 and x2

    bmastats jointness x1 x2

Compute the default Doppelhofer–Weeks jointness measure for predictors x1 through x4

    bmastats jointness x1-x4

Same as above, but compute Ley–Steel type 2 jointness measure

    bmastats jointness x1-x4, lsteel2

Same as above, but compute modified Yule's Q jointness measure

    bmastats jointness x1-x4, yuleqm frequency

Compute all available jointness measures

    bmastats jointness x1-x4, all

Menu

Statistics > Bayesian model averaging > Jointness measures

169

Syntax

> `bmastats jointness` *varlist* [, *jointopts* `frequency`]

jointopts	Description
<u>all</u>	all available jointness measures; default with two predictors
<u>dw</u>eeks	Doppelhofer–Weeks measure; default with more than two predictors
<u>lsteel1</u>	Ley–Steel type 1 measure
<u>lsteel2</u>	Ley–Steel type 2 measure
<u>yuleq</u>	Yule's Q measure
yuleqm	modified Yule's Q measure; available only with sampling and requires option frequency
ls1	synonym for lsteel1
ls2	synonym for lsteel2
yq	synonym for yuleq
yqm	synonym for yuleqm

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

Options

> [Main]

`all` computes and reports all available jointness measures. This is the default for two predictors. If more than two predictors are specified, one table for each measure is reported.

`dweeks` computes and reports the Doppelhofer–Weeks measure. This is the default when more than two predictors are specified. `dw` is a synonym for `dweeks`.

`lsteel1` computes and reports the Ley–Steel type 1 measure. `ls1` is a synonym for `lsteel1`.

`lsteel2` computes and reports the Ley–Steel type 2 measure. `ls2` is a synonym for `lsteel2`.

`yuleq` computes and reports Yule's Q measure. `yq` is a synonym for `yuleq`.

`yuleqm` computes and reports the modified Yule's Q measure. `yqm` is a synonym for `yuleqm`. This measure is available only with sampling and requires that you also specify option `frequency`.

`frequency` specifies that frequency estimates of posterior model probabilities (PMPs) based on Markov chain Monte Carlo be used in computations. These estimates are available only with sampling methods, when MCMC sample is available. That is, they are not available with model enumeration. With a fixed g, analytical PMPs are used in computations by default. With a random g, analytical PMPs are not available, and thus frequency PMPs are used (option `frequency` is implied).

Remarks and examples

Remarks are presented under the following headings:

> *Jointness as a measure of variable-inclusion dependence*
> *Example: Jointness of growth determinants*

Jointness as a measure of variable-inclusion dependence

In Bayesian model averaging (BMA), marginal inferential measures such as posterior inclusions probabilities (PIPs) describe the importance of a single predictor in explaining the outcome after accounting for model uncertainty. However, joint inferential measures that explore the interrelations between predictors are often of interest too. For instance, we might want to know whether the importance of one predictor is affected by the presence of another across all the models. BMA provides various jointness measures to address this.

BMA defines jointness measures with respect to the joint posterior distribution of inclusion of predictors over the model space. Specifically, bivariate jointness measures are used to describe the inclusion patterns for pairs of predictors. That is, bivariate jointness assesses the tendency of two predictors to be included together or exclusively across the model space. In what follows, we will imply "bivariate jointness measures" when we say "jointness measures".

In general, jointness means that the two predictors are *complements*; that is, they add additional explanatory power for the outcome when included together. Thus, these predictors tend to be included together in a model with high PIP. Disjointness means that the two predictors are *substitutes*; that is, when one predictor is in a model, the inclusion of the other does not add any additional explanatory power for the outcome. These predictors tend to be excluded whenever the other one is in a model.

BMA literature introduces various jointness measures. The working paper by Doppelhofer and Weeks (2006) was the first to introduce a jointness measure in the context of BMA. Later, Doppelhofer and Weeks (2009) introduced a modified version of the original measure, which we refer to as the Doppelhofer–Weeks measure, or DW.

DW is defined as the log of the cross-product ratio of binary indicators of predictor inclusion. It corresponds to the log of the ratio of posterior odds of including one predictor given the other is included to that of including that predictor given the other is not included. DW takes values in the $(-\infty, \infty)$ interval. A positive DW indicates jointness between two predictors, and a negative DW indicates disjointness. DW is undefined whenever the PIP of either predictor is zero or one. The authors provide the following thresholds to interpret this measure:

Table 1. Thresholds for the DW measure

DW	Interpretation
$(-\infty, -2)$	Strong disjointness
$[-2, -1)$	Significant disjointness
$[-1, 1]$	Independent inclusion
$(1, 2]$	Significant jointness
$(2, \infty)$	Strong jointness

Ley and Steel (2007) introduce two other measures of jointness, the Ley–Steel type 1 measure (LS1) and the Ley–Steel type 2 measure (LS2). LS1 is defined as the ratio of the posterior probability of including both predictors to that of including either one. LS2 is defined as the ratio of the posterior probability of including both predictors to that of including either one, but not both. The latter corresponds to the posterior odds ratio of the models that include both predictors to the models that include these predictors individually. These measures take values on $[0, 1]$ and $[0, \infty)$, respectively. Unlike DW, LS1 and LS2 are defined even when the PIP of one of the predictors equals zero or one. For both measures, low values

are interpreted as evidence of disjointness between the two predictors, and high values as evidence of jointness. The authors propose the following thresholds to interpret LS2 [LS1 can be determined based on $LS1 = LS2/(1 + LS2)$]:

Table 2. Thresholds for the LS2 and LS1

LS2	LS1*	Interpretation
$[0, 0.01)$	$[0, 0.01)$	Decisive disjointness
$[0.01, 0.03)$	$[0.01, 0.03)$	Very strong disjointness
$[0.03, 0.1)$	$[0.03, 0.09)$	Strong disjointness
$[0.1, 0.22)$	$[0.09, 0.18)$	Favorable disjointness
$[0.22, 3)$	$[0.18, 0.75)$	Independent inclusion
$[3, 10)$	$[0.75, 0.91)$	Favorable jointness
$[10, 30)$	$[0.91, 0.97)$	Strong jointness
$[30, 100)$	$[0.97, 0.99)$	Very strong jointness
$[100, \infty)$	$[0.99, 1]$	Decisive jointness

* rounded to 0.01 based on $LS1 = LS2/(1 + LS2)$

The Yule's Q binary similarity measure (YQ; for example, Yule [1912]) is also used as one of the jointness measures in BMA, where the inclusion of a predictor in a model is viewed as a binary event. This measure then compares the number of pairs in agreement (both predictors included or excluded) with the number of pairs in disagreement (only one of the predictors is included) relative to the total number of pairs. It takes values on $[-1, 1]$ and, as pointed out by Hofmarcher et al. (2018), can be viewed as a mapping of DW from $(-\infty, \infty)$ to $[-1, 1]$ with more interpretable bounds. Positive YQ implies jointness, negative YQ implies disjointness, and YQ close to zero means there is no association between the inclusion of the two predictors.

Like DW, YQ is undefined for predictors with PIPs equal to zero or one. To rectify this, in the case of sampling, Hofmarcher et al. (2018) proposed a modified Yule's Q measure (YQM), which can be viewed as a zero-cells adjustment in a cross-tabulation of the two binary inclusion indicators that adds 0.5 to zero cells (with a slight denominator adjustment). The rationale for this adjustment comes from assuming a multinomial distribution for the four counts in the cross-tabulation and a conjugate Dirichlet prior for the multinomial probabilities. An uninformative Jeffreys hyperprior is assumed for the parameters of the Dirichlet prior, which leads to the equal-cell adjustments of 0.5 for each cell. The interpretation of YQM is the same as for YQ.

There is no definitive recommendation in the literature for which measure should be preferred in practice. Hofmarcher et al. (2018) compare several theoretical properties of the five measures, and only YQM satisfies all properties considered by the authors. Unlike LS1, LS2, and YQM, the DW and YQ measures are undefined for predictors with PIPs of zero or one, but Man (2018) finds that DW produces more diverse results across various datasets, one of the desirable properties of patterns, compared with LS1 and LS2. The latter tend to produce more similar results across different datasets often favoring disjointness. Man (2018) concludes that jointness measures should be interpreted carefully in practice and that perhaps an aggregation of information from different measures might be a better approach than choosing just one measure.

Example: Jointness of growth determinants

In this section, we demonstrate the use of `bmastats jointness` to compute several jointness measures for the economic growth data considered by Hofmarcher et al. (2018) and Ley and Steel (2007), among others.

For a pair of predictors, `bmastats jointness` reports all available jointness measures. If more predictors are specified, the command reports the DW measure by default, but you can request other measures by specifying the respective options or use the `all` option to report all measures. The YQM measure is available only with sampling and only when the `frequency` option is also specified.

▷ Example 1: Jointness of growth determinants

Consider the `econgrowth` dataset from example 17 of [BMA] **bmaregress**, where a country's economic growth is modeled using 41 potential predictors, but here, to save time, we use a smaller MCMC size of 50,000.

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

. bmaregress gdpgrowth abslat-yrsopen, mprior(uniform) mcmcsize(50000) rseed(18)

Burn-in ...
Simulation ...
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	=	6,394
		For CPMP >= .9	=	1,100
Priors:		Mean model size	=	9.422
Models: Uniform		Burn-in	=	2,500
Cons.: Noninformative		MCMC sample size	=	50,000
Coef.: Zellner's g		Acceptance rate	=	0.1720
g: Benchmark, g = 1,681		Shrinkage, g/(1+g)	=	0.9994
sigma2: Noninformative		Mean sigma2	=	.000055

Sampling correlation = 0.7884

gdpgrowth	Mean	Std. dev.	Group	PIP
gdp60	-.0162155	.0029558	16	.99968
confucian	.0563032	.0125301	9	.9993
lifeexp	.0008464	.0003022	22	.96505
equipinv	.1647531	.0609101	12	.95748
subsahara	-.0119874	.007829	38	.7794
muslim	.0087687	.0070261	24	.68554
ruleoflaw	.0087141	.0085463	35	.56117
ecoorg	.0014106	.0015073	10	.51286
protestants	-.0060471	.0070828	30	.48112
yrsopen	.0069856	.0080953	41	.4714
nequipinv	.0257697	.0321904	25	.43821
mining	.0163398	.0221579	23	.40574
latamerica	-.0013252	.0035112	21	.15603
prscenroll	.0032339	.0084244	31	.15337
buddha	.0015631	.0046289	6	.12457
blmktpm	-.0009528	.0028545	4	.12028
catholic	-.000449	.0026236	7	.092943
hindu	-.001704	.0068042	18	.073258
civllib	-.0001468	.0006411	8	.062838
rfexdist	-2.53e-06	.0000139	34	.042526
prexports	-.0004325	.0023676	29	.041549
polrights	-.0000654	.0003753	27	.039649
wardummy	-.0000986	.0007291	39	.026089
english	-.0001632	.0012182	11	.025993
age	-1.13e-06	8.39e-06	2	.024626
labforce	1.26e-09	1.29e-08	20	.023675
foreign	.0001093	.0008783	14	.021918
spanish	.0000587	.0007927	36	.016174
stdbmp	-2.46e-07	2.48e-06	37	.015815
french	.0000664	.0006795	15	.01551
ethnol	.0000719	.0008001	13	.013968
Always				
_cons	.0700867	.0203619	0	1

Note: Coefficient posterior means and std. dev. estimated from 6,394 models.
Note: Default prior is used for parameter g.
Note: 10 predictors with PIP less than .01 not shown.

Let's compute all available jointness measures for a pair of predictors, say, political rights, polrights, and civil liberties, civllib.

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

	Jointness
Doppelhofer–Weeks	-2.994163
Ley–Steel type 1	.0013503
Ley–Steel type 2	.0013522
Yule's Q	-.9046195

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

All reported jointness measures suggest disjointness or substitutability between these two predictors. Based on table 1, the DW value suggests strong disjointness ($-2.99 < -2$). Based on table 2, LS1 and LS2 values suggest decisive disjointness ($0.0014 < 0.01$). The YQ value of -0.9 is close to -1, which also suggests strong disjointness. The results suggest that the presence of one of polrights or civllib in the model greatly reduces the probability of inclusion of the other predictor.

From bmaregress, the initial GDP (gdp60), the fraction Confucian (confucian), and the life expectancy (lifeexp) are among the influential predictors of the economic growth. Let's examine their pairwise jointness by computing their DW scores first.

```
. bmastats jointness gdp60 confucian lifeexp, dweeks
Computing model probabilities ...
Doppelhofer–Weeks jointness
```

	gdp60	confucian	lifeexp
gdp60	.	.	7.073402
confucian	.	.	3.076793
lifeexp	7.073402	3.076793	.

```
Notes: Using analytical PMPs. See thresholds.
       The measure is undefined in some cases.
```

Variables lifeexp and confucian exhibit strong jointness (DW $= 3.08 > 2$) and thus can be viewed as complements. The same applies to predictors lifeexp and gdp60. But the measure is missing (undefined) for gdp60 and confucian, which can happen as discussed in *Jointness as a measure of variable-inclusion dependence*.

Let's take a closer look at this pair of predictors:

```
. bmastats jointness confucian gdp60, dweeks
Computing model probabilities ...
Variables: confucian gdp60
```

	Jointness
Doppelhofer–Weeks	.

```
Notes: Using analytical PMPs. See
       thresholds. Some measures are
       undefined.
```

```
. display as txt "Prob. both excluded = " as res r(p00)
Prob. both excluded = 0
. display as txt "Prob. only gdp60 included = " as res r(p01)
Prob. only gdp60 included = .00070228
. display as txt "Prob. only confucian included = " as res r(p10)
Prob. only confucian included = .00031608
. display as txt "Prob. both included = " as res r(p11)
Prob. both included = .99898164
```

For a pair of predictors, `bmastats jointness` stores the respective joint probabilities of inclusion in `r(p00)`, `r(p01)`, `r(p10)`, and `r(p11)`; see *Methods and formulas* for details. We can see that there are no models in which both predictors are excluded, `r(p00) = 0`, and there are very few models in which `gdp60` is included but `confucian` is not, and vice versa. These results are not surprising given that `gdp60`'s PIP is almost 1. Given the high probability `r(p11) = 0.999` for both predictors to be included together, it might not be unreasonable to suspect jointness between these two predictors.

But let's explore other measures. Because the LS measures tend to provide similar results, let's look at just one of them.

```
. bmastats jointness gdp60 confucian lifeexp, lsteel2
Computing model probabilities ...
Ley-Steel type 2 jointness
```

	gdp60	confucian	lifeexp
gdp60	.	980.9682	27.85594
confucian	980.9682	.	27.5357
lifeexp	27.85594	27.5357	.

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

All three pairs of predictors exhibit strong jointness according to the LS2 measure.

Because of the similarity in the definitions of the DW and YQ measures, we would expect the latter to be missing, too, for the `confucian` and `gdp60` pair. We leave this for you to verify by specifying the `yuleq` option with `bmastats jointness`.

Because sampling is used in this example, we can also compute the YQM measure. To do this, we should also specify the `frequency` option to use frequency PMPs in the computation.

```
. bmastats jointness gdp60 confucian lifeexp, yuleqm frequency
Modified Yule's Q jointness
```

	gdp60	confucian	lifeexp
gdp60	.	-.9999963	.9843265
confucian	-.9999963	.	.5851771
lifeexp	.9843265	.5851771	.

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

All values are above 0.5, which suggests strong jointness for all pairs of predictors.

It appears that gdp60, confucian, and lifeexp can be viewed as complements in describing the economic growth, gdpgrowth, which is consistent with findings in Hofmarcher et al. (2018) and Ley and Steel (2007).

◁

Stored results

bmastats jointness stores the following in r() with two predictors:

Scalars
r(dw)	Doppelhofer–Weeks measure
r(ls1)	Ley–Steel type 1 measure
r(ls2)	Ley–Steel type 2 measure
r(yq)	Yule's Q measure
r(yqm)	modified Yule's Q measure (when available)
r(p00)	posterior probability of excluding both predictors
r(p01)	posterior probability of including only the second predictor
r(p10)	posterior probability of including only the first predictor
r(p11)	posterior probability of including both predictors

Macros
r(varnames)	names of specified variables
r(pmptype)	analytical or frequency

bmastats jointness stores the following in r() with more than two predictors:

Macros
r(varnames)	names of specified variables
r(pmptype)	analytical or frequency

Matrices
r(dwmat)	Doppelhofer–Weeks measures
r(ls1mat)	Ley–Steel type 1 measures
r(ls2mat)	Ley–Steel type 2 measures
r(yqmat)	Yule's Q measures
r(yqmmat)	modified Yule's Q measures (when available)

Methods and formulas

Consider two predictors X_u and X_v of outcome $\mathbf{y}$. Let U denote the event that X_u is included in a model and V denote the event that X_v is included in a model. Let U^c and V^c be the respective complementary events. Let J be the full model space $J_F = \{1, 2, \ldots, 2^p\}$ in the case of model enumeration or the set of distinct models visited by the MCMC sampler in the case of sampling. The computation of jointness measures involves the following posterior probabilities:

$$P_U = \Pr(U|\mathbf{y}) = \sum_{j \in J} I(X_u \in M_j) P_a(M_j|\mathbf{y})$$

$$P_V = \Pr(V|\mathbf{y}) = \sum_{j \in J} I(X_v \in M_j) P_a(M_j|\mathbf{y})$$

$$P_{UV} = \Pr(U, V|\mathbf{y}) = \sum_{j \in J} I(X_u \in M_j, X_v \in M_j) P_a(M_j|\mathbf{y})$$

$$P_{U^cV} = \Pr(U^c, V|\mathbf{y}) = \sum_{j \in J} I(X_u \notin M_j, X_v \in M_j) P_a(M_j|\mathbf{y})$$

$$P_{UV^c} = \Pr(U, V^c|\mathbf{y}) = \sum_{j \in J} I(X_u \in M_j, X_v \notin M_j) P_a(M_j|\mathbf{y})$$

$$P_{U^cV^c} = \Pr(U^c, V^c|\mathbf{y}) = \sum_{j \in J} I(X_u \notin M_j, X_v \notin M_j) P_a(M_j|\mathbf{y})$$

where $I(\cdot)$ is the indicator function and $P_a(M_j|\mathbf{y})$ is the analytical PMP defined by (7) in *Posterior model probability* in *Methods and formulas* of [BMA] **bmaregress**. For a *random g* or with option `frequency`, the above posterior probabilities are estimated from an MCMC posterior sample of models $\{m_t\}_{t=1}^T$:

$$\hat{P}_U = \frac{1}{T} \sum_{t=1}^T I(X_u \in m_t)$$

$$\hat{P}_V = \frac{1}{T} \sum_{t=1}^T I(X_v \in m_t)$$

$$\hat{P}_{UV} = \frac{1}{T} \sum_{t=1}^T I(X_u \in m_t, X_v \in m_t)$$

$$\hat{P}_{U^cV} = \frac{1}{T} \sum_{t=1}^T I(X_u \notin m_t, X_v \in m_t)$$

$$\hat{P}_{UV^c} = \frac{1}{T} \sum_{t=1}^T I(X_u \in m_t, X_v \notin m_t)$$

$$\hat{P}_{U^cV^c} = \frac{1}{T} \sum_{t=1}^T I(X_u \notin m_t, X_v \notin m_t)$$

Doppelhofer and Weeks (2009) proposed the following measure of jointness:

$$J_{\text{DW}} = \log\left(\frac{P_{UV} P_{U^cV^c}}{P_{U^cV} P_{UV^c}}\right) \in (-\infty, \infty)$$

Ley and Steel (2007) proposed the following measures of jointness:

$$J_{\text{LS1}} = \frac{P_{UV}}{P_U + P_V - P_{UV}} \in [0, 1]$$

$$J_{\text{LS2}} = \frac{P_{UV}}{P_{UV^c} + P_{U^cV}} \in [0, \infty)$$

The YQ measure (Yule 1912) is

$$J_{\text{YQ}} = \frac{P_{UV}P_{U^cV^c} - P_{UV^c}P_{U^cV}}{P_{UV}P_{U^cV^c} + P_{UV^c}P_{U^cV}} \in [-1, 1]$$

When sampling is used, the posterior probabilities in the above formulas are replaced with their respective frequency-based estimates, such as $\hat{P}_{UV}$ and $\hat{P}_{U^cV}$. Also, when the `frequency` option is specified, the YQM measure (Hofmarcher et al. 2018) is computed as

$$J_{\text{YQM}} = \frac{(\hat{P}_{UV} + \xi)(\hat{P}_{U^cV^c} + \xi) - (\hat{P}_{UV^c} + \xi)(\hat{P}_{U^cV} + \xi)}{(\hat{P}_{UV} + \xi)(\hat{P}_{U^cV^c} + \xi) + (\hat{P}_{UV^c} + \xi)(\hat{P}_{U^cV} + \xi) - 2\xi^2} \in [-1, 1]$$

where $\xi = 1/(2T)$.

References

Doppelhofer, G., and M. Weeks. 2006. Jointness of growth determinants. Cambridge Working Papers in Economics (CWPE) 0542, University of Cambridge, Faculty of Economics. https://doi.org/10.17863/CAM.5050.

———. 2009. Jointness of growth determinants. *Journal of Applied Econometrics* 24: 209–244. https://doi.org/10.1002/jae.1046.

Hofmarcher, P., J. C. Cuaresma, B. Grün, S. Humer, and M. Moser. 2018. Bivariate jointness measures in Bayesian model averaging: Solving the conundrum. *Journal of Macroeconomics* 57: 150–165. https://doi.org/10.1016/j.jmacro.2018.05.005.

Ley, E., and M. F. J. Steel. 2007. Jointness in Bayesian variable selection with applications to growth regression. *Journal of Macroeconomics* 29: 476–493. https://doi.org/10.1016/j.jmacro.2006.12.002.

Man, G. 2018. Critical appraisal of jointness concepts in Bayesian model averaging: Evidence from life sciences, sociology, and other scientific fields. *Journal of Applied Statistics* 45: 845–867. https://doi.org/10.1080/02664763.2017.1318839.

Yule, G. U. 1912. On the methods of measuring association between two attributes. *Journal of the Royal Statistical Society* 75: 579–652. https://doi.org/10.2307/2340126.

Also see

Description

bmastats lps computes and optionally stores in a new variable the log predictive-score (LPS) for observations after the bmaregress command. LPS is used to assess the predictive performance of Bayesian model averaging (BMA) models.

Quick start

Compute and summarize LPS for all observations

    bmastats lps

Same as above, but use a known variance to compute and report the entropy

    bmastats lps, sigma2(0.1)

Compute LPS for the first 12 observations, and store them in newvar

    bmastats lps in 1/12, generate(newvar)

Compute LPS for previously stored bma1 and bma2 estimation results, and store respective LPS in variables newvar1 and newvar2

    bmastats lps bma1 bma2, generate(newvar*)

Display multiple estimation results compactly in one table

    bmastats lps bma1 bma2, compact

After fitting bmaregress using a training sample, sample == 1, compute LPS for out-of-sample observations—test sample sample == 2—to compare out-of-sample predictive performance of two BMA models with estimation results stored in bma1 and bma2

    bmastats lps bma1 bma2 if sample == 2

Menu

Statistics > Bayesian model averaging > Log predictive-score

Syntax

Summarize LPS for current estimation results

> bmastats lps [*if*] [*in*] [, <u>sigma2</u>(#) <u>gen</u>erate(*newvar*[, double])]

Summarize LPS for multiple estimation results

> bmastats lps *namelist* [*if*] [*in*] [, <u>sigma2</u>(# [# [. . .]])
>
> <u>gen</u>erate(*newvarspec*[, double]) compact]

namelist includes names of previously stored estimation results after bmaregress. The current (active) results are the default.

newvarspec is a *newvar* for a single estimation result or *newvarlist* or *stub** for multiple estimation results.

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

Options

       Main

sigma2(# [# [. . .]]) specifies the variance to be used to compute the entropy. With out-of-sample observations, the posterior mean estimate of the error variance from bmaregress is used. With multiple estimation results, you can specify just one variance value to be used for all estimation results or specific values for each result. A missing value (.) in place of a # means that the default variance estimate is to be used for the corresponding estimation result. This option is useful to compute the entropy more precisely when the variance is known, such as during simulation studies.

generate(*newvarspec*[, double]) specifies the name (or names) of new variables to store the LPS values. A new variable name is specified with one estimation result. A list of new variable names or *stub** is specified with multiple estimation results.

> double specifies that the new variables be stored as double. If the double suboption is not specified, the variables are stored using the current type as set by set type (see [D] **generate**), which is float by default.

compact specifies that multiple estimation results be displayed in a compact form. By default, a separate header and table are displayed for each estimation result. If option compact is specified, the results are displayed in one table with a common header. If the variances differ across multiple estimation results, the variance and entropy are not reported in the header. This option requires at least two estimation results.

Remarks and examples

To evaluate the predictive performance of a model, one needs a measure that summarizes the quality of predictions (Piironen and Vehtari 2017). Good (1952) suggests that such a measure be defined based on the posterior predictive distribution. The LPS is defined as the negative of the log of the posterior predictive density evaluated at an observation. The LPS for out-of-sample observations can be used to evaluate the predictive performance of models. Fernández, Ley, and Steel (2001b) were among the first to use the LPS to evaluate the predictive performance of BMA. And Ley and Steel (2012) compared a variety of *g*-priors for the linear BMA using the LPS.

As pointed out in Fernández, Ley, and Steel (2001a), the LPS is related to the Kullback–Leibler divergence between the actual sampling density and the out-of-sample predictive density. In theory, the expected LPS is bounded from below by the entropy of the sampling density, but in practice, this may not always hold because the entropy itself is being estimated. The closer the expected value is to the entropy, the more accurate the prediction. For the normal linear model, the entropy is given by $0.5\{\log(2\pi\sigma^2) + 1\}$, where σ^2 is the error variance.

bmastats lps reports the mean, minimum, and maximum of LPS for each specified estimation result. By default, the command reports the LPS summaries computed using the current dataset that includes in-sample observations, which were used during estimation and are identified by e(sample) == 1, for the current or active estimation results. These summaries are useful when comparing multiple models. By themselves, their values are not directly interpretable, unless the entropy is known or can be estimated reliably. In this case, the closer the mean LPS is to the entropy, the better the model fit. For instance, if you are using bmastats lps with a simulated dataset, you can specify the known error variance in the sigma2() option, and the corresponding entropy value will be reported. You can compare this value with the mean LPS to check model fit. When bmastats lps is used with only out-of-sample observations, observations not used to fit the model for which e(sample) == 0, the command uses the posterior mean estimate of the error variance from the fitted bmaregress model to estimate and report the entropy. You can then compare the out-of-sample mean LPS with the estimated entropy to check the out-of-sample model fit, assuming that the entropy estimate is a reliable estimate of the entropy of the true model. The entropy estimate is not reported when bmastats lps is used with in-sample observations.

You can use the generate() option to store observation-specific LPS values in a new variable. This is equivalent to using bmapredict, lps; see [BMA] **bmapredict**. With multiple estimation results, you must specify the same number of new variable names or one variable stub in generate().

With multiple estimation results, bmastats lps displays a separate table for each result. You can use the compact option to display the results in one table, which is more convenient for comparison.

Below, we demonstrate a more common usage of bmastats lps for comparing out-of-sample predictive performance of multiple BMA models.

▷ Example 1: Prediction of systolic blood pressure

Consider the chd dataset from Hastie, Tibshirani, and Friedman (2009), which is a subset of the coronary heart disease dataset used in Rossouw et al. (1983). It contains information about potential heart disease factors such as age, adiposity, sbp (systolic blood pressure), tobacco (consumption of), ldl (low density liboprotein), and others.

```
. use https://www.stata-press.com/data/r19/chd
(Coronary risk factor data from rural communities)
. describe
Contains data from https://www.stata-press.com/data/r19/chd.dta
 Observations:             462              Coronary risk factor data from
                                              rural communities
    Variables:              10              7 Mar 2025 09:19
                                            (_dta has notes)
```

Variable name	Storage type	Display format	Value label	Variable label
sbp	int	%8.0g		Systolic blood pressure
tobacco	float	%6.2f		Yearly tobacco use (kg)
ldl	float	%6.2f		Low-density lipoprotein (LDL)
adiposity	float	%6.2f		Body adiposity index
famhist	byte	%9.0g	noyes	Family history of ischaemic heart disease
typea	byte	%8.0g		Type A personality score
obesity	float	%6.2f		Obesity (body mass index)
alcohol	float	%7.2f		Alcohol use (grams/day)
age	byte	%8.0g		Age (years)
chd	byte	%8.0g	noyes	Diagnosed with CHD

```
Sorted by:
```

We wish to use BMA to build a prediction model for systolic blood pressure using all other available factors, tobacco through age, as predictors. We consider the log-transformed sbp variable, lsbp, to make its distribution closer to normal. We also split the sample into five groups to have training and test subsamples to evaluate the predictive performance of a model. Because the sample is split randomly, we specify a random-number seed with splitsample for reproducibility.

```
. splitsample, generate(sample) nsplit(5) rseed(100)
. generate double lsbp = log(sbp)
```

We will use the first four subsamples as our training data to fit a model. And we will check the model's performance using the fifth, test, subsample. To demonstrate, we fit the bmaregress command using the default settings.

```
. bmaregress lsbp tobacco-age if sample != 5

Enumerating models ...
Computing model probabilities ...

Bayesian model averaging              No. of obs        =      370
Linear regression                     No. of predictors =        8
Model enumeration                            Groups =           8
                                             Always =           0
Priors:                               No. of models     =      256
  Models: Beta-binomial(1, 1)             For CPMP >= .9 =        7
   Cons.: Noninformative             Mean model size    =    2.132
   Coef.: Zellner's g
       g: Benchmark, g = 370         Shrinkage, g/(1+g) = 0.9973
  sigma2: Noninformative             Mean sigma2        =    0.015
```

lsbp	Mean	Std. dev.	Group	PIP
age	.0020223	.0008055	8	.93569
adiposity	.0042121	.0016916	3	.92348
obesity	.0004441	.001573	6	.10111
typea	−.0000344	.0002133	5	.047293
alcohol	9.80e−06	.0000733	7	.03971
tobacco	−.0000101	.0002961	1	.028973
ldl	−9.92e−06	.0005911	2	.02786
famhist	.0000648	.0022977	4	.027833
Always				
_cons	4.71339	.0366564	0	1

```
Note: Coefficient posterior means and std. dev. estimated from 256 models.
Note: Default priors are used for models and parameter g.
```

With a small number of predictors, the model space is explored fully by model enumeration. The default fixed benchmark g-prior is used, which results in a negligible shrinkage factor of almost 1. The posterior mean size of the models is 2.1, and all potential predictors have PIPs greater than 2%.

We use the bmastats lps command to evaluate the prediction performance of the fit model on the test sample by using the condition if sample == 5.

```
. bmastats lps if sample == 5

Log predictive-score (LPS)

Number of observations =        92
Posterior mean sigma2  =   .015329
Entropy                =  -.670055
```

	Mean	Minimum	Maximum
LPS	−.5183101	−1.169732	4.861686

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

The command reports the number of predicted observations, the estimated entropy, and the variance estimate used to compute the entropy in the header. The output table reports the mean, minimum, and maximum LPS values.

The mean LPS of −0.52 is, as expected, greater than the entropy estimate, −0.67. The closer it is to the entropy, the better the predictive performance. The actual LPS values cannot be interpreted, but we can use them to compare the predictive performances among several BMA models.

Based on their PIPs above 90%, `age` and `adiposity` appear to be the more important predictors of the elevated blood pressure (at least based on our training sample). We may ask what would be the performance of the model using only these two predictors?

To fit just one model of interest using `bmaregress`, we can specify all predictors of interest as one always-included group as follows. (In this case, the results from `bmaregress` will be similar to those from `regress`; see [R] **regress**.)

```
. bmaregress lsbp (age adiposity, always) if sample != 5

Enumerating models ...
Computing model probabilities ...

Bayesian model averaging                No. of obs        =       370
Linear regression                       No. of predictors =         2
Model enumeration                                 Groups  =         0
                                                  Always  =         2
Priors:                                 No. of models     =         1
  Models: Beta-binomial(1, 1)               For CPMP >= .9 =         1
   Cons.: Noninformative                 Mean model size   =     2.000
   Coef.: Zellner's g
       g: Benchmark, g = 370             Shrinkage, g/(1+g) = 0.9973
  sigma2: Noninformative                 Mean sigma2       =     0.015
```

lsbp	Mean	Std. dev.	Group	PIP
Always				
age	.002073	.0005608	0	1
adiposity	.0044281	.0010493	0	1
_cons	4.715601	.0236217	0	1

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

`bmaregress` visits only one model: `regress lsbp age adiposity`. In this case, the prior model probability has no effect on the BMA results, and `bmaregress` is equivalent to fitting a Bayesian linear model with Zellner's g-prior on regression coefficients with $g = 370$ and Jeffreys prior for the error variance parameter; see [BAYES] **bayes: regress** or [BAYES] **bayesmh**.

Let's calculate the mean LPS score for this model on the test sample.

```
. bmastats lps if sample == 5

Log predictive-score (LPS)

Number of observations =        92
Posterior mean sigma2  =   .015277
Entropy                =   -.67177
```

	Mean	Minimum	Maximum
LPS	-.5173158	-1.171843	4.890529

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

The mean LPS score of -0.52 is about the same as that of the full BMA regression, which suggests a similar predictive performance for the two models.

We can also investigate the impact of a particular factor on the prediction performance of our BMA model. For instance, in the first model, the tobacco usage term had a PIP of about 3%. We may investigate further whether the tobacco usage indeed has a relatively low impact on systolic blood pressure. To do this, we can fit BMA to all predictors except `tobacco` and compare its predictive performance to the full model.

```
. bmaregress lsbp ldl-age if sample != 5

Enumerating models ...
Computing model probabilities ...

Bayesian model averaging                        No. of obs         =      370
Linear regression                               No. of predictors  =        7
Model enumeration                                       Groups =             7
                                                        Always =             0
Priors:                                         No. of models      =      128
  Models: Beta-binomial(1, 1)                       For CPMP >= .9 =        6
   Cons.: Noninformative                        Mean model size    =    2.150
   Coef.: Zellner's g
       g: Benchmark, g = 370                    Shrinkage, g/(1+g) = 0.9973
  sigma2: Noninformative                        Mean sigma2        =    0.015
```

lsbp	Mean	Std. dev.	Group	PIP
age	.0020391	.0007811	7	.94501
adiposity	.0041866	.0016762	2	.92399
obesity	.0004608	.0015953	5	.10886
typea	−.0000412	.0002329	4	.056801
alcohol	.0000118	.0000802	6	.04785
ldl	−.0000121	.0006497	1	.033757
famhist	.0000758	.0025216	3	.03369
Always				
_cons	4.713174	.0370391	0	1

```
Note: Coefficient posterior means and std. dev. estimated from 128 models.
Note: Default priors are used for models and parameter g.

. bmastats lps if sample == 5

Log predictive-score (LPS)

Number of observations =        92
Posterior mean sigma2  =  .015323
Entropy                = -.670276
```

	Mean	Minimum	Maximum
LPS	−.5186576	−1.169858	4.855259

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

The mean LPS score of −0.52 is again close to that of the full model. This suggests that excluding the `tobacco` predictor does not diminish the predictive performance of the model. It does not appear to be a strong predictor in this dataset.

To more easily compare the LPS from the above models, we could have stored the estimation results from each model first and then specified these results with bmastats lps. To store the estimation results, we must use the saving() option with bmaregress. For instance,

```
. quietly bmaregress lsbp tobacco-age if sample != 5, saving(bma1)
. estimates store bma1
. quietly bmaregress lsbp (age adiposity, always) if sample != 5, saving(bma2)
. estimates store bma2
. quietly bmaregress lsbp ldl-age if sample != 5
. bmastats lps bma1 bma2 . if sample == 5, compact
Log predictive-score (LPS)
Number of observations = 92
```

LPS	Mean	Minimum	Maximum
bma1	-.5183101	-1.169732	4.861686
bma2	-.5173158	-1.171843	4.890529
Active	-.5186576	-1.169858	4.855259

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

We also specified the compact option to display the results in one table with a common header. Because the variance estimates differ between the estimation results, the variance and entropy are not shown in the header. (You can see them if you do not specify compact.) If the variances were the same, the variance and the entropy would have been displayed.

Remember to remove the generated datasets if they are no longer needed.

```
. erase bma1.dta
. erase bma2.dta
```

◁

In addition to the regression-function specification, in BMA, it is also important to evaluate the effect of different prior specifications on the results. Depending on how much information there is in the data about the data-generating process, different prior specifications may lead to different conclusions. We explore this in example 2.

▷ Example 2: Comparing different g-priors using cross-validation

For some data, the performance of linear BMA may depend strongly on the choice of the prior for regression coefficients (Fernández, Ley, and Steel 2001a). The classical Zellner's g-prior is commonly used in BMA for regression coefficients. The g parameter of this prior can be fixed or random. For a random g, a hyperprior for g must be specified. With a variety of prior choices available for g, the BMA specification can be challenging. As an objective criterion for choosing a prior, we can use a model's predictive performance on the data not used for fitting the model. In this example, we show how we can use cross-validation of log predictive-score to compare different choices of priors.

Let's continue with the analysis of the chd dataset from example 1. Here we will compare five different priors for g: two fixed and three random. The two fixed priors are the default benchmark, bench, and empirical Bayes local, ebl. The former uses a constant g-value for all models, whereas the latter uses a model-specific g. The three random priors are the benchmark beta-shrinkage, betabench(1); hyper-g/n, hypergn(3); and robust, robust; see *Priors for parameter g* in *Methods and formulas* of [BMA] **bmaregress** for details.

In example 1, we randomly split our sample into five subsamples. For cross-validation, we will treat each of five subsamples as a test sample and fit each of five models to the remaining four subsamples, which will give us 25 results. For each model i and test sample j, we will save the mean LPS in a 5-by-5 matrix.

To avoid manually fitting 25 models, we will automate this task by writing a simple program later. But first, let's go over one step of this automation manually. We will use a robust prior for g and, as in example 1, withhold the fifth sample for testing.

We use the `gprior(robust)` option to specify the robust g-prior and, because this is a random prior, use the `rseed()` option for reproducibility.

```
. bmaregress lsbp tobacco-age if sample != 5, gprior(robust) rseed(18)

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

Bayesian model averaging                    No. of obs        =      370
Linear regression                           No. of predictors =        8
MC3 and adaptive MH sampling                            Groups =        8
                                                        Always =        0
                                            No. of models     =       78
                                               For CPMP >= .9 =        9
Priors:                                     Mean model size   =    2.366
  Models: Beta-binomial(1, 1)               Burn-in           =    2,500
   Cons.: Noninformative                    MCMC sample size  =   10,000
   Coef.: Zellner's g                       Acceptance rate   =   0.3419
       g: Robust
  sigma2: Noninformative                    Mean sigma2       =    0.015

Sampling correlation = 0.9974
```

lsbp	Mean	Std. dev.	Group	PIP
age	.0020676	.0007043	8	.9688
adiposity	.0041715	.0015489	3	.9442
obesity	.0004562	.0015458	6	.1378
typea	−.0000531	.0002624	5	.0739
ldl	−.0000286	.0009378	2	.0716
alcohol	.0000163	.0000937	7	.0665
famhist	.0001123	.0032103	4	.0555
tobacco	−.000021	.0003703	1	.0474
Always				
_cons	4.713145	.0364241	0	1

Note: Coefficient posterior means and std. dev. estimated from 78 models.
Note: Default prior is used for models.

	Mean	Std. dev.	MCSE	Median	Equal-tailed [95% cred. interval]	
g	325.2053	671.8658	18.0907	185.5779	82.95733	1396
Shrinkage	.9945226	.0029212	.000183	.9946403	.9880892	.9992842

The results are somewhat similar to those from the first model in example 1, except g is now random and Markov chain Monte Carlo (MCMC) sampling is used to estimate its posterior distribution.

```
. bmastats lps if sample == 5

Log predictive-score (LPS)

Number of observations =        92
Posterior mean sigma2  =  .015309
Entropy                =  -.670731
```

	Mean	Minimum	Maximum
LPS	-.5191868	-1.170108	4.839892

```
Note: Using frequency PMPs.

. return list

scalars:
                  r(N) =  92
             r(sigma2) =  .0153086079526961
            r(entropy) =  -.6707314655467329

macros:
            r(pmptype) : "frequency"

matrices:
            r(summary) :  1 x 3

. matrix list r(summary)

r(summary)[1,3]
            Mean       Minimum      Maximum
LPS  -.51918683   -1.1701082     4.839892

. display el(r(summary),1,1)
-.51918683
```

After running bmastats lps, we use return list to display the stored results. The LPS summary is
stored in the r(summary) matrix. We can access its first element, the mean LPS, by using the el() Stata
matrix function.

Let's now automate the above in a program.

```
. program bma_cv
  1.          args matlps i gprior extraopts
  2.          forvalues j = 1/5 {
  3.                  bmaregress lsbp tobacco-age if sample != `j', ///
  >                                          gprior(`gprior') `extraopts'
  4.                  bmastats lps if sample == `j'
  5.                  matrix `matlps'[`i',`j'] = el(r(summary),1,1)
  6.          }
  7. end
```

The bma_cv program takes four arguments: a matrix name matlps to store the mean LPS values, the ma-
trix row index i, the prior specification gprior, and any additional options extraopts for bmaregress,
such as rseed() for the random g-priors. We then perform cross-validation by looping over the five sub-
samples and withholding each one of them for testing. For each prior i, we save the mean LPS for each
cross-validation sample j in row `i' and column `j' of the specified matrix `matlps'.

We create a 5-by-5 matrix `lpscvmat` to store the mean LPS values and call the `bma_cv` program for each prior specification. Because the last three priors are random and thus use MCMC sampling to estimate the results, the program may take a few moments longer to run compared with the one for the fixed priors.

```
. matrix lpscvmat = J(5, 5, .)
. quietly bma_cv lpscvmat 1 "bench"
. quietly bma_cv lpscvmat 2 "ebl"
. quietly bma_cv lpscvmat 3 "betabench 1" "rseed(18)"
. quietly bma_cv lpscvmat 4 "hypergn 3" "rseed(18)"
. quietly bma_cv lpscvmat 5 "robust" "rseed(18)"
. matrix rownames lpscvmat = "bench" "ebl" "betabench 1" "hypergn 3" "robust"
. matrix colnames lpscvmat = "CV1" "CV2" "CV3" "CV4" "CV5"
. matlist lpscvmat
```

	CV1	CV2	CV3	CV4	CV5
bench	-.5771455	-.6311993	-.6583148	-.7233847	-.5183101
ebl	-.5778856	-.6339036	-.6623052	-.7182023	-.5224309
betabench 1	-.5774324	-.6320177	-.6591742	-.7232513	-.5199717
hypergn 3	-.5774459	-.6322509	-.6600753	-.7212745	-.5203261
robust	-.5758851	-.6344241	-.6594026	-.7210064	-.5191868

Smaller mean LPS scores indicate better predictive performance. All mean LPSs are rather similar, which suggests a similar predictive performance for the models with considered priors (assuming a similar variability between LPSs).

We can use a Mata matrix function, `colmin()`, to compute the smallest mean LPS value among all models for each cross-validated sample.

```
. mata: colmin(st_matrix("lpscvmat"))
```

	1	2	3	4
1	-.5778855962	-.6344241355	-.6623052219	-.723384674

	5
1	-.5224308762

The second model, `ebl`, has the lowest mean LPS for the majority of models.

Let's fit a model using the `ebl` *g*-prior to the entire dataset.

```
. bmaregress lsbp tobacco-age, gprior(ebl)

Enumerating models ...
Computing model probabilities ...

Bayesian model averaging                         No. of obs        =     462
Linear regression                                No. of predictors =       8
Model enumeration                                         Groups  =       8
                                                          Always  =       0
Priors:                                          No. of models     =     256
  Models: Beta-binomial(1, 1)                       For CPMP >= .9 =      46
   Cons.: Noninformative                         Mean model size   =   3.516
   Coef.: Zellner's g
       g: Empirical Bayes (local)                Shrinkage, g/(1+g) = 0.9660
  sigma2: Noninformative                         Mean sigma2       =   0.016
```

lsbp	Mean	Std. dev.	Group	PIP
age	.0026375	.0006026	8	.99981
adiposity	.0024261	.0016807	3	.7727
alcohol	.0003029	.0003148	7	.58378
obesity	.0014017	.0021937	6	.40683
tobacco	.0001807	.0007427	1	.2009
typea	−.0000758	.0003079	5	.19862
famhist	−.0005746	.0052701	4	.17773
ldl	−8.63e−06	.0013313	2	.17572
Always				
_cons	4.706904	.0433882	0	1

```
Note: Coefficient posterior means and std. dev. estimated from 256 models.
Note: Default prior is used for models.
```

Using the full dataset and the `ebl` prior, `age` is still a highly important predictor with a PIP of close to 1. The PIP of 0.77 for `adiposity` is now smaller, but the PIPs of 0.58 and 0.41 for `alcohol` and `obesity`, respectively, are now noticeably higher. The other predictors have slightly higher PIPs too, but they are still somewhat weak predictors relative to the other four.

If we run models using the other considered priors, we will find that our conclusions about the relative importance of predictors are almost the same. Of course, the PIP estimates themselves will be different because different priors impose different levels of shrinkage for the coefficients.

Beware that the results will also change if you use a different random-number seed. But for stable datasets and models, changing the random-number seed should not affect the conclusions.

◁

Stored results

bmastats lps stores the following in r() with one estimation result:

Scalars
  r(N)   number of observations
  r(sigma2)   variance used to compute entropy
  r(entropy)   entropy estimate

Macros
  r(name)   name of the estimation result (if any specified)
  r(pmptype)   analytical or frequency

Matrices
  r(summary)   summary matrix with mean, minimum, and maximum LPS

bmastats lps stores the following in r() for each #th estimation result:

Scalars
  r(N#)   number of observations
  r(sigma2_#)   variance used to compute entropy
  r(entropy#)   entropy estimate

Macros
  r(name#)   name of the #th estimation result
  r(pmptype#)   analytical or frequency

Matrices
  r(summary#)   summary matrix with mean, minimum, and maximum LPS

bmastats lps, compact stores the following in r() with multiple estimation results:

Macros
  r(names)   names of estimation results
  r(pmptypes)   type of PMP for each model: analytical or frequency

Matrices
  r(N)   numbers of observations
  r(sigma2)   variances used to compute entropy
  r(entropy)   entropy estimates
  r(summary)   summary matrix with mean, minimum, and maximum LPS for all estimation results

Methods and formulas

Let y^* be a new observation of the outcome and $\mathbf{x}^*$ be a p-vector of new values for the predictors, and let $\mathbf{z}^* = \mathbf{x}^* - \overline{\mathbf{x}}$, where $\overline{\mathbf{x}}' = \mathbf{1}'\mathbf{X}/n$ is a $p \times 1$ vector of predictor means. Let $\mathbf{y}$ be an n-vector of observed outcome values. LPS for y^* is defined as

$$\text{LPS}_{y^*} = -\log\{f(y^*|\mathbf{y},\mathbf{z}^*)\}$$

For a BMA linear model with a Zellner's g-prior for the regression coefficients and fixed g, $f(y^*|\mathbf{y},\mathbf{z}^*)$ is a 2^p mixture of t distributions. In this case, LPS can be calculated exactly, if we enumerate the entire model space (Fernández, Ley, and Steel 2001a),

$$\text{LPS}_{y^*} = -\log\left\{\sum_{j=1}^{2^p} P_a(M_j|\mathbf{y})f(y^*|M_j,\mathbf{y},\mathbf{z}_j^*)\right\}$$

where $P_a(M_j|\mathbf{y})$'s are the analytical PMPs defined by (7) in *Posterior model probability* in *Methods and formulas* of [BMA] **bmaregress** and $f(y^*|M_j,\mathbf{y},\mathbf{z}_j^*)$ is the conditional posterior predictive distribution, which is a t-distribution with $(n-1)$ degrees of freedom and location and scale given by (5) and (6) in *Conditional posterior predictive distribution* in *Methods and formulas* of [BMA] **bmaregress** with $q = 1$.

With fixed g and when model sampling is used, such as when bmaregress's `sampling` option is specified, the above formula is computed with respect to the set $J \subset \{1, 2, \ldots, 2^p\}$ of distinct models visited by the MCMC sampler.

With random g, we have an MCMC sample $\{g_t, m_t\}$ of size T from the joint g and model posterior distribution. Then, $f(y^*|\mathbf{y},\mathbf{z}^*)$ in the definition of LPS is replaced with the following frequency-based MCMC estimator,

$$\widehat{f}(y^*|\mathbf{y},\mathbf{z}^*) = \frac{1}{T}\sum_{t=1}^{T} f(y^*|g_t, m_t, \mathbf{y}, \mathbf{z}^*)$$

where $f(y^*|g_t, m_t, \mathbf{y}, \mathbf{z}^*)$ is the density of the location-scale t distribution with the respective mean and scale given by (5) and (6) in *Conditional posterior predictive distribution* in *Methods and formulas* of [BMA] **bmaregress** with $q = 1$ and j substituted for t. The bmastats lps command thus computes the following frequency-based LPS estimator:

$$\widehat{\text{LPS}}_{y^*} = -\log\{\widehat{f}(y^*|\mathbf{y},\mathbf{z}^*)\}$$

Fernández, Ley, and Steel (2001b) and Ley and Steel (2012) suggest exploring the summary statistics of LPS such as the mean, minimum, and maximum to evaluate the quality of the predictions.

The entropy of a continuous distribution with density $f(x)$ is defined to be $-E[\log\{f(x)\}]$. For a normal distribution with mean μ and variance σ^2, the entropy is

$$\frac{1}{2}\{1 + \log(2\pi\sigma^2)\}$$

bmastats lps reports the estimated entropy with σ^2 in the above replaced by the estimated posterior mean variance as described by (13) and (14) in *Posterior inclusion probability* in *Methods and formulas* of [BMA] **bmaregress**.

References

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.

Good, I. J. 1952. Rational decisions. *Journal of the Royal Statistical Society*, B ser., 14: 107–114. https://doi.org/10.1111/j.2517-6161.1952.tb00104.x.

Hastie, T. J., R. J. Tibshirani, and J. H. Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* 2nd ed. New York: Springer. https://doi.org/10.1007/978-0-387-84858-7.

Ley, E., and M. F. J. Steel. 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.

Piironen, J., and A. Vehtari. 2017. Comparison of Bayesian predictive methods for model selection. *Statistics and Computing* 27: 711–735. https://doi.org/10.1007/s11222-016-9649-y.

Rossouw, J. J., J. E. du Plessis, J. P. Benadé, A. J. S. Jordaan, P. C. J. Kotzé, J. P. Jooste, and P. L. Ferreira. 1983. Coronary risk factor screening in three rural communities. The CORIS baseline study. *South African Medical Journal* 64: 430–436. https://doi.org/10.10520/AJA20785135_9894.

Also see

Description

bmastats models provides summary information for the models visited by the bmaregress command, including their posterior model probabilities (PMPs), cumulative PMPs (CPMPs), ranks by PMP, and more. The variable-inclusion patterns for these models are also reported.

Quick start

Show summary statistics for the top five visited models with the highest PMPs

    bmastats models

Same as above, but report CPMPs and suppress the variable-inclusion table

    bmastats models, cumulative novartable

List models with CPMPs up to 0.9

    bmastats models, cumulative(0.9)

Same as above, but suppress all table output for brevity, and report only the header with the number of models

    bmastats models, cumulative(0.9) notable

List the top 10 models with the highest PMPs

    bmastats models, top(10)

List models ranked from 10 to 15 by PMP and the median probability model (MPM)

    bmastats models, ranks(10/15) mpm

Show summary statistics for models that include the two predictors x1 and x2

    bmastats models, include(x1 x2)

Same as above, but display only the header with the number of reported models

    bmastats models, include(x1 x2) notable

Menu

Statistics > Bayesian model averaging > Model and variable-inclusion summaries

195

Syntax

bmastats <u>model</u>s [, *modelopts varinclopts options*]

modelopts	Description
Main	
<u>top</u>(#)	show top # models with highest PMPs; default is top(5)
<u>ranks</u>(*numlist*)	show models ranked by highest PMPs with ranks in *numlist*
hpm	show highest probability model (HPM) with the highest posterior probability
mpm	show MPM having predictors with posterior inclusion probability (PIP) $\geq$ 0.5
<u>include</u>(*varlist*)	show models that include *varlist*
<u>cumul</u>ative	display CPMPs instead of PMPs for the specified models
<u>cumul</u>ative(#)	display models such that the last model has CPMP of at least # (if no other options specified)
<u>pmpc</u>utoff(#)	do not show models with PMP less than #; default is # = 0
<u>format</u>(%*fmt*)	use numerical format %*fmt* for PMP and CPMP values
[<u>no</u>]pmptable	display or suppress model-summary table; default is pmptable
all	show all models

varlist may contain factor variables; see [U] **11.4.3 Factor variables**.

all does not appear in the dialog box.

varinclopts	Description
Main	
<u>pipc</u>utoff(#)	do not show predictors with PIP less than #; default is pipcutoff(.01)
<u>display</u>(x)	indicate included predictors with an x; the default
<u>display</u>(u)	same as display(x), except predictors not included indicated with a u
sort(none)	order of predictors as originally specified; the default
sort(names)	order by the names of the variables
<u>noleg</u>end	suppress table legend
<u>nofvl</u>abel	display factor-variable level values rather than value labels
<u>nolst</u>retch	do not stretch the width of the table to accommodate long variable names
[<u>no</u>]<u>vart</u>able	display or suppress the variable-inclusion table; default varies

options	Description
Main	
<u>maxm</u>odels(#)	display results for the first # models; default is maxmodels(50)
<u>freq</u>uency	use frequency PMP estimates for model ranking; default is analytical PMPs (if available)
[<u>no</u>]<u>table</u>	display or suppress all tables; default varies

table and notable do not appear in the dialog box.

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

Options

top(*#*) specifies that the top *#* models with highest PMPs be shown. By default, the top five are shown. top(*#*) is equivalent to ranks(1/*#*).

ranks(*numlist*) specifies that models with ranks in *numlist* be included. Models are ranked by highest PMPs.

hpm specifies that the model with the highest PMP be shown. It is known as the HPM.

mpm specifies that the MPM be shown. MPM is the model in which all predictors have PIPs greater or equal to 0.5. The MPM may not always exist.

include(*varlist*) specifies that models that include all of *varlist* be shown. *varlist* may contain factor variables; see [U] **11.4.3 Factor variables**.

Options top(), ranks(), hpm, mpm, and include(), when specified together, provide results for all models that meet any of the specified criteria, that is, for the union of the models.

cumulative and cumulative(*#*) specify that CPMPs be reported for the models instead of the default PMPs. In the absence of other options that identify models such as ranks() or include(), cumulative(*#*) displays all models for which CPMPs are less than or equal to *#*. If there is no model with CPMP exactly equal to *#*, the model with the next highest CPMP is reported as the last model. If cumulative(*#*) is specified with other options, the results are reported for the intersection of the models provided by cumulative() and models provided by other options, regardless of whether the specified CPMP cutoff value is reached.

pmpcutoff(*#*) specifies that models with PMP less than *#* not be shown. The default is pmpcutoff(0). This option is useful when there are many models with small PMPs.

format(*%fmt*) specifies how numeric PMP and CPMP values are to be formatted. The default is format(%9.0g).

pmptable and nopmptable specify whether to display the model-summary table. The default is pmptable, which displays the table for the first 50 models. You can change the maximum number of displayed models by specifying the maxmodels() option.

pipcutoff(*#*) specifies that predictors with PIPs less than *#* not be shown in the variable-inclusion table. The default is pipcutoff(0.01). This option is useful when there are many predictors with small PIPs.

display(*displayspec*) specifies what to display in the variable-inclusion table. The default is display(x).

Blank cells in the table indicate that the corresponding variable or predictor was not included in the model.

For some predictors without estimated coefficients, a code that indicates the reason for omission is reported in the table.

Empty levels of factors and interactions are coded with the letter e.

Base levels of factors and interactions are coded with the letter b.

Variables omitted because of collinearity are coded with the letter o.

display(x) displays an x in the table when the variable or predictor has been included in the displayed model.

display(u) is the same as display(x), except that when a variable or predictor was not included in the model, u (for unavailable) is displayed instead of a blank cell.

sort(*sortspec*) specifies that the rows of the variable-inclusion table be ordered by specification given by *sortspec*.

> sort(none) specifies that the rows of the table be ordered by the order the predictors were specified in the model specification. This is the default.

> sort(names) orders rows alphabetically by the variable names of the predictors. For factor variables, main effects and nonfactor variables are displayed first in alphabetical order. Then, all two-way interactions are displayed in alphabetical order, then, all three-way interactions, and so on.

nolegend specifies that the legend at the bottom of the variable-inclusion table not be displayed. By default, it is shown.

nofvlabel displays factor-variable level numerical values rather than attached value labels. This option overrides the fvlabel setting. See [R] **set showbaselevels**.

nolstretch specifies that the width of the variable-inclusion table not be automatically widened to accommodate long variable names. When nolstretch is specified, names are abbreviated to make the table width no more than 79 characters. The default, lstretch, is to automatically widen the table up to the width of the Results window. To change the default, use set lstretch off.

vartable and novartable specify whether to display the variable-inclusion table. By default, the table is displayed when the number of the reported models does not exceed 12. You can specify vartable to display models beyond 12. The maximum number of models that will be displayed is 50, but you can change this by specifying the maxmodels() option.

maxmodels(#) specifies the maximum number of models to be displayed in model-summary and variable-inclusion tables. The default is to display the first 50 models.

frequency specifies that frequency estimates of PMPs based on a Markov chain Monte Carlo (MCMC) sample be used for model ranking when sampling is used by bmaregress. By default, analytical PMPs are used whenever they are available. Option frequency is not relevant with model enumeration, because frequency PMPs are not available. This option, however, is implied in the case of a random g parameter of Zellner's g-prior for regression coefficients, because analytical PMPs are not available in that case. See example 2 for details.

The following options are available with bmastats models but are not shown in the dialog box:

all specifies that all models, up to maxmodels(), be displayed. This option may be useful when the number of visited models is small.

table and notable display or suppress both the model-summary table and the variance-inclusion table. table implies pmptable and vartable. And notable implies nopmptable and novartable.

Remarks and examples

In the Bayesian model averaging (BMA) framework, inference includes exploring PMPs. Models with high PMPs and the predictors they include are often of interest. Clarke (2003), among others, comments that BMA puts the most weight on the model closest to the data-generating model whether the latter is in the explored model space or not. Thus, the models with the highest PMPs, HPMs, are often of interest.

Another model of potential interest is the MPM (Barbieri and Berger 2004). This model can be viewed as a model that includes only influential predictors, that is, predictors that are more likely than not to be included in the model. Such predictors have PIPs of at least 0.5. The MPM may not always exist.

By default, bmastats models reports the top five models with the highest PMPs and the predictors included in each of these models. You can use the top() option to display more top models and ranks() to display models with specific ranks. You can use options hpm and mpm to display the respective HPM and MPM. And you can use the include(*varlist*) option to display models that contain the specified predictors in *varlist* and their respective PMP rankings. When you combine all of these options, the union of the models is displayed.

The command reports PMPs by default, but you can specify the cumulative option to report CPMPs instead. The cumulative(#) option is useful if you wish to see the models with CPMPs up to #. This option (as well as the include() option), however, may report many models, especially for high # values. You may want to combine it with the notable option to see only the number of models to be reported first. You can also use the pmpcutoff(#) option to limit the models to those with PMPs of at least #.

bmastats models reports two tables for the models: one with the PMPs and the other one with variable-inclusion patterns. The variable-inclusion table can be rather lengthy depending on the number of included predictors. By default, only predictors with PIPs above 0.01 are displayed, but you can change this with the pipcutoff() option. Also, this table is suppressed when the number of reported models exceeds 12. You can use the vartable option to display more models. The maximum number of reported models for both tables is 50, but you can change this by specifying maxmodels().

See example 1 for various uses of bmastats models's options.

bmastats models reports analytical and frequency PMPs, whenever they are available. Analytical PMPs are computed using analytical formulas. They are reported only with a fixed g because analytical formulas are not available with a random g. Frequency PMPs are computed from an MCMC sample of models, which is available only when sampling is used by bmaregress. That is, they are not available with model enumeration. Analytical PMPs are used for model ranking whenever they are available. See example 2.

▷ Example 1: Tour of the bmastats models command

Recall the performance dataset (Chatterjee and Hadi 2012, sec. 3.3) analyzed in example 1 of [BMA] **bmaregress**, where the employee satisfaction with their supervisors, rating, is modeled by six potential predictors.

Let's fit a linear BMA regression to these data using bmaregress and explore various model summaries by using bmastats models.

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

. 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.

. 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` reports the top five models ranked by PMPs and their respective variable-inclusion summary. We see that the top model, also known as HPM, has a PMP of 0.56 and includes only one predictor, `complaints`. This is not surprising because, from the output of `bmaregress`, predictor `complaints` has by far the highest PIP, 0.9997, of all predictors. The next highest-ranking model, with a PMP of 0.12, includes both `complaints` and `learning`, the predictor with the next highest PIP, 0.25. The rest of the predictors have similar probabilities, which are less than 0.15.

We can display more than five top models by specifying the `top()` option. Here, to demonstrate, we list only the top six models to keep the output short.

```
. bmastats models, top(6)
Computing model probabilities ...
Model summary                 Number of models:
                                       Visited = 64
                                      Reported =  6
```

	Analytical PMP	Model size
Rank		
1	.5556	1
2	.1169	2
3	.04072	2
4	.03932	2
5	.03804	2
6	.03654	2

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

The above is equivalent to ranks(1/6), or you can specify any list of ranks if desired.

```
. bmastats models, ranks(1/6)
Computing model probabilities ...
Model summary                 Number of models:
                                    Visited = 64
                                    Reported =  6
```

	Analytical PMP	Model size
Rank		
1	.5556	1
2	.1169	2
3	.04072	2
4	.03932	2
5	.03804	2
6	.03654	2

```
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
```

We can also use options hpm and mpm to display the respective HPM and MPM.

```
. bmastats models, hpm mpm
Computing model probabilities ...
Model summary                 Number of models:
                                    Visited = 64
                                    Reported =  1
```

	Analytical PMP	Model size
Rank		
(HPM) 1	.5556	1

```
Note: HPM and MPM are the same model.
Variable-inclusion summary
```

	(HPM) 1
complaints	x

```
Legend:
  x - estimated
```

In our example, HPM and MPM are the same.

Instead of PMPs, we can use the `cumulative` option to display CPMPs.

```
. bmastats models, cumulative
Computing model probabilities ...
Model summary                  Number of models:
                                      Visited = 64
                                     Reported =  5
```

	Analytical CPMP	Model size
Rank		
1	.5556	1
2	.6724	2
3	.7132	2
4	.7525	2
5	.7905	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
```

You might also want to see the models that contribute to a specific, typically larger, CPMP value. For instance, let's use 0.9 for the CPMP cutoff here.

```
. 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
```

There are 10 models that contribute to the CPMP of at least 0.9. This is the same number as reported in the bmaregress header output under For CPMP >= 0.9. This number is useful to determine whether there are only a few high-probability models that are consistent with your data or there are many different models that are plausible.

Beware that cumulative() may report many models, especially for high CPMP values. In that case, to keep the table output manageable, bmastats models will display only the first 50 regardless of the specified CPMP value. But you can use the maxmodels() option to display more. You might also consider

using the `notable` option to suppress the table output altogether, if you just want to see the number of models corresponding to a specific CPMP value. We demonstrate this later in combination with the `include()` option, which also may lead to many models being displayed.

In fact, for brevity, the variance-inclusion table is suppressed when the number of reported models exceeds 12, as we see in the output.

```
. bmastats models, cumulative(0.95)

Computing model probabilities ...

Model summary                Number of models:
                                     Visited = 64
                                    Reported = 16
```

	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
11	.9172	4
12	.9255	4
13	.9333	4
14	.9396	3
15	.9452	3
16	.9503	3

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

But you can specify the `vartable` option to see those models.

```
. bmastats models, cumulative(0.95) vartable
Computing model probabilities ...
Model summary              Number of models:
                                   Visited = 64
                                  Reported = 16
```

	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
11	.9172	4
12	.9255	4
13	.9333	4
14	.9396	3
15	.9452	3
16	.9503	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	Rank 11	Rank 12
complaints	x	x	x	x	x	x
learning	x	x	x	x	x	x
raises			x			x
privileges		x			x	
advance	x				x	x
critical				x		

Legend:
  x - estimated

	Rank 13	Rank 14	Rank 15	Rank 16
complaints	x	x	x	x
learning	x			
raises		x	x	x
privileges			x	
advance	x	x		
critical	x			x

Legend:
  x - estimated

We can explore the models that contain specific predictors by using the include() option. We use the complaints variable. We also specify the notable option to see only the number of models to be reported first.

```
. bmastats models, include(complaints) notable
Computing model probabilities ...
Model summary              Number of models:
                                   Visited = 64
                                  Reported = 32
```

There are 32 different models that include complaints. We probably do not want to display all 32 models, because many of them will have a low probability. We can use the pmpcutoff() option to display only models with higher probabilities.

```
. bmastats models, include(complaints) pmpcutoff(0.01)
Computing model probabilities ...
Model summary                 Number of models:
                                     Visited = 64
                                    Reported = 10
```

	Analytical PMP	Model size
Rank		
1	.5556	1
2	.1169	2
3	.04072	2
4	.03932	2
5	.03804	2
6	.03654	2
7	.03272	3
8	.01985	3
9	.01459	3
10	.01441	3

Note: 22 models with PMP less than .01 not shown.

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 could further explore a few higher-probability models in more detail by specifying their ranks in the ranks() option and looking at their variable-inclusion summary. The top two models above are the same ones we already explored earlier: the one containing complaints and the other one containing complaints and learning.

◁

▷ Example 2: Analytical and frequency PMPs

In this example, we explore the analytical and frequency PMPs as reported by bmastats models.

Let's revisit example 1, where model enumeration was used for estimation. In what follows, we will use the novartable option with bmastats models to focus only on the reported PMPs.

```
. 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.
. bmastats models, novartable
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

With a small number of predictors, bmaregress explores a full space of all possible models, so no sampling is performed. In this case, bmastats models reports only analytical PMPs.

With many predictors, a full model enumeration may not be feasible, and thus a sampling algorithm is used to explore the model space. To demonstrate, let's use sampling instead of the default model enumeration in our example by specifying the sampling option with bmaregress.

```
. 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.

. bmastats models, novartable

Computing model probabilities ...

Model summary                               Number of models:
                                                      Visited = 32
                                                      Reported =  5
```

	Analytical PMP	Frequency PMP	Model size
Rank			
1	.5557	.5167	1
2	.1169	.1248	2
3	.04073	.0427	2
4	.03933	.0392	2
5	.03805	.0492	2

```
Note: Using analytical PMP for model ranking.
```

Analytical and frequency PMPs are available in this case, and bmastats models reports both. The analytical and frequency estimates are similar, which should be the case for a converged model. bmaregress reported a sampling correlation of 0.9990, which is a strong indication of convergence. When both types of PMP estimates are available, bmastats models always uses analytical PMPs for model ranking, but you can use the frequency option to rank models by frequency PMPs.

Both BMA models above assumed a fixed parameter g that controls the shrinkage of coefficients toward zero. We can instead specify a prior distribution, a hyperprior, for g. For instance, we use a robust hyperprior below.

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

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

```
. bmastats model, novartable

Computing model probabilities ...
```

Model summary		Number of models:	
		Visited =	34
		Reported =	5

		Frequency PMP	Model size
Rank			
	1	.5708	1
	2	.102	2
	3	.0403	2
	4	.04	2
	5	.0305	2

Only frequency PMPs are reported in this case because analytical formulas for PMPs are not available for a random g.

◁

Stored results

bmastats models stores the following in r():

Scalars
  r(k_models)   number of models
  r(k_models_rpt)   number of reported models

Macros
  r(pmptype)   analytical or frequency

Matrices
  r(summary)   model summary matrix
  r(rank)   model ranks
  r(varinclmat)   variable-inclusion matrix

Methods and formulas

For methods and formulas of PMPs and PIPs, see *Posterior model probability* and *Posterior inclusion probability* in *Methods and formulas* of [BMA] **bmaregress**.

The CPMP for a model j is a cumulative sum of the first jth highest PMPs. The analytical CPMP is a cumulative sum of analytical PMPs, and frequency CPMP is a cumulative sum of frequency PMPs.

The HPM is the model with the highest PMP.

The MPM is the model with predictors that have PIPs greater than or equal to 0.5 (Barbieri and Berger 2004). That is, if $X_1, X_2, \ldots, X_p$ are predictors used with bmaregress, then X_k is in the MPM if and only if $\text{PIP}(X_k) \geq 0.5$. If there are no such predictors, then MPM does not exist.

References

Barbieri, M. M., and J. O. Berger. 2004. Optimal predictive model selection. *Annals of Statistics* 32: 870–897. https://doi.org/10.1214/009053604000000238.

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

Clarke, B. 2003. Comparing Bayes model averaging and stacking when model approximation error cannot be ignored. *Journal of Machine Learning Research* 4: 683–712.

Also see

Description

bmastats msize provides a model-size summary after the bmaregress command. bmastats msize is useful to assess the overall complexity of the models in Bayesian model averaging (BMA) weighted by their prior and posterior model probabilities. By comparing the overall posterior model sizes with the prior model sizes, we can also assess the impact of the data on the BMA analysis.

Quick start

Display prior and posterior model-size summary

    bmastats msize

Include a constant term in the model-size computations

    bmastats msize, constant

Menu

Statistics > Bayesian model averaging > Model-size summary

Syntax

    bmastats msize [, constant]

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

Option

> Main

constant specifies that the constant term be included in model-size computations. By default, the constant term is not included.

Remarks and examples

Model size is the number of predictors included in a model. The constant term is typically not included in the model size. In BMA, there are multiple models, so we have a distribution of the model sizes. The model prior determines the prior model-size distribution. After you observe the data, the model prior is updated to form the posterior model-size distribution.

You can use the model-size distribution summaries to explore the complexity of the fitted BMA model. For instance, when the posterior median model size is small relative to the total number of regression terms, this means there are only a few strong predictors of the outcome. Conversely, when it is large, there are many weak potential predictors. And by comparing the posterior model size with the prior one, we can assess how the data affect our prior knowledge.

213

The prior model-size distribution can be computed analytically based on the known formula, with a caveat that in cases when the model space is not explored fully, this analytical distribution is conditional on the visited models.

The posterior model-size distribution is available analytically only with fixed g and is thus not provided with random g.

Whenever a posterior Markov chain Monte Carlo (MCMC) model sample is available, such as with random g or when bmaregress's `sampling` option is used, the posterior model-size distribution can be estimated from the sample by using sampling frequencies. See *Methods and formulas* for details.

bmastats msize reports prior and posterior model-size summaries, including mean and median model sizes. For prior model-size summary, it always provides the analytical estimates. For posterior model-size summary, it provides the analytical estimates with fixed g and the frequency estimates with random g. With fixed g and MC3 sampling, the command provides both types of estimates. By default, the constant is not included in the model-size computations, but you can specify the `constant` option to include it.

▷ Example 1: Model-size analysis after BMA regression using enumeration

Consider the performance dataset (Chatterjee and Hadi 2012, sec. 3.3) analyzed in example 1 of [BMA] **bmaregress**. The employees' satisfaction with their supervisors, rating, is modeled by six potential predictors. The total of 30 observations represent 30 different departments in the surveyed organization.

We fit a linear BMA regression using the `bmaregress` command.

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

From the output, the model space is fully explored through enumeration. There is a total of $2^6 = 64$ models in the full model space.

Let's use `bmastats msize` to compute the model-size summary.

```
. 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 reported model size does not include the constant, so its range is between 0 and 6. The prior mean and median model sizes are both 3. The posterior mean and median model sizes are, respectively, 1.699 and 1. (The posterior mean model size is the same as the one reported by `bmaregress`.)

The default model prior distribution assumes a uniform prior on the model size. The prior mean model size, 3, is larger than the posterior one, 1.699. The posterior favors smaller models.

For the prior model-size summary, bmastats msize always reports analytical estimates. With model enumeration, it reports analytical estimates for the posterior model-size summary too. Frequency-based posterior estimates are not available here because there is no MCMC sample for the models—the models were enumerated.

◁

▷ Example 2: Model-size analysis after BMA regression using MC3 sampling

We fit the same BMA model as in example 1, but this time we use the MC3 sampling algorithm.

```
. 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.
```

Instead of enumerating models, bmaregress generates a sample from the posterior model distribution that includes 32 different models.

```
. bmastats msize

Model-size summary

Number of models = 32
Model size:
   Minimum = 1
   Maximum = 6
```

	Mean	Median
Prior		
Analytical	4.3333	5
Posterior		
Analytical	1.6985	1
Frequency	1.7791	1

Here the minimum model size is 1, compared with 0 in example 1. This means that the null model, having a low posterior probability, was not visited by the MC3 sampler.

Although we used the same model prior as in example 1, the prior model-size estimates are different. This is because our explored model space now contains 32 models instead of all 64, and the prior model-size estimates are now conditional on the visited models.

With fixed g and when we fit a BMA model using MC3 sampling, in addition to analytical model-size estimates, the frequency estimates are also available. Provided that the model-space sampling converges, the analytical and frequency estimates should be close. In our example, the analytical and frequency model-size estimates, 1.7 and 1.8, are close.

We can also explore the effect of the g parameter on the complexity of our BMA model. Let us, for example, fix g to 1, which is much lower than the default value of 36 used above.

```
. bmaregress rating complaints-advance, gprior(fixed 1) 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       =        63
                                               For CPMP >= .9 =         29
Priors:                                     Mean model size     =     3.731
  Models: Beta-binomial(1, 1)               Burn-in             =     2,500
   Cons.: Noninformative                    MCMC sample size    =    10,000
   Coef.: Zellner's g                       Acceptance rate     =    0.5678
       g: g = 1                             Shrinkage, g/(1+g)  =    0.5000
  sigma2: Noninformative                    Mean sigma2         =   103.952

Sampling correlation = 0.9957
```

rating	Mean	Std. dev.	Group	PIP
complaints	.3188165	.158556	1	.94416
learning	.0944454	.1508832	3	.61347
advance	−.0520093	.137935	6	.55788
raises	.030129	.1654381	4	.5473
privileges	−.0163312	.1027766	2	.53808
critical	.0062495	.1066877	5	.52978
Always				
_cons	38.76265	10.36275	0	1

```
Note: Coefficient posterior means and std. dev. estimated from 63 models.
Note: Default prior is used for models.

. bmastats msize

Model-size summary

Number of models = 63
Model size:
  Minimum = 0
  Maximum = 6
```

	Mean	Median
Prior		
Analytical	3.0488	3
Posterior		
Analytical	3.7307	4
Frequency	3.6529	4

The posterior mean model size has increased to 3.73, and the posterior median model size has increased to 4. With $g = 1$, BMA appears to favor larger models.

◁

▷ Example 3: Model-size analysis after BMA regression with random g

Both example 1 and example 2 used a fixed g. Let's explore the case of a random g. (An in-depth coverage of the effects of the g-prior on model complexity can be found in, for example, Ley and Steel [2012].)

To demonstrate, we will use a robust prior for g.

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

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

`bmaregress` now uses MC3 sampling for the models and an adaptive Metropolis–Hastings sampling for g.

```
. bmastats msize
Model-size summary
Number of models = 34
Model size:
   Minimum = 1
   Maximum = 6
```

	Mean	Median
Prior		
Analytical	4.1786	4
Posterior		
Frequency	1.7338	1

```
Note: Analytical summaries not available.
```

Analytical posterior estimates are not available with random g. The frequency posterior estimates are similar to those in example 2 for fixed $g = 36$.

◁

Stored results

`bmastats msize` stores the following in `r()`:

Scalars
r(k_models)	number of models
r(msize_mean_prior)	prior mean model size
r(msize_mean_a)	analytical posterior mean model size (not available with random g)
r(msize_mean_f)	frequency posterior mean model size (not available with model enumeration)
r(constant)	1 if constant is specified; 0 otherwise

Matrices
r(modelsize)	model-size summary

Methods and formulas

Consider a BMA regression model for an outcome vector $\mathbf{y}$ 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 $|M|$ denote the model size of a regression model M from $\mathcal{M}_F$; that is, $|M|$ equals the number of predictors included in model M.

The prior distribution of $|M|$ is a discrete distribution on the set $\{0, 1, \ldots, p\}$ such that

$$\Pr(|M| = s) = \sum_{j=1}^{2^p} I(|M_j| = s) P(M_j)$$

where $I(\cdot)$ is an indicator function, M_j's are all possible enumerated models from $\mathcal{M}_F$, and $P(M_j)$'s are the prior model probabilities.

Similarly, the posterior distribution of $|M|$ is defined as

$$\Pr(|M| = s | \mathbf{y}) = \sum_{j=1}^{2^p} I(|M_j| = s) P_a(M_j | \mathbf{y})$$

where $P_a(M_j | \mathbf{y})$'s are the analytical posterior model probabilities defined by (7) in *Posterior model probability* in *Methods and formulas* of [BMA] **bmaregress**.

When the model space is fully explored through enumeration, the analytical prior mean model size is

$$E(|M|) = \sum_{j=1}^{2^p} |M_j| P(M_j)$$

and the analytical posterior mean model size is

$$E(|M| \,|\, \mathbf{y}) = \sum_{j=1}^{2^p} |M_j| P_a(M_j | \mathbf{y})$$

When model sampling is used instead of model enumeration, the analytical prior mean model size is estimated conditionally on the subspace of the visited models indexed by $J \subset J_F$:

$$\widehat{E}(|M|) = \frac{\sum_{j \in J} |M_j| P(M_j)}{\sum_{j \in J} P(M_j)}$$

With fixed g and when model sampling is used, such as when bmaregress's sampling option is specified, the analytical posterior mean model size is estimated as

$$\widehat{E}_a(|M| \,|\, \mathbf{y}) = \frac{\sum_{j \in J} |M_j| P_a(M_j | \mathbf{y})}{\sum_{j \in J} P_a(M_j | \mathbf{y})}$$

With random g, analytical formulas for posterior model probabilities and posterior model-size probabilities are not available.

When a posterior MCMC sample of models, $\{m_t\}_{t=1}^T$, is available, such as with random g or when bmaregress's sampling option is used, the frequency estimate of the posterior mean model size is computed as follows:

$$\widehat{E}_f(|M| \,|\, \mathbf{y}) = \frac{1}{T} \sum_{t=1}^T |m_t|$$

References

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

Ley, E., and M. F. J. Steel. 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.

Also see

[BMA] **bmagraph msize** — Model-size distribution plots after BMA regression

[BMA] **bmastats** — Summary for models and predictors after BMA regression

[BMA] **bmaregress** — Bayesian model averaging for linear regression

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

[BMA] **Glossary**

Description

bmastats pip reports posterior inclusion probabilities (PIPs) and grouping information for predictors specified with the bmaregress command.

Quick start

Show predictors with PIP above 0.01, ordered by highest to lowest PIP

    bmastats pip

Same as above, but report predictors in their input order in bmaregress

    bmastats pip, inputorder

Show PIP for predictors x1 and x2

    bmastats pip x1 x2

Show predictors with PIP above 0.5

    bmastats pip, cutoff(0.5)

Show PIP for all predictors

    bmastats pip, all

Menu

Statistics > Bayesian model averaging > Posterior inclusion probabilities

Syntax

> bmastats pip [*varlist*] [, *options*]

options	Description
Main	
<u>inputorder</u>	display results in a variable input order; default is descending order of PIP
<u>cut</u>off(*#*)	suppress predictors with PIP less than *#*; default is *# = 0.01*
display_options	control spacing, line width, and base and empty cells
all	show PIP for all predictors
[<u>no</u>]table	display or suppress table output

varlist may contain factor variables; see [U] **11.4.3 Factor variables**.

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

all, table, and notable do not appear in the dialog box.

Options

> ┌─── Main ──

inputorder specifies that the results be displayed in the same order the predictors are specified in *varlist* or, if *varlist* is not specified, with bmaregress. By default, the results are displayed in the descending order of PIP of predictors.

cutoff(*#*) specifies that predictors with PIP less than *#* not be shown. The default is cutoff(0.01). This option is useful when there are many predictors with small PIPs.

display_options: vsquish, <u>noemptycells</u>, <u>base</u>levels, <u>allbase</u>levels, <u>nofvlab</u>el, fvwrap(*#*), fvwrapon(*style*), and <u>nolstretch</u>; see [R] **Estimation options**.

The following options are available with bmastats pip but are not shown in the dialog box:

all shows PIPs and grouping information for all predictors or for all predictors in *varlist* if specified with bmastats pip. all is equivalent to cutoff(0).

table and notable display or suppress the table output. The table is shown by default. This option is useful with many predictors when you wish to see only the number of reported predictors in the header.

Remarks and examples

PIPs are used in Bayesian model averaging (BMA) inference to investigate the importance of various predictors in explaining the outcome. A PIP is the probability that a predictor is included in a regression model, given the observed data and prior inclusion probability. High PIP values indicate strong (more important) predictors, and low PIP values indicate weak predictors.

bmaregress reports PIPs as part of its output. bmastats pip provides the same PIP and grouping information but displays it more concisely—without the posterior coefficient summaries. bmastats pip also allows you to investigate PIPs of specific predictors more easily.

Let's see an example.

We will use the `auto` dataset to fit a BMA linear regression of car prices on various car characteristics such as mileage, weight, and whether a car is foreign or domestic. And we will use `bmastats pip` to explore PIPs.

```
. use https://www.stata-press.com/data/r19/auto
(1978 automobile data)

. order make price foreign rep78

. describe

Contains data from https://www.stata-press.com/data/r19/auto.dta
 Observations:              74                  1978 automobile data
    Variables:              12                  13 Apr 2024 17:45
                                                (_dta has notes)
```

Variable name	Storage type	Display format	Value label	Variable label
make	str18	%-18s		Make and model
price	int	%8.0gc		Price
foreign	byte	%8.0g	origin	Car origin
rep78	int	%8.0g		Repair record 1978
mpg	int	%8.0g		Mileage (mpg)
headroom	float	%6.1f		Headroom (in.)
trunk	int	%8.0g		Trunk space (cu. ft.)
weight	int	%8.0gc		Weight (lbs.)
length	int	%8.0g		Length (in.)
turn	int	%8.0g		Turn circle (ft.)
displacement	int	%8.0g		Displacement (cu. in.)
gear_ratio	float	%6.2f		Gear ratio

```
Sorted by: foreign
```

```
. bmaregress price i.foreign##i.rep78 mpg-gear_ratio, rseed(18)

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

Bayesian model averaging			No. of obs	=	69
Linear regression			No. of predictors	=	16
MC3 sampling			Groups	=	16
Heredity: Strong			Always	=	0
			No. of models	=	105
			For CPMP >= .9	=	14
Priors:			Mean model size	=	2.625
Models: Beta-binomial(1, 1)			Burn-in	=	2,500
Cons.: Noninformative			MCMC sample size	=	10,000
Coef.: Zellner's g			Acceptance rate	=	0.2686
g: Benchmark, g = 256			Shrinkage, g/(1+g)	=	0.9961
sigma2: Noninformative			Mean sigma2	=	4.239e+06

Sampling correlation = 0.9962

price	Mean	Std. dev.	Group	PIP
foreign				
Foreign	3597.21	733.8864	1	.99846
weight	3.504294	2.315263	12	.78147
length	-33.74574	49.96541	13	.35377
displacement	6.896085	11.95028	15	.27357
headroom	-32.06511	153.7326	10	.0575
turn	-6.257741	40.06633	14	.037379
gear_ratio	-40.80854	290.5467	16	.031843
rep78				
5	12.02161	138.5304	5	.020152
trunk	-.3727775	12.85491	11	.018163
rep78				
4	-4.524674	82.48986	4	.015373
3	1.565563	67.56235	3	.013669
2	-1.818892	97.07111	2	.013639
Always				
_cons	-115.0626	5192.111	0	1

Note: Coefficient posterior means and std. dev. estimated from 105 models.
Note: Default priors are used for models and parameter *g*.
Note: 4 predictors with PIP less than .01 not shown.

Let's start with the default specification:

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

No. of obs          =      69
No. of predictors   =      16
          Groups =      16
          Always =       0
        Reported =      12
No. of models       =     105
Mean model size     = 2.625
```

	PIP	Group
foreign		
Foreign	.99846	1
weight	.78147	12
length	.35377	13
displacement	.27357	15
headroom	.0575	10
turn	.037379	14
gear_ratio	.031843	16
rep78		
5	.020152	5
trunk	.018163	11
rep78		
4	.015373	4
3	.013669	3
2	.013639	2
Always		
_cons	1	0

```
Note: Using analytical PMPs.
Note: 4 predictors with PIP less than
      .01 not shown.
```

As we already mentioned, by default, bmastats pip reports the same PIP and grouping information as bmaregress.

By default, the results are displayed in the decreasing order of PIP. We can use the `inputorder` option to display the results in the order in which the predictors were specified with `bmaregress`. The command also suppresses results for predictors with PIP less than 0.01. If desired, we can use the `all` option to display the results for all predictors.

```
. bmastats pip, inputorder all

Posterior inclusion probability (PIP)

No. of obs        =      69
No. of predictors =      16
           Groups =      16
           Always =       0
         Reported =      16
No. of models     =     105
Mean model size   =   2.625
```

	PIP	Group
foreign		
Foreign	.99846	1
rep78		
2	.013639	2
3	.013669	3
4	.015373	4
5	.020152	5
foreign#rep78		
Foreign#1	(empty)	
Foreign#2	(empty)	
Foreign#3	0	6
Foreign#4	.0002665	7
Foreign#5	0	8
mpg	.0094456	9
headroom	.0575	10
trunk	.018163	11
weight	.78147	12
length	.35377	13
turn	.037379	14
displacement	.27357	15
gear_ratio	.031843	16
Always		
_cons	1	0

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

We can identify a set of more influential predictors by specifying a higher PIP cutoff:

```
. bmastats pip, cutoff(0.5)
Posterior inclusion probability (PIP)

No. of obs         =      69
No. of predictors  =      16
          Groups   =      16
          Always   =       0
        Reported   =       2
No. of models      =     105
Mean model size    =   2.625
```

	PIP	Group
foreign		
Foreign	.99846	1
weight	.78147	12
Always		
_cons	1	0

```
Note: Using analytical PMPs.
Note: 14 predictors with PIP less than
      .5 not shown.
```

And we can investigate PIPs for specific predictors:

```
. bmastats pip i.rep78
Posterior inclusion probability (PIP)

No. of obs         =      69
No. of predictors  =      16
          Groups   =      16
          Always   =       0
        Reported   =       4
No. of models      =     105
Mean model size    =   2.625
```

	PIP	Group
rep78		
5	.020152	5
4	.015373	4
3	.013669	3
2	.013639	2
Always		
_cons	1	0

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

Stored results

bmastats pip stores the following in r():

Scalars
  r(N)   number of observations
  r(p)   number of predictors
  r(p_groups)   number of groups of predictors
  r(p_always)   number of predictors always in the model
  r(p_rpt)   number of reported predictors
  r(k_models)   number of models
  r(msize_mean)   posterior mean model size

Macros
  r(varnames)   specified variables
  r(pmptype)   analytical or frequency

Matrices
  r(summary)   PIP and group summary

Methods and formulas

For definitions of PIPs, see *Posterior inclusion probability* in *Methods and formulas* of [BMA] **bmaregress**.

Also see

[BMA] **bmagraph varmap** — Variable-inclusion map after BMA regression

[BMA] **bmastats** — Summary for models and predictors after BMA regression

[BMA] **bmaregress** — Bayesian model averaging for linear regression

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

[BMA] **Glossary**

Glossary

See [BAYES] **Glossary** for definitions standard to Bayesian analysis.

always predictor. Predictor that is always included in a model, that is, included in every model in the model space. In bmaregress, it is specified with the command within the always group of variables:

> . bmaregress (*varlist*, always) ...

analytical mean model size. See *mean model size*.

analytical model-size distribution. See *model-size distribution*.

analytical PIP. See *posterior inclusion probability*.

analytical PMP. See *posterior model probability (PMP)*.

Bayesian model averaging (BMA). A special case of model averaging, where the weights correspond to posterior model probabilities. BMA considers a space of plausible models and views a model as a discrete random variable over this space. It uses the Bayes theorem to estimate the posterior model probability based on the model prior and observed data. The posterior distribution of model parameters is then estimated by an average over the posterior distributions conditional on a model weighted by the corresponding posterior model probability.

Bayesian model averaging (BMA) regression. BMA applied to regression analysis, where each model from the model space corresponds to a unique set of predictors. Model uncertainty in the context of Bayesian model averaging regression amounts to the uncertainty of the inclusion of predictors in a model. See [BMA] **bmaregress**.

bivariate inclusion probability. See *joint inclusion probability*.

bivariate jointness. See *jointness*.

BMA. See *Bayesian model averaging (BMA)*.

coefficient sample. See *model parameter sample*.

complements. See *jointness*.

CPMP. See *cumulative posterior model probability (CPMP)*.

cumulative posterior model probability (CPMP). The running sum of the highest to lowest PMP of models. See [BMA] **bmastats models**.

disjointness. A tendency of the two predictors to be included in a model exclusively; that is, if one is included, the other is not. Such predictors are known as *substitutes*, which means that each of them carries the same amount of information about the outcome. There are several measures proposed to estimate disjointness, see [BMA] **bmastats jointness**. Also see *jointness*.

entropy. The negative of the mean of the log density. For a normal distribution with mean μ and variance σ^2, it is defined as $\{1 + \log(2\pi\sigma^2)\}/2$. It is used for comparison with the mean log predictive-score (LPS) when checking model fit using out-of-sample observations. When the variance is known (see bmastats lps's sigma2() option), the comparison can also be made for in-sample observations. The closer the mean LPS to the entropy, the better the model fit. See [BMA] **bmastats lps**.

enumeration. See *model enumeration*.

explored model space. See *visited model space*.

fixed g, **fixed-g case**. A case when a fixed value is used for the g parameter of a Zellner's g-prior. This is the case when one of the following is specified in the gprior() option of bmaregress: bench (the default), uip, ric, sqrtn, fixed, or ebl. Also see random g.

frequency mean model size. See *mean model size*.

frequency model-size distribution. See *model-size distribution*.

frequency PIP. See *posterior inclusion probability (PIP)*.

frequency PMP. See *posterior model probability (PMP)*.

fully explored model space. See *model enumeration*.

full model space. An entire model space as defined for a Bayesian model averaging model.

g-prior, **g-hyperprior**. A prior for the g parameter; see *random g* and *fixed g*.

grouped predictors. See *in–out group*.

highest probability model (HPM). Model with the highest posterior model probability. Clarke (2003), among others, states that the HPM is the model closest to the data-generating model whether the latter is in the explored model space or not. See [BMA] **bmastats models**.

HPM. See *highest probability model (HPM)*.

hyperprior. A distribution for a parameter of a prior distribution. In the context of a Bayesian model averaging regression, this is a prior distribution for the g parameter; see *g-prior*.

important predictor. See *predictor*.

in–out group. A group of predictors that is included in or excluded from a model together during estimation. In bmaregress, it is specified with the command in parentheses:

    . bmaregress (*varlist*) ...

in–out predictor, **in–out variable**. A predictor (variable) that is included in or excluded from a model during estimation. Also see *always predictor* and [BMA] **bmaregress**.

in–out term. An in–out predictor (*varname*) or an in–out group of predictors (*varlist*); see [BMA] **bmaregress**.

inclusion map. See *variable-inclusion map*.

inclusion probability. In the context of a Bayesian model averaging regression, a probability over the considered model space that a predictor is included in a regression model. This is known as a marginal inclusion probability. Also see *joint inclusion probability* and *posterior inclusion probability (PIP)*.

influential predictor. See *predictor*.

joint inclusion probability. In the context of a Bayesian model averaging regression, a joint probability of inclusion of multiple predictors in a model. This probability is defined over the considered model space. For example, for a pair of predictors X_1 and X_2 and a model M, a bivariate inclusion probability is defined as $P(X_1 \in M, X_2 \in M)$. If the inclusion of X_1 is independent of X_2, then the bivariate joint probability is simply the product of the corresponding marginal inclusion probabilities, $P(X_1 \in M, X_2 \in M) = P(X_1 \in M)P(X_2 \in M)$. Similarly, one can define a joint noninclusion probability as $P(X_1 \notin M, X_2 \notin M)$. The other two joint probabilities of interest are $P(X_1 \in M, X_2 \notin M)$ and $P(X_1 \notin M, X_2 \in M)$. You can view these four probabilities as

probabilities corresponding to the cross-tabulation of two random binary indicators $I(X_1 \in M)$ and $I(X_2 \in M)$, where $I(\cdot)$ is an indicator function. The joint probabilities are used to define various jointness measures. See [BMA] **bmastats jointness**.

joint inference. In the context of Bayesian model averaging, joint inference considers joint inclusion probabilities and, more specifically, various jointness measures. Also see *marginal inference*.

joint noninclusion probability. See *joint inclusion probability*.

jointness. A tendency of the two predictors to be included in a model together; that is, if one is included, the other is included too. Such predictors are known as *complements*, which means that, when included together, they provide additional information about the outcome. There are several measures proposed to estimate jointness; see [BMA] **bmastats jointness**. Also see *disjointness*.

jointness measures. Jointness measures assess the degree of inclusion dependency between predictors across the considered model space. They are typically defined for a pair of predictors and are based on the joint inclusion probabilities. See *jointness* and *disjointness*. Also see [BMA] **bmastats jointness**.

log predictive-score (LPS). The negative of the log of the posterior predictive density evaluated at an observation. It is often used to evaluate the model's predictive performance. The smaller the LPS, the better the predictive performance. Also see *entropy* and [BMA] **bmastats lps**.

LPS. See *log predictive-score (LPS)*.

marginal inference. In the context of Bayesian model averaging, marginal inference explores individual characteristics of predictors such as posterior inclusion probability. Also see *joint inference*.

MC3. See *Markov chain Monte Carlo model composition (MC3) algorithm*.

Markov chain Monte Carlo model composition (MC3) algorithm, **MC3 sampling**. MC3 (Madigan and York 1995) is a stochastic algorithm used in Bayesian model averaging to sample models from their posterior distribution over the model space. It explores discrete model spaces formed by subsets of potential predictors by changing one variable, or group of variables, at a time. See [BMA] **bmaregress**.

MCMC model sample. See *model sample*.

MCMC model parameter sample. See *model parameter sample*.

mean model size. The mean of the model-size distribution. In the Bayesian model averaging context, there are prior and posterior mean model sizes. The prior mean model size is the mean of the prior model-size distribution. The posterior mean model size is the mean of the posterior model-size distribution. Posterior mean model size can be computed by using analytical or frequency posterior model probability, in which case it is referred to as, respectively, the analytical or frequency posterior mean model size. The prior mean model size is always computed analytically, but cases when the model space is not explored fully, it is conditional on the visited models. It is often compared with the posterior mean model size to evaluate the impact of the data on the prior assumption about the model size. See [BMA] **bmastats msize**.

median probability model (MPM). The MPM is a model that includes only influential predictors, predictors with posterior inclusion probability of 0.5 or above. See [BMA] **bmastats models**.

model averaging. In statistics, model averaging is an inferential technique that estimates a quantity of interest by a weighted average of individual model estimates over a space of candidate models. The weights are chosen according to various criteria with the aim of achieving certain asymptotic properties or improving prediction performance. Also see *Bayesian model averaging (BMA)*.

model distribution. In the Bayesian model averaging context, a model is viewed as a random quantity and thus has a distribution. This distribution is defined by model probabilities. There are prior and posterior model distributions. A prior model distribution is a (discrete) distribution defined over the model space and assumed for a model before observing the data. A posterior model distribution is a distribution of the model after observing the data given the assumed prior model distribution.

A prior model distribution represents an assumption about a model before observing the data. Common choices include a uniform prior on the model size and a uniform prior on the model space. The former assigns an equal probability to a model of any size. The latter assigns the same probability $1/|\mathcal{M}|$ to each model M, where $|\mathcal{M}|$ is the total number of models in the model space $\mathcal{M}$.

A posterior model distribution reflects the effect of the observed data on the prior assumption. The estimation of a posterior model distribution is at the heart of Bayesian model averaging.

model enumeration. A full enumeration of the model space. In the regression framework with p predictors, model enumeration considers all 2^p models formed by including or excluding each of the p predictors. Enumeration is feasible only with a moderate number of predictors. In bmaregress, model enumeration is available only when $p \leq 24$. If $p \leq 12$, bmaregress uses model enumeration automatically. For $12 < p \leq 24$, you can use the enumeration option to perform model enumeration. Also see *model sampling*.

model parameter sample. A Markov chain Monte Carlo sample from the posterior distribution of a model parameter. In a Bayesian model averaging regression, model parameters are the regression coefficients, the intercept, and the error variance. The posterior samples for these parameters are generated by the bmacoefsample command; see [BMA] **bmacoefsample**.

model posterior. See *posterior model distribution*.

model prior. See *prior model distribution*.

model probability. In the context of Bayesian model averaging, a model M is viewed as a discrete random variable defined on a model space $\mathcal{M}$ with a probability $0 \leq P(M) \leq 1$, such that $\sum_{M^* \in \mathcal{M}} P(M^*) = 1$. Model probabilities $P(M)$'s define a model distribution. Prior model probabilities $P(M)$'s define a prior model distribution. Posterior model probabilities $P(M|\mathbf{y})$'s, model probabilities given the observed outcome $\mathbf{y}$, define a posterior model distribution; also see *posterior model probability (PMP)*.

model sample. A Markov chain Monte Carlo sample from a posterior model distribution, estimated by the MC3 algorithm. Also see *posterior model probability (PMP)*.

model sampling. Simulation of a Markov chain Monte Carlo model sample. See [BMA] **bmaregress**.

model size. The number of predictors in the model, typically ignoring the constant term. For a model M, the model size is commonly denoted as $|M|$. See [BMA] **bmastats msize** and [BMA] **bmagraph msize**.

model space. A set of models considered for model averaging. In a regression setting, the model space includes 2^p distinct models, which correspond to all possible combinations of inclusions and exclusions of p predictors. In the presence of always predictors p_a, the model space contains 2^{p-p_a} models. In the presence of groups of predictors, an entire group is considered as one predictor in the definition of the model space. Depending on the context, the model space can sometimes imply the visited model space.

model uncertainty. Model uncertainty can be defined in many ways. In a regression setting, model uncertainty is uncertainty associated with the inclusion of predictors in a regression model. With p predictors, there are 2^p possible models. In this case, model uncertainty is defined with respect to these 2^p models.

model-size distribution. In the Bayesian model averaging context, a model is viewed as a random quantity. Its model size is random too and thus has a distribution. There are prior and posterior model-size distributions. The prior model-size distribution represents an assumption about the model size before observing the data. The posterior model-size distribution reflects the effect of the observed data on this prior assumption. It is useful to explore these distributions to evaluate the presumed and observed complexity of a Bayesian model averaging model.

The posterior model-size distribution can be computed by using the analytical or frequency posterior model probability, in which case it is referred to as, respectively, the analytical or frequency posterior model-size distribution. The prior model-size distribution is always computed analytically, but when the model space is not explored fully, it is conditional on the visited models. A beta-binomial distribution with shape parameters of one is commonly used as a noninformative prior for the model size. This is the default used by bmaregress. See [BMA] **bmagraph msize** and [BMA] **bmastats msize**.

modified MC3. A Markov chain Monte Carlo sampling algorithm used by bmaregress in the case of a random g. It uses the MC3 algorithm to sample from the model space and an adaptive Metropolis–Hastings algorithm to sample the g parameter. See *Methods and formulas* of [BMA] **bmaregress**.

MPM. See *median probability model (MPM)*.

noninclusion probability. The complementary probability of the inclusion probability.

PIP. See *posterior inclusion probability (PIP)*.

PMP. See *posterior model probability (PMP)*.

posterior coefficient sample. See *model parameter sample*.

posterior distribution. In a Bayesian model averaging regression, there is a posterior distribution of models and parameter g, which is estimated or simulated by bmaregress. There is also a posterior distribution of model parameters, which is simulated by bmacoefsample after bmaregress and defined as a mixture of the conditional posterior distributions given a model weighted by the posterior model probability.

posterior inclusion probability (PIP). PIP is the probability that a predictor is included in a model computed over the model space given the observed data and a prior model probability. PIP is used as a measure of the predictor's importance. Often, predictors with PIP of 0.5 or above are considered important predictors. Analytical and frequency PIPs are computed based on the respective analytical or frequency posterior model probabilities. See [BMA] **bmaregress** and [BMA] **bmastats pip**.

posterior mean model size. See *mean model size*.

posterior model distribution. See *model distribution*.

posterior model parameter sample. See *model parameter sample*.

posterior model probability (PMP). PMP is a model probability after observing the data with respect to the considered model space and given the assumed prior model probability. Consider a model M from a model space $\mathcal{M}$ and an observed outcome $\mathbf{y}$. Let $P(M)$ be a prior model probability, $f(\mathbf{y}|M)$ be the density of $\mathbf{y}$ given M, and $f(\mathbf{y})$ be the marginal density of $\mathbf{y}$. Then, using the Bayes formula,

PMP is defined as

$$P(M|\mathbf{y}) = \frac{f(\mathbf{y}|M)P(M)}{f(\mathbf{y})}$$

Analytical PMP is computed by using the analytical expressions, which are available only with a fixed g. Frequency PMP is computed from a Markov chain Monte Carlo model sample, which is available with a random g or with a fixed g when bmaregress's sampling option is specified or implied. See [BMA] **bmastats models** and [BMA] **bmagraph pmp**.

posterior model sample. See *model sample*.

posterior model-size distribution. See *model-size distribution*.

posterior noninclusion probability, **1 - PIP**. The complementary probability of the posterior inclusion probability. This is the probability mass at zero in a mixture posterior distribution of a regression coefficient. See [BMA] **bmagraph coefdensity**.

predictor, **predictor variable**. A variable used to predict an outcome or included in a model for the outcome. In the Stata context, this can be an existing variable in the dataset, or it can be a virtual variable, as described in [U] **11.4.3 Factor variables**, corresponding to a level of a factor variable or to an interaction term. In a regression context, predictor refers to any term in the specification of a regression function. A predictor with a high posterior inclusion probability (PIP), typically 0.5 or above, is considered an important predictor. A predictor with a lower PIP, typically less than 0.5, is considered a weak predictor. By default, bmaregress does not report predictors with PIP less than 0.01.

predictor-inclusion map. See *variable-inclusion map*.

prior mean model size. See *mean model size*.

prior model distribution. See *model distribution*.

prior model probability. See *model probability*.

prior model-size distribution. See *model-size distribution*.

random g, **random-g case**. A case when a random value is used for the g parameter of a Zellner's g-prior. This is the case when g is assumed to follow a distribution, hyperprior, and one of the following hyperpriors is specified in the gprior() option of bmaregress: betashrink, betabench, hyperg, hypergn, zsiow, or robust. Also see *fixed g*.

regression, Bayesian model averaging. See *Bayesian model averaging (BMA) regression*.

regression coefficient sample. See *model parameter sample*.

sampling correlation. A correlation between the analytical posterior model probability (PMP) and the frequency PMP. bmaregress reports it in the header when Markov chain Monte Carlo sampling is used. The sampling correlation is used to assess Markov chain Monte Carlo convergence. The closer it is to unity, the better. With a random g, when the analytical PMP is not available, the sampling correlation is computed as a correlation between the harmonic-mean estimator of the analytical PMP and the frequency PMP. See *Convergence of BMA* in *Remarks and examples* of [BMA] **bmaregress**.

shrinkage, **shrinkage factor**, **shrinkage parameter**. Shrinkage is defined as $g/(g+1)$, where g is the parameter of a Zellner's g-prior. The smaller this value, the more the coefficients are shrunk toward zero.

strong predictor. See *predictor*.

substitutes. See *disjointness*.

variable-inclusion summary, **predictor-inclusion summary**. A summary of the inclusion or exclusion of a predictor in a model. `bmastats models` provides this information in a tabular format with a column for each model and with a row for each predictor. Predictors that are included in the model are marked with an x in the corresponding row. For brevity, predictors with posterior inclusion probability less than 0.01 are not displayed. Also see *variable-inclusion map* for the graphical representation. See [BMA] **bmastats models**.

variable-inclusion map. A bar graph that plots models and their cumulative posterior model probability on the x axis and the predictors on the y axis. The bars of this graph represent coefficients. The signs of the coefficients are distinguished by color. For instance, `bmagraph varmap` uses blue for positive coefficients, red for negative coefficients, and gray for zero coefficients of predictors that were not included in a model. Also see *variable-inclusion summary* for the tabular representation. See [BMA] **bmagraph varmap**.

visited model space, **visited models**. A subset of models from a model space considered or "visited" by a Markov chain Monte Carlo sampling algorithm. With model enumeration, the visited model space is the same as the full model space.

weak predictor. See *predictor*.

Zellner's g-prior. A prior assumed for regression coefficients in a Bayesian model averaging regression. The g parameter controls the shrinkage of the coefficients toward zero. It can be fixed or random. Large g-values mean less shrinkage. In general, the larger the values of g, the more similar the Bayesian model averaging results are to the standard regression results. See [BMA] **bmaregress**.

References

Clarke, B. 2003. Comparing Bayes model averaging and stacking when model approximation error cannot be ignored. *Journal of Machine Learning Research* 4: 683–712.

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.

Subject and author index

See the combined subject index and the combined author index in the *Stata Index*.