

A REVIEW OF THE ‘BMS’ PACKAGE FOR R

SHAHRAM M. AMINI AND CHRISTOPHER F. PARMETER

ABSTRACT. This paper describes the relative merits and attractiveness of the newest Bayesian model averaging package, **BMS**, available in the statistical software R to implement a Bayesian model averaging exercise. This package provides the user with a wide range of customizable priors for conducting a BMA analysis, provides ample graphs to visualize results and offers several alternative model search mechanisms.

1. INTRODUCTION

The **BMS** package, created by Feldkircher & Zeugner (2009), implements linear Bayesian model averaging with a comprehensive choice of priors, Monte Carlo Markov Chain samplers, and post-processing functions in the statistical environment R (R Development Core Team 2009). This package provides a large degree of flexibility in the specification of the underlying features used to analyze the model and produces a wide range of easily-readable outputs including the posterior inclusion probabilities, means, and standard deviations, the number of models visited, and the number of draws from the sample. This package also takes advantage of R’s powerful graphical interface and creates useful graphs that allow the user to visualize the posterior model size distribution, posterior model probabilities and cumulative model probabilities as well as providing plotting facilities for the marginal densities of the posterior coefficients.

The following sections provide an overview of how to install the **BMS** package and implement a BMA empirical exercise using this package. We describe the main features under the user’s control for this package including the set of prior probabilities and model sampling algorithms as well as the plot diagnostics available to visualize the results. We show how a user can obtain estimates, posterior inclusion probabilities, and model statistics from this package with a mock empirical example. We do this both with a set of covariates that allows for full enumeration of the model space as well as requiring the implementation of a model space search mechanism.

2. INSTALLATION AND INSTALL SETUP

Prior to invocation of the **BMS** package, R must be installed. R can be downloaded at no charge from <http://www.cran.r-project.org>. The R statistical environment runs on all common operating

Date: June 24, 2011.

Key words and phrases. Model Averaging, Zellner’s g Prior, **BMS**.

Shahram M. Amini, Department of Economics, Virginia Polytechnic Institute and State University, email: shahram@vt.edu. Christopher F. Parmeter, Corresponding Author, Department of Economics, University of Miami, 305-284-4397, e-mail: cparmeter@bus.miami.edu.

systems including Linux, Unix, Windows, and Mac-OS X.¹ Once R is installed it is preferable to install BMS directly within R simply by typing

```
> install.packages("BMS")
```

in your R console. Then you will be prompted to choose a CRAN repository and the package should be installed after a few seconds. The package can then be loaded into the R workspace by typing `library(BMS)` at the command prompt.² For users who prefer to work with MATLAB, the core functionality of the BMS package can be accessed via the *BMS Toolbox for Matlab*. This toolbox provides MATLAB core functions that perform Bayesian model averaging by calling a hidden instance of R. This functionality requires a Windows operating system (NT, XP, Vista, or 7) and the installation of both MATLAB (Version 6 or higher) and the R statistical environment.³

3. FUNCTIONALITY

The main command within the BMS package is `bms()` which deploys a standard Bayesian normal-conjugate linear model. The various options available to the user to control aspects of the Bayesian model averaging exercise via `bms()` are summarized in Table 1. Primarily, the options for the `bms()` command hinge on sampling from the model space and selection of both model priors and Zellner's g priors. These three categories comprise the most important aspects of conducting a BMA exercise.

3.1. Controlling the BMA Exercise.

3.1.1. *Model Sampling.* A BMA analysis begins with constructing model weights for each model within the defined model space. However, since enumerating all potential variable combinations quickly becomes infeasible for even a modest number of covariates, the BMS package uses a Markov Chain Monte Carlo (MCMC) samplers to gather results on the most important part of the posterior distribution when more than 14 covariates exist. The BMS package offers two different MCMC samplers to probe the model space. These two methods differ in the way they propose candidate models. The first method is called the *birth-death* sampler (`mcmc=bd`). In this case, one of the potential regressors is randomly chosen; if the chosen variable is already in the current model M_i , then the candidate model M_j will have the same set of covariates as M_i but drops the chosen variable. If the chosen covariate is not contained in M_i , then the candidate model will contain all the variables from M_i plus the chosen covariate; hence the appearance (birth) or disappearance (death) of the chosen variable depends if it already appears in the model. The second approach is called the *reversible-jump* sampler (`mcmc=rev.jump`). This sampler draws a candidate model by the birth-death method with 50% probability and with 50% probability the candidate model

¹See Cribari-Neto & Zarkos (1999) and Racine & Hyndman (2002) for an overview of R.

²Ample applications, tutorials, empirical examples, and additional documentation can be found at <http://bms.zeugner.eu>.

³For more details see <http://bms.zeugner.eu/matlab/>.

TABLE 1. A summary of available `bms()` options

Option	Description	Default
<code>mcmc</code>	Indicates the model sampler to be used. If <code>bd</code> , a birth/death MCMC algorithm will be used. If <code>rev.jump</code> , a reversible jump algorithm is used that adds a swap step to the birth/death algorithm. If <code>enumerate</code> , the entire model space will be enumerated.	<code>bd</code>
<code>burn</code>	The number of burn-in draws for the MCMC sampler. Not taken into account if <code>mcmc=enumerate</code>	1000
<code>iter</code>	Shows the number of iteration draws to be sampled excluding burn-ins.	3000
<code>nmodel</code>	The number of best models for which information is stored.	500
<code>g</code>	Indicates the hyperparameters on Zellner's g -prior for the regression coefficients. If <code>UIP</code> , the number of observations, $g = N$, is used as g -prior. If <code>BRIC</code> , $g = \max(N, K^2)$, where K is the number of covariates, will be used. If <code>RIC</code> , $g = K^2$ is considered. If <code>HQ</code> , the package sets $g = \log(N)^3$. If <code>EBL</code> , a local empirical Bayes g -parameter will be used. If <code>hyper</code> , the package takes the hyper- g prior distribution.	N
<code>mprior</code>	Indicates the choice of model prior. If <code>fixed</code> , then fixed common prior inclusion probabilities for each regressor is employed. The <code>random</code> option employs the random theta prior. If <code>uniform</code> , the uniform model prior will be utilized. The <code>customk</code> option, allows for custom model size priors. Alternatively, the <code>pip</code> option allows for custom prior inclusion probabilities.	<code>random</code>
<code>start.value</code>	Specifies the starting model of the iteration algorithm.	NA
<code>g.stats</code>	If <code>TRUE</code> , statistics on the shrinkage factor $g/(1+g)$ will be collected. Set to <code>FALSE</code> for faster iteration.	<code>TRUE</code>
<code>force.full.ols</code>	If <code>TRUE</code> , the OLS estimation part of the sampling algorithm relies on a slower matrix inversion procedure. Setting this option to <code>TRUE</code> , slows down sampling but more capably handles highly collinear data.	<code>FALSE</code>
<code>fixed.reg</code>	Includes the explanatory variables that the user wants to always have included in every sampled model.	<code>numeric(0)</code>
<code>mprior.size</code>	If <code>mprior=fixed</code> or <code>mprior=random</code> , then <code>mprior.size</code> is a scalar that denotes the prior expected value of the model size prior. If <code>mprior=customk</code> , then a custom model size prior can be provided as a $K+1$ vector detailing the priors from model size 0 to K . If <code>mprior=pip</code> , then custom prior inclusion probabilities can be provided as a vector of size K , with elements in the interval $(0, 1)$.	$K/2$
<code>user.int</code>	If <code>TRUE</code> , the package prints out results to R console after ending the routine.	<code>TRUE</code>
<code>logfile</code>	If <code>TRUE</code> , a logfile named "test.log" will be produced in the current working directory, in order to keep track of the sampling procedure.	<code>FALSE</code>
<code>logstep</code>	Specifies at which number of posterior draws information is written to the log file.	10000

Notes: This table presents the available options, their default settings, and a brief description of each option for the Bayesian model averaging package, `BMS`, within the statistical software environment R. More details are provided in sections 3.1.1 through 3.1.3.

randomly drops one covariate with respect to M_i and randomly adds one random variable from the potential covariates that were not included in model M_i .

The precision of any MCMC sampling mechanism depends on the number of draws the procedure runs through. Given that the MCMC algorithms used in the `BMS` package may begin using models which might not necessarily be classified as 'good' models, the first set of iterations do not usually

draw models with high posterior model probabilities (PMP). This indicates that the sampler will only converge to spheres of models with the largest marginal likelihoods after some initial set of draws (known as the burn-in) from the candidate space. Therefore, this first set of iterations will be omitted from the computation of future results. In the `BMS` package the argument (`burn`) specifies the number of burn-ins (models omitted), and the argument (`iter`) the number of subsequent iterations to be retained. The default number of burn-in draws for either MCMC sampler is 1000 and the default number of iteration draws to be sampled (excluding burn-ins) is 3000 draws.

3.1.2. *Model Priors.* The `BMS` package offers considerable freedom in the choice of model prior. One can employ the uniform model prior as the choice of prior model size (`mprior="uniform"`), the Binomial model priors⁴ where the prior probability of a model is the product of inclusion and exclusion probabilities (`mprior="fixed"`), the Beta-Binomial model prior (`mprior="random"`) that puts a hyperprior on the inclusion probabilities drawn from a Beta distribution⁵ or a custom model size and prior inclusion probabilities (`mprior="customk"`).

3.1.3. *Alternative Zellner's g Priors.* Different mechanisms have been proposed in the literature for specifying g priors. The options in the `BMS` package are as follows

- (1) `g="UIP"`; Unit Information Prior, that corresponds to $g = N$, the sample size.⁶
- (2) `g="RIC"`; Sets $g = K^2$ and conforms to the risk inflation criterion.⁷
- (3) `g="BRIC"`; A mechanism that asymptotically converges to the unit information prior ($g = N$) or the risk inflation criterion ($g = K^2$). That is, the g prior is set to $g = \max(N, K^2)$.⁸
- (4) `g="HQ"`; Follows the Hannan-Quinn criterion asymptotically and sets $g = \log(N^3)$.
- (5) `g="EBL"`; Estimates a local empirical Bayes g -parameter.⁹
- (6) `g="hyper"`; Takes the "hyper- g " prior distribution.¹⁰

3.2. **Outputs.** In addition to posterior inclusion probabilities (PIP) and posterior means and standard deviations, `bms()` returns a list of aggregate statistics including the number of draws, burn-ins, models visited, the top models, and the size of the model space. It also returns the correlation between iteration counts and analytical PMPs for the best models, those with the highest PMPs.

3.3. **Plot Diagnostics.** `BMS` package users have access to plots of the prior and posterior model size distributions, a plot of posterior model probabilities based on the corresponding marginal likelihoods and MCMC frequencies for the best models that visualize how well the sampler has converged. A grid with signs and inclusion of coefficients vs. posterior model probabilities for the best models and plots of predictive densities for conditional forecasts are also produced.

⁴See Sala-i-Martin, Doppelhofer & Miller (2004)

⁵See Ley & Steel (2009)

⁶See Fernandez, Ley & Steel (2001)

⁷See Foster & George (1994) for more details

⁸See Fernandez et al. (2001)

⁹See George & Foster (2000) and Liang, Paulo, Molina, Clyde & Berger (2008)

¹⁰See Liang et al. (2008) and Feldkircher & Zeugner (2009)

3.4. Additional Functions. Beyond the baseline `bms()` command, the BMS package has many additional controls available to the user. Table 2 lists these alternative options. The first three functions create nicely formatted matrices containing posterior probabilities, means, standard errors, and R^2 s for each of the best models in a BMA object. If the BMS package is used for prediction purposes, then plotting commands listed in this table are useful tools that plot predictive densities and provide a visualization of PIPs. The package also provides an accessible function that extracts quantiles from the produced densities.

TABLE 2. A summary of available functions within the BMS package

Function	Description
<code>beta.draws.bma</code>	Returns a matrix whose columns are the expected value and standard deviations of coefficients for the best models in a BMA object.
<code>post.var</code>	Returns posterior residual variance, deviance, or pseudo R-squared, according to the choice of prior distributions.
<code>pmp.bma</code>	Returns the posterior model probabilities for the best models computed by the <code>bms()</code> command.
<code>plot.pred.density</code>	Plots predictive densities for conditional forecasts.
<code>plotComp</code>	Plots a comparison of PIPs, coefficients or their standard errors between various BMA objects.
<code>gdensity</code>	Calculates the mixture marginal posterior density for the shrinkage factor $g/(1+g)$ from a BMA object under the hyper- g prior and plots the density.
<code>quantile.density</code>	Extracts quantiles from density objects.
<code>lps.bma</code>	Computes the Log Predictive Score, an indicator for the likelihood of several forecasts, to evaluate a forecast based on a BMA object.
<code>c.bma</code>	Combines BMA objects and can be used to split estimation over several machines, or combine the MCMC results obtained from different starting points.

Notes: This table lists some of the important functions within the BMS package and provides a brief description of each function.

4. EMPIRICAL ILLUSTRATION

This section shows the results of a basic BMA exercise using the `Hedonic` dataset available in the `plm` package (Croissant & Millo 2008). This dataset contains information aggregated across census tracts for the metropolitan area that comprised Boston in 1970, originally published in Harrison & Rubinfeld (1978). This cross-section dataset consists of 506 observations along with 13 potential determinants of median housing prices in Boston. The dependent variable (MV) is the corrected median value of owner-occupied homes in \$1000's, and among the explanatory variables we have are the per capita crime rate ($CRIM$), the pupil-teacher ratio by town school district ($PTRATIO$), the index of accessibility to radial highways (RAD), the proportion of owner-occupied units built prior to 1940 (AGE), the proportion of non-retail business acres per town ($INDUS$), the proportion of African-Americans per town (B), and a variable reflecting the nitric oxides concentration (NOX). See Harrison & Rubinfeld (1978, Table IV) for more details.

The authors use a hedonic housing price model to generate quantitative estimates of the willingness to pay for air quality improvements. We assume there exists uncertainty about which housing attributes should be a part of the final model. The small number of covariates allows the `bms()` command to fully enumerate the model space. To show how to call options for the model space search we append 12 fictitious variables (generated as independent standard normal random deviates) which pushes the number of covariates beyond the threshold search level. We present diagnostic plots to illustrate the flexibility of the calls within the package.

4.1. Enumeration of the Model Space. We first present the results of performing basic Bayesian model averaging with the original `Hedonic` dataset, which is small enough to fully enumerate the model space. First, we load the required packages and set the R options via the following commands:

```
> set.seed(2011)
> library(plm)
> library(BMS)
> library(xtable)
```

Next we load the dataset, append the 12 fictitious variables, and manipulate it using the following commands:

```
> data(Hedonic, package = "plm")
> Hedonic$chas <- as.numeric(Hedonic$chas)
> Hedonic <- subset(Hedonic, select = -townid)
> noise <- matrix(rnorm(12 * nrow(Hedonic)), ncol = 12)
> colnames(noise) <- paste("noise", 1:12, sep = "")
> Hedonic.large <- cbind(Hedonic, noise)
```

Now, we perform Bayesian model averaging via the `bms()` command, and retrieve the main results via the `coef` function. The following commands will implement the Bayesian model averaging analysis over the `Hedonic` dataset.

```
> bms_enu <- bms(Hedonic, mcmc = "enumerate", g = "UIP", mprior = "uniform",
+   user.int = FALSE)
> model_enu <- coef(bms_enu)
```

In this setup, `bms()` enumerates all $2^{13} = 8,192$ models. The current call uses a uniform distribution for model priors and unit information priors (UIP) for the distribution of regressions coefficients.

Table 3 shows the output for `model_enu`. The first column, named PIP, shows the posterior inclusion probabilities indicating the probability that each variable belongs to the final model. `BMS` sorts these probabilities from the highest to the lowest automatically. The second and third columns show the posterior means and standard errors of regressors calculated from this package.

Other additional information about the model averaging procedure can be obtained using the `summary` function. In our case the output would be

TABLE 3. model.enu Estimation Output

	PIP	Post Mean	Post SD
crim	1.000	-0.012	0.001
lstat	1.000	-0.371	0.023
dis	1.000	-0.194	0.027
ptratio	1.000	-0.031	0.005
nox	1.000	-0.006	0.001
rm	1.000	0.006	0.001
rad	0.999	0.095	0.019
tax	0.988	-0.000	0.000
blacks	0.964	0.355	0.122
chas	0.695	0.064	0.050
indus	0.044	0.000	0.001
age	0.044	0.000	0.000
zn	0.043	0.000	0.000

```
> summary(bms_enu)
```

```

Mean no. regressors          Draws          Burnins          Time
      "9.7776"                "8192"                "0"          "1.343281 secs"
No. models visited          Modelspace 2^K          % visited          % Topmodels
      "8192"                "8192"                "100"          "6.1"
Corr PMP                    No. Obs.          Model Prior          g-Prior
      "NA"                  "506"          "uniform / 6.5"          "UIP"
Shrinkage-Stats
      "Av=0.998"

```

As is apparent, some useful statistics including the number of models visited, the average number of regressors used, and time performance are produced.

4.2. Model Sampling. In this section we present results obtained when full enumeration is computationally infeasible or undesirable. Using the `Hedonic.large` dataset, the following commands will implement the Bayesian model averaging analysis using the birth-death MCMC search algorithm (see section 3.1.1) with 10000 iterations and a burn in of 2000 models.

```

> bms_sam <- bms(Hedonic.large, burn = 2000, iter = 10000, mcmc = "bd",
+   g = "UIP", mprior = "uniform", nmodel = 2000, user.int = FALSE)
> model.sam <- coef(bms_sam)[1:10, ]

```

The PIPs for this exercise are shown in the first column of Table 4 after invocation of the `coef` function on the `bms` object. The choices for model priors and the distribution of the model coefficients are uniform distribution and unit information priors, respectively. The next two columns show the estimated posterior means and standard deviations of the covariates. Even with the addition of the 12 fictitious variables the birth-death sampler provides PIPs consistent with the earlier full enumeration example. Additional statistical information is revealed via the call

TABLE 4. model.sam Estimation Output

	PIP	Post Mean	Post SD
crim	1.000	-0.012	0.001
nox	1.000	-0.006	0.001
rm	1.000	0.006	0.001
dis	1.000	-0.194	0.027
rad	1.000	0.096	0.018
ptratio	1.000	-0.031	0.005
lstat	1.000	-0.371	0.023
tax	0.997	-0.000	0.000
blacks	0.979	0.360	0.115
chas	0.667	0.061	0.051

```

> summary(bms_sam)
Mean no. regressors           Draws           Burnins           Time
      "10.5527"           "10000"           "2000"           "1.52063 secs"
No. models visited           Modelspace 2^K           % visited           % Topmodels
      "1205"           "3.4e+07"           "0.0036"           "100"
Corr PMP           No. Obs.           Model Prior           g-Prior
      "0.9943"           "506"           "uniform / 12.5"           "UIP"
Shrinkage-Stats
      "Av=0.998"

```

Note that the MCMC search algorithm visited less than one half of one percent of the total number of models in the model space and still provided results consistent with full enumeration over the model space.

4.3. Plot Diagnostics. The plotting facilities within *BMS* are an important tool that allows the user to visualize the shape of the posterior distributions of coefficients, assess the final model size, compare PIPs and investigate model complexity. The *BMS* package plots the marginal inclusion densities as well as images that show inclusion and exclusion of variables within models using separate colors. This package also provides graphics for both the PMPs and posterior model size distribution. The following figures are some examples of the available plots corresponding to the above example where we added 12 fictitious covariates to force the package to engage in searching over the model space. In order to produce a plot for comparison of prior and posterior distributions of the model size simply use

```
> plot(bms_sam[1:200], lty = c(1, 2))
```

The first argument in the command is the R object where we have saved our estimation results and the second argument defines the type of line we are going to use for our graphs.

Figure 1 is a combined plot from the *BMS* package. The upper plot shows the prior and posterior distribution of model sizes and helps the user illustrate the impact of the model prior assumption

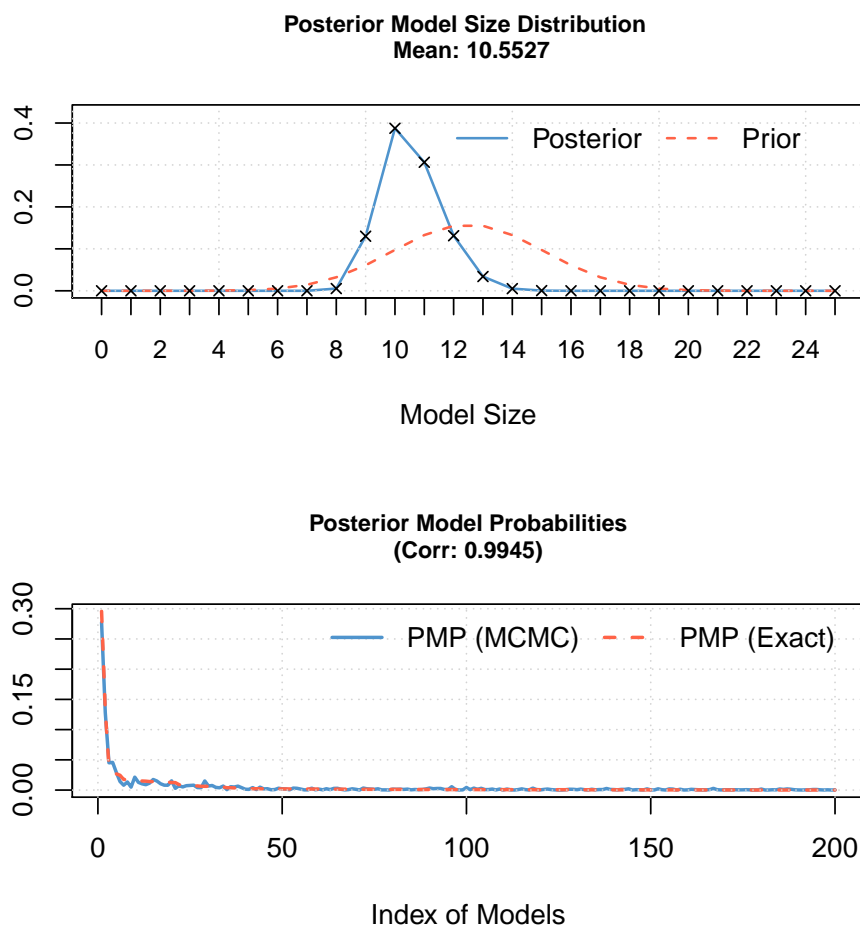


FIGURE 1. Posterior model size distribution and model probabilities produced from the BMS package with “uniform” model priors.

on the estimation results. For example, the upper plot of Figure 1 assumes a uniform distribution on model priors and if the user wants to examine how far the posterior model size distribution matches up to the prior, he might consider other popular model priors that allow more freedom in choosing prior expected model size and other factors. The lower plot of the figure is an indicator of how well the first 200 best models encountered by BMS have converged. These plots are useful in determining the search mechanisms ability to find ‘good’ models throughout the model space.

Figure 2 is a visualization of the mixture marginal posterior density for the coefficient for the Charles River Dummy produced by the BMS package. The bar above the density shows the PIP, and the dotted vertical lines show the corresponding standard deviation bounds. This figure can be generated simply via `density.bma` function within the BMS package

```
> density.bma(bms_sam, reg = "chas", addons = "esEpl")
```

This is an extremely flexible command and provides several options for the user. `addons` receives string values specifying which additional information should be added to the plot via low-level commands. To see more details run `help(density.bma)` in your R Console.

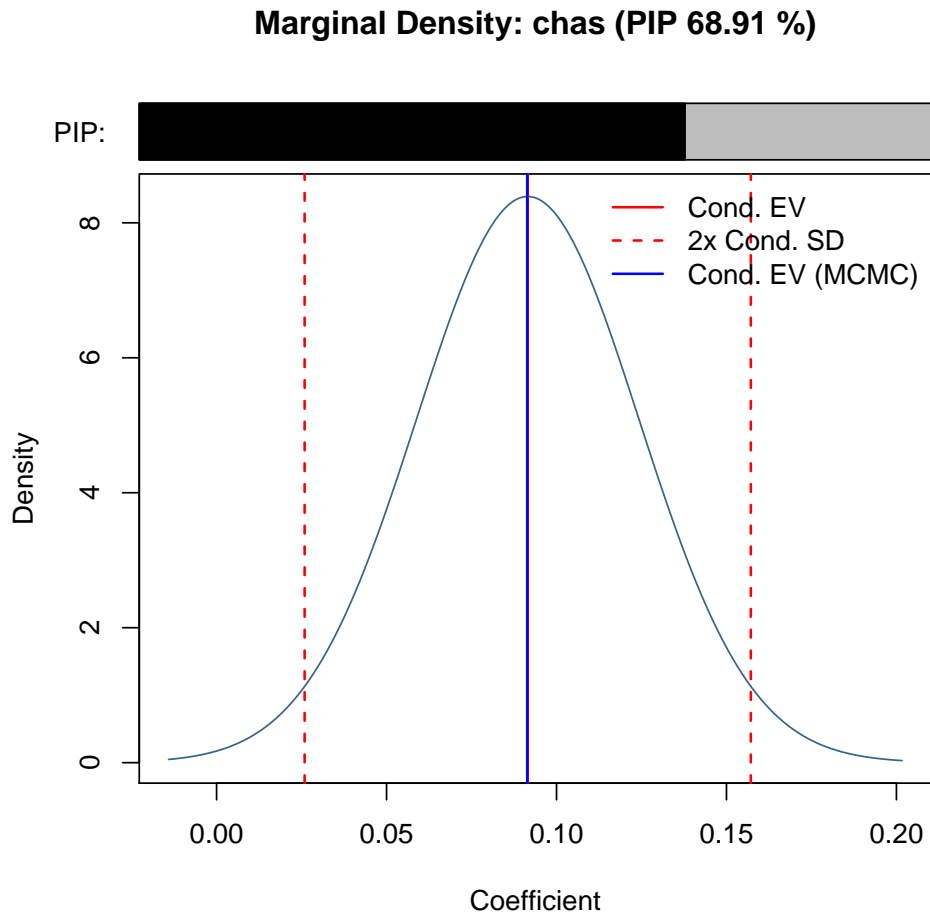


FIGURE 2. Marginal density of the coefficient for the Charles River dummy (`chas`) from the BMS package.

5. CONCLUSIONS

This paper has outlined the most recently available Bayesian model averaging package (BMS) in the statistical computing environment R. Our goal was to familiarize users with the different options that this package has to offer. We highlighted how this package implements a BMA analysis as well as the options available to the user and the outputs that are returned. To demonstrate the operation of this package in practice, we presented both a simple empirical example that allowed full enumeration of the model space as well as requiring engagement of MCMC methods to search across the model space. In both instances the insights gleaned were virtually identical.

REFERENCES

- Cribari-Neto, F. & Zarkos, S. (1999), ‘R: Yet another econometric programming environment’, *Journal of Applied Econometrics* **14**(3), 319–329.
- Croissant, Y. & Millo, G. (2008), ‘Panel data econometrics in R: The plm package’, *Journal of Statistical Software* **27**(2).
URL: <http://www.jstatsoft.org/v27/i02/>
- Feldkircher, M. & Zeugner, S. (2009), Benchmark priors revisited: On adaptive shrinkage and the supermodel effect in Bayesian model averaging, IMF Working Papers 09/202, International Monetary Fund.
URL: <http://ideas.repec.org/p/imf/imfwpa/09-202.html>
- Fernandez, C., Ley, E. & Steel, M. (2001), ‘Benchmark priors for Bayesian model averaging’, *Journal of Econometrics* **100**(2), 381–427.
- Foster, D. & George, E. (1994), ‘The risk inflation criterion for multiple regression’, *The Annals of Statistics* **22**(4), 1947–1975.
- George, E. & Foster, D. (2000), ‘Calibration and empirical Bayes variable selection’, *Biometrika* **87**(4), 731–747.
- Harrison, D. & Rubinfeld, D. (1978), ‘Hedonic housing prices and the demand for clean air’, *Journal of Environmental Economics and Management* **5**(1), 81–102.
- Ley, E. & Steel, M. (2009), ‘On the effect of prior assumptions in Bayesian model averaging with applications to growth regression’, *Journal of Applied Econometrics* **24**(4), 651–674.
- Liang, F., Paulo, R., Molina, G., Clyde, M. & Berger, J. (2008), ‘Mixtures of g priors for Bayesian variable selection’, *Journal of the American Statistical Association* **103**(481), 410–423.
- R Development Core Team (2009), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.
URL: <http://www.R-project.org>
- Racine, J. & Hyndman, R. (2002), ‘Using R to teach econometrics’, *Journal of Applied Econometrics* **17**(2), 175–189.
URL: <http://dx.doi.org/10.1002/jae.657>
- Sala-i-Martin, X., Doppelhofer, G. & Miller, R. (2004), ‘Determinants of long-term growth: A Bayesian averaging of classical estimates (BACE) approach’, *American Economic Review* **94**(4), 813–835.