Journal of Statistical Software

November 2015, Volume 68, Issue 4. doi: 10.18637/jss.v068.i04

Bayesian Model Averaging Employing Fixed and Flexible Priors: The BMS Package for R

Stefan Zeugner
European Center for
Advanced Research in Economics

Martin Feldkircher
Oesterreichische Nationalbank

Abstract

This article describes the **BMS** (Bayesian model sampling) package for R that implements Bayesian model averaging for linear regression models. The package excels in allowing for a variety of prior structures, among them the "binomial-beta" prior on the model space and the so-called "hyper-g" specifications for Zellner's g prior. Furthermore, the **BMS** package allows the user to specify her own model priors and offers a possibility of subjective inference by setting "prior inclusion probabilities" according to the researcher's beliefs. Furthermore, graphical analysis of results is provided by numerous built-in plot functions of posterior densities, predictive densities and graphical illustrations to compare results under different prior settings. Finally, the package provides full enumeration of the model space for small scale problems as well as two efficient MCMC (Markov chain Monte Carlo) samplers that sort through the model space when the number of potential covariates is large.

Keywords: hyper-g prior, binomial-beta prior, empirical Bayes, customized prior inclusion probabilities, **BMS**, R.

1. A brief introduction to Bayesian model averaging

Model uncertainty is a problem that arises frequently in applied econometrics: Which set of the covariates is appropriate to explain variation of the response variable? Are my results robust to in-/exclusion of additional explanatory variables? In addressing these issues Bayesian model averaging (BMA) has become a popular alternative to model selection. The remainder of this article is structured as follows: This section re-iterates some basic concepts, and introduces notation for readers with limited knowledge of BMA. In Section 2 estimation using package **BMS** (Bayesian model sampling) is explained using employing a default prior setting. Section 3 introduces different priors on the model space, while Section 4 introduces

the MCMC sampler implemented in package **BMS**. Section 5 provides an overview about the g prior settings, and Section 6 describes prediction in BMA. Section 7 concludes.

1.1. Bayesian model averaging

Bayesian model averaging[1] addresses model uncertainty in a canonical regression problem. Suppose a linear model structure, with y being the dependent variable, α_γ a constant, β_γ the coefficients, and ε a normal IID error term with variance σ^2:

$$y = \alpha_\gamma + X_\gamma \beta_\gamma + \varepsilon, \qquad \varepsilon \sim N(0, \sigma^2 I). \tag{1}$$

A problem arises when there are many potential explanatory variables in a matrix X: Which variables $X_\gamma \in \{X\}$ should be then included in the model? And how important are they? The direct approach to do inference on a single linear model that includes all variables is inefficient or even infeasible with a limited number of observations.

BMA tackles the problem by estimating models for all possible combinations of $\{X\}$ and constructing a weighted average over all of them. If X contains K potential variables, this means estimating 2^K variable combinations and thus 2^K models. The model weights for this averaging stem from posterior model probabilities that arise from Bayes' theorem:

$$p(M_\gamma|y, X) \;=\; \frac{p(y|M_\gamma, X)p(M_\gamma)}{p(y|X)} \;=\; \frac{p(y|M_\gamma, X)p(M_\gamma)}{\sum_{s=1}^{2^K} p(y|M_s, X)p(M_s)}. \tag{2}$$

Here, $p(y|X)$ denotes the *integrated* likelihood which is constant over all models and is thus simply a multiplicative term.[2] Therefore, the posterior model probability (PMP) is proportional (embodied by the sign $\propto$) to the *integrated likelihood* $p(y|M_\gamma, X)$, which reflects the probability of the data given model M_γ. The marginal likelihood of model M_γ is multiplied by its prior model probability $p(M_\gamma)$ indiating how probable the researcher thinks model M_γ is before looking at the data. The difference between $p(y|X)$ and $p(y|M_\gamma, X)$ is that integration is once over the model space ($p(y|X)$) and once for a given model over the parameter space $p(y|M_\gamma, X)$. By re-normalization of the product from above one can infer the PMPs and thus the model weighted posterior distribution for any statistic θ (e.g., the estimator of the coefficient β_γ):

$$p(\theta|y, X) = \sum_{\gamma=1}^{2^K} p(\theta|M_\gamma, y, X) \frac{p(M_\gamma|X, y)p(M_\gamma)}{\sum_{s=1}^{2^K} p(M_s|y, X)p(M_s)}.$$

The model prior $p(M_\gamma)$ has to be elicited by the researcher and should reflect prior beliefs. A popular choice is to set a uniform prior probability for each model $p(M_\gamma) \propto 1$ to represent the lack of prior knowledge. Further model prior options will be explored in Section 3.

1.2. Bayesian linear models and Zellner's g prior

The specific expressions for the marginal likelihoods $p(M_\gamma|y, X)$ and the posterior distributions $p(\theta|M_\gamma, y, X)$ depend on the chosen estimation framework. The literature standard is

[1] For an excellent introduction see Hoeting, Madigan, Raftery, and Volinsky (1999).

[2] According to the literature in what follows we will use the terms *integrated* and *marginal* likelihood interchangeably.

to use a "Bayesian regression" linear model with a specific prior structure called "Zellner's g prior" as will be outlined in this section.[3]

For each individual model M_γ suppose a normal error structure as in (1). The need to obtain posterior distributions requires to specify the priors on the model parameters. Here, we place "improper" priors on the constant and error variance, which means they are evenly distributed over their domain: $p(\alpha_\gamma) \propto 1$, i.e., complete prior uncertainty where the constant is located. Similarly, set $p(\sigma) \propto \sigma^{-1}$.

The crucial prior is the one on the regression coefficients β_γ: Before looking at the data (y, X), the researcher formulates her prior beliefs on coefficients into a normal distribution with a specified mean and variance. It is common to assume a conservative prior mean of zero for the coefficients to reflect that not much is known about them. Their variance structure is defined according to Zellner's g: $g\sigma^2({X_\gamma}^\top X_\gamma)^{-1}$:

$$\beta_\gamma | g \sim N\left(0, g\sigma^2\left({X_\gamma}^\top X_\gamma\right)^{-1}\right).$$

This means that the researcher thinks coefficients are zero, and that their variance-covariance structure is broadly in line with that of the data X_γ. The hyperparameter g embodies how certain the researcher is that coefficients are indeed zero: A small g means small prior coefficient variance and therefore implies the researcher is quite certain (or conservative) that the coefficients are indeed zero. In contrast, a large g means that the researcher is very uncertain that coefficients are zero.

The posterior distribution of coefficients reflects prior uncertainty: Given g, it follows a t-distribution with expected value $\mathsf{E}(\beta_\gamma | y, X, g, M_\gamma) = \frac{g}{1+g}\hat{\beta}_\gamma$, where $\hat{\beta}_\gamma$ is the standard OLS estimator for model γ. The expected value of coefficients is thus a convex combination of OLS estimator and prior mean (zero). The more conservative (smaller) g, the more important is the prior, and the more the expected value of coefficients is shrunk toward the prior mean zero. As $g \to \infty$, the coefficient estimator approaches the OLS estimator. Similarly, the posterior variance of β_γ is affected by the choice of g:[4]

$$\mathsf{COV}(\beta_\gamma | y, X, g, M_\gamma) = \frac{(y-\bar{y})^\top (y-\bar{y})}{N-3}\frac{g}{1+g}\left(1 - \frac{g}{1+g}R^2_\gamma\right)({X_\gamma}^\top X_\gamma)^{-1}.$$

I.e., the posterior covariance is similar to that of the OLS estimator times a factor that includes g. The Appendix A.3 shows how to apply the function `zlm` in order to estimate such models outside of the BMA context.

For BMA, this prior framework results in a very simple marginal likelihood $p(y|M_\gamma, X, g)$, that is related to the R-squared and includes a size penalty factor adjusting for model size k_γ:

$$p(y|M_\gamma, X, g) \propto (y-\bar{y})^\top (y-\bar{y})^{-\frac{N-1}{2}}(1+g)^{-\frac{k_\gamma}{2}}\left(1 - \frac{g}{1+g}\right)^{-\frac{N-1}{2}}.$$

The crucial choice here concerns the form of the hyperparameter g. A popular "default" approach is the "unit information prior" (UIP), which sets $g = N$ commonly for all models

[3]Note that the presented framework is very similar to the natural normal-gamma-conjugate model – which employs proper priors for α and σ. Nonetheless, the resulting posterior statistics are virtually identical under uninformative priors on the model and parameter space.

[4]Here, N denotes sample size, and $\bar{y}$ the sample mean of the response variable.

and thus attributes about the same information to the prior as is contained in one observation. Please refer to Section 5 for a discussion of other g priors.[5]

2. A BMA example: Attitude data

This section shows how to run BMA using the R (R Core Team 2015) package **BMS** (Feldkircher and Zeugner 2015). Package **BMS** is available from the Comprehensive R Archive Network (CRAN) at `http://CRAN.R-project.org/package=BMS`. We will use a small data set for illustration and show how to obtain posterior coefficient and model statistics.

2.1. Model sampling

Equipped with this basic framework, let us explore one of the data sets built into R: The `attitude` dataset describes the overall satisfaction rating of a large organization's employees, as well as several specific factors such as `complaints`, the way of handling complaints within the organization (for more information type `help("attitude")`). The data includes 6 variables, which means $2^6 = 64$ model combinations. Let us stick with the UIP g prior (in this case $g = N = 30$). Moreover, assume uniform model priors (which means that our expected prior model parameter size is $K/2 = 3$).

First load the data set by typing

```
R> data("attitude", package = "datasets")
```

In order to perform BMA you have to load the **BMS** package first, via the command:

```
R> library("BMS")
```

Now perform Bayesian model sampling via the function `bms`, and write results into the variable `att`.

```
R> att <- bms(attitude, mprior = "uniform", g = "UIP", user.int = FALSE)
```

`mprior = "uniform"` means to assign a uniform model prior, `g = "UIP"`, the unit information prior on Zellner's g. The option `user.int = FALSE` is used to suppress user-interactive output for the moment.[6] The first argument is the data frame `attitude`, and `bms` assumes that its first column is the response variable.[7]

2.2. Coefficient results

The coefficient results can be obtained via

```
R> coef(att)
```

[5]Note that package **BMS** is, in principle not restricted to Zellner's g priors, as quite different coefficient priors might be defined by R-savvy users.

[6]Note that the argument `g = "UIP"` is actually redundant, as this is the default option for `bms`. The default model prior is somewhat different but does not matter very much with this data. Therefore, the command `att = bms(attitude)` gives broadly similar results.

[7]The specification of data can be supplied in different manners, e.g., using a '`formula`'. Type `help("lm")` for a comparable function.

```
                PIP    Post Mean    Post SD Cond.Pos.Sign Idx
complaints 0.9996351  0.684449094 0.13038429    1.00000000   1
learning   0.4056392  0.096481513 0.15135419    1.00000000   3
advance    0.2129325 -0.026686161 0.09133894    0.00000107   6
privileges 0.1737658 -0.011854183 0.06143387    0.00046267   2
raises     0.1665853  0.010567022 0.08355244    0.73338938   4
critical   0.1535886  0.001034563 0.05465097    0.89769774   5
```

The above matrix shows the variable names and corresponding statistics: The second column `Post Mean` displays the coefficients averaged over all models, including the models wherein the variable was not contained (implying that the coefficient is zero in this case). The covariate `complaints` has a comparatively large coefficient and seems to be most important. The importance of the variables in explaining the data is given in the first column `PIP` which represents posterior inclusion probabilities – i.e., the sum of PMPs for all models wherein a covariate was included. We see that with 99.96%, virtually all of posterior model mass rests on models that include `complaints`. In contrast, `learning` has an intermediate PIP of 40.6%, while the other covariates do not seem to matter much. Consequently their (unconditional) coefficients[8] are quite low, since the results quite often include models where these coefficients are zero.

The coefficients' posterior standard deviations (`Post SD`) provide further evidence: for example, `complaints` is certainly positive, while `advance` is most likely negative. In fact, the coefficient sign can also be inferred from the fourth column `Cond.Pos.Sign`, the "posterior probability of a positive coefficient expected value conditional on inclusion", respectively "sign certainty". Here, we see that in all encountered models containing these variables, the (expected values of) coefficients for `complaints` and `learning` were positive. In contrast, the corresponding number for `privileges` is near to zero, i.e., in virtually all models that include `privileges`, its coefficient sign is negative. Finally, the last column `idx` denotes the index of the variables' appearance in the original data set, as our results are obviously sorted by PIP.

In addition to inferring about the importance of our variables, it might be really more interesting to look at their standardized coefficients.[9] Type:

```
R> coef(att, std.coefs = TRUE, order.by.pip = FALSE,
+    include.constant = TRUE)
```

```
                 PIP     Post Mean    Post SD Cond.Pos.Sign Idx
complaints 0.9996351  0.7486734114 0.14261872    1.00000000   1
privileges 0.1737658 -0.0119154065 0.06175116    0.00046267   2
learning   0.4056392  0.0930292869 0.14593855    1.00000000   3
raises     0.1665853  0.0090258498 0.07136653    0.73338938   4
critical   0.1535886  0.0008409819 0.04442502    0.89769774   5
```

[8]Unconditional coefficients are defined as $\mathsf{E}(\beta|y,X) = \sum_{\gamma=1}^{2^K} p(\beta_\gamma|,y,X,M_\gamma)p(M_\gamma|y,X)$ i.e., a weighted average over all models, including those where this particular coefficient was restricted to zero. A conditional coefficient in contrast, is "conditional on inclusion", i.e., a weighted average only over those models where its regressor was included. Conditional coefficients may be obtained with the command `coef(att, condi.coef = TRUE)`.

[9]Standardized coefficients arise if both the response y and the regressors X are normalized to mean zero and variance one – thus effectively bringing the data down to the same order of magnitude.

```
advance     0.2129325 -0.0225561446 0.07720309    0.00000107   6
(Intercept) 1.0000000  1.2015488514         NA            NA   0
```

The standardized coefficients reveal similar importance as discussed above, but one sees that `learning` actually does not matter much in terms of magnitude. Note that using argument `order.by.pip = FALSE` leads to the covariates being represented in their original order. The argument `include.constant = TRUE` also prints out a (standardized) constant.

2.3. Other results

Other basic information about the sampling procedure can be obtained via.[10]

```
R> summary(att)
```

```
Mean no. regressors               Draws             Burnins
          "2.1121"                "64"                 "0"
              Time  No. models visited      Modelspace 2^K
 "0.02240419 secs"                "64"                "64"
         % visited         % Topmodels            Corr PMP
             "100"               "100"                "NA"
          No. Obs.         Model Prior             g-Prior
              "30"       "uniform / 3"               "UIP"
   Shrinkage-Stats
       "Av=0.9677"
```

It reiterates some of the facts we already know, but adds some additional information such as `Mean no. regressors`, posterior expected model size (cf., Section 3).

Finally we can examine the distribution of posterior model probabilities by invoking the function `topmodels`. This yields a binary matrix with the variables arranged row-wise and the models column-wise. For a particular model (i.e., column) a 0 indicates exclusion and 1 inclusion of a variable associated with a given row. For the sake of illustration we will focus on the three models with highest posterior model probabilities:

```
R> topmodels(att)[, 1:3]
```

```
                  20        28         29
complaints  1.0000000 1.0000000 1.00000000
privileges  0.0000000 0.0000000 0.00000000
learning    0.0000000 1.0000000 1.00000000
raises      0.0000000 0.0000000 0.00000000
critical    0.0000000 0.0000000 0.00000000
advance     0.0000000 0.0000000 1.00000000
PMP (Exact) 0.2947721 0.1661679 0.06678871
PMP (MCMC)  0.2947721 0.1661679 0.06678871
```

[10] Note that the command `print(att)` is equivalent to `coef(att); summary(att)`.

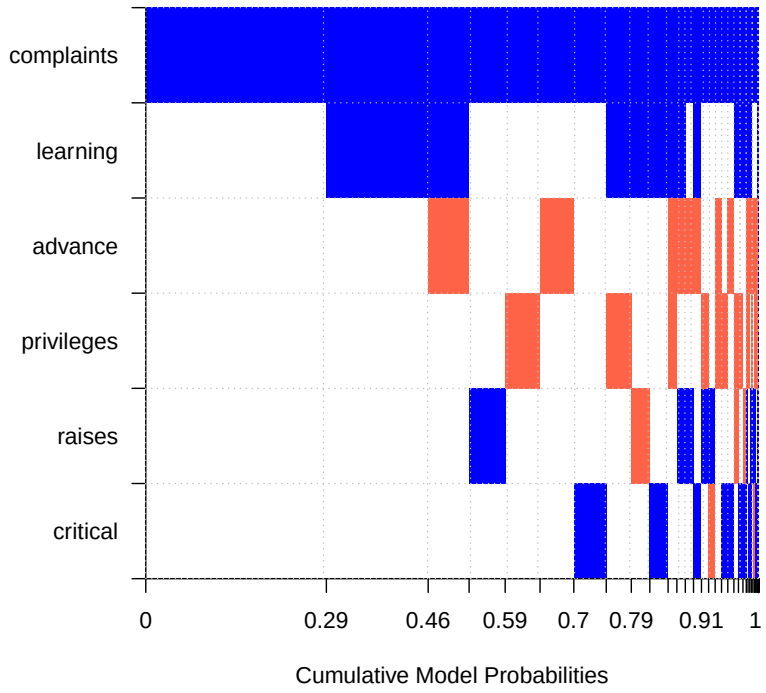

Figure 1: Image plot. Blue color corresponds to a positive coefficient, red to a negative coefficient, and white to non-inclusion of the respective variable. The horizontal axis is scaled by the models' posterior model probabilities.

The posterior model probability for each model is given at the bottom of the matrix. The distinction between `PMP (Exact)` and `PMP (MCMC)` is of importance if an MCMC sampler was used and will be discussed in Section 4.3. Note that we can access the PMP for any model directly using the function `pmpmodel` – cf., `help("pmpmodel")`. The best model, with 29% posterior model probability, is the one that only includes `complaints`. However the second best model includes `learning` in addition and has a PMP of 17%. Use the command `beta.draws(att)` to obtain the actual (expected values of) posterior coefficient estimates for each of these models.

In order to get a more comprehensive overview over the models, use the command

```
R> image(att)
```

that produces Figure 1.

Here, blue color corresponds to a positive coefficient, red to a negative coefficient, and white to non-inclusion (a zero coefficient). On the horizontal axis the best models are shown, scaled by their PMPs. We see again that the best model with most mass only includes `complaints`. Moreover we see that complaints is included in virtually all model mass, and unanimously with a positive coefficient. In contrast, `raises` is included very infrequently, and its coefficient sign changes according to the model. Use `image(att, yprop2pip = TRUE)` for another illustrating variant of this plot.

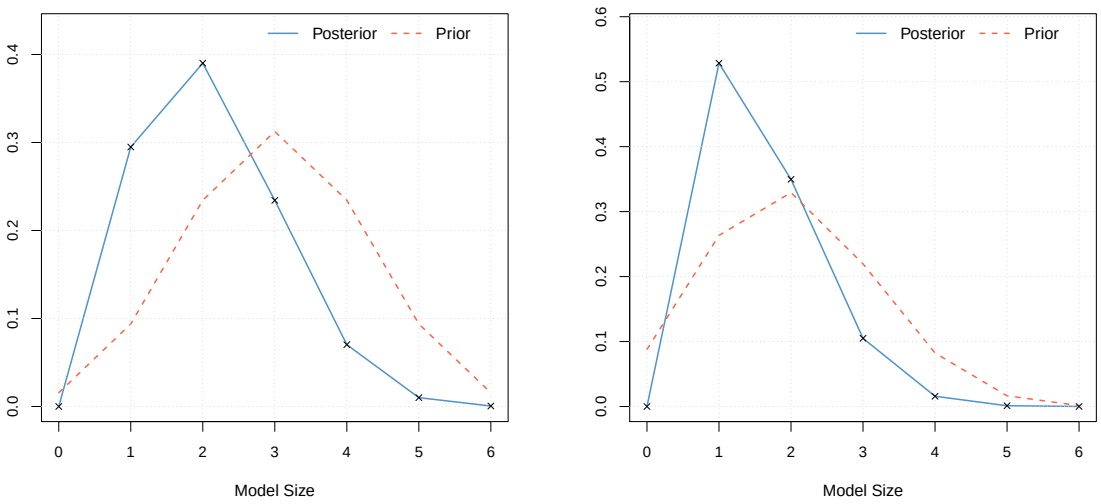

Figure 2: Distribution of posterior model size. Left panel: Uniform prior on the model space, prior expected model size equals $\bar{m} = 3$. Right panel: Informative prior on model space used, prior expected model size equals $\bar{m} = 2$.

3. Model size and model priors

Invoking the command `summary(att)` yielded the important posterior statistic `Mean no. regressors`, the posterior expected model size (i.e., the average number of included regressors), which in our case was 2.11. Note that the posterior expected model size is equal to the sum of PIPs – verify via:

```
R> sum(coef(att)[, 1])

[1] 2.112147
```

This value contrasts with the prior expected model size implicitly used in our model sampling: With 2^K possible variable combinations, a uniform model prior means a common prior model probability of $p(M_\gamma) = 2^{-K}$. However, this implies a prior expected model size of $\sum_{k=0}^{K} \binom{K}{k} k 2^{-K} = K/2$. Moreover, since there are more possible models of size 3 than e.g., of size 1 or 5, the uniform model prior puts more mass on intermediate model sizes – e.g., expecting a model size of $k_\gamma = 3$ with $\binom{6}{3} 2^{-K} = 31\%$ probability. In order to examine how far the posterior model size distribution matches up to this prior, type:

```
R> plotModelsize(att)
```

The results are illustrated in Figure 2, left panel.

We see that while the model prior implies a symmetric distribution around $K/2 = 3$, updating it with the data yields a posterior that puts more importance on parsimonious models. In order to illustrate the impact of the uniform model prior assumption, we might consider other popular model priors that allow more freedom in choosing prior expected model size and other factors.

3.1. Binomial model prior

The binomial model prior constitutes a simple and popular alternative to the uniform prior we just employed. It starts from the covariates' viewpoint, placing a common and fixed inclusion probability θ on each regressor. The prior probability of a model of size k_γ is therefore the product of inclusion and exclusion probabilities:

$$p(M_\gamma) = \theta^{k_\gamma}(1-\theta)^{K-k_\gamma}.$$

Since expected model size is $\bar{m} = K\theta$, the researcher's prior choice reduces to eliciting a prior expected model size $\bar{m}$ (which defines θ via the relation $\theta = \bar{m}/K$). Choosing a prior model size of $K/2$ yields $\theta = \frac{1}{2}$ and thus exactly the uniform model prior $p(M_\gamma) = 2^{-K}$ illustrated in the previous section. Therefore, putting prior model size at a value $< \frac{1}{2}$ tilts the prior distribution toward smaller model sizes and vice versa. For instance, let us impose a fixed inclusion probability prior such that prior model size equals $\bar{m} = 2$: Here, the option `user.int = TRUE` directly prints out the results as from `coef` and `summary`:[11]

```
R> att_fixed <- bms(attitude, mprior = "fixed", mprior.size = 2,
+    user.int = TRUE)

                  PIP     Post Mean    Post SD Cond.Pos.Sign Idx
complaints 0.99971415  0.7034253730 0.12131094    1.00000000   1
learning   0.23916017  0.0536357004 0.11957391    1.00000000   3
advance    0.10625062 -0.0103177406 0.05991418    0.00000250   6
privileges 0.09267430 -0.0057118663 0.04446276    0.00040634   2
raises     0.09089754  0.0061503218 0.06011618    0.81769332   4
critical   0.08273046  0.0002573042 0.03992658    0.92899714   5

Mean no. regressors              Draws             Burnins
          "1.6114"               "64"                "0"
               Time  No. models visited      Modelspace 2^K
    "0.029845 secs"               "64"                "64"
         % visited        % Topmodels           Corr PMP
             "100"              "100"                "NA"
          No. Obs.        Model Prior             g-Prior
              "30"        "fixed / 2"               "UIP"
    Shrinkage-Stats
        "Av=0.9677"

Time difference of 0.029845 secs
```

As seen in `Mean no. regressors` and illustrated in the right panel of Figure 2, the posterior model size is now 1.61 which is somewhat smaller than with uniform model priors. Since posterior model size equals the sum of PIPs, many of them have also become smaller than in `att`. But interestingly, the PIP of `complaints` has remained at near 100%.

[11] Note that it is not necessary to specify `g = "UIP"` explicitly since it corresponds to the default setting of `bms`.

3.2. Custom prior inclusion probabilities

In view of the pervasive impact of `complaints`, one might wonder whether its importance would also remain robust to a greatly unfair prior. For instance, one could define a prior inclusion probability of only $\theta = 0.01$ for `complaints` while setting a "standard" prior inclusion probability of $\theta = 0.5$ for all other variables. Such a prior might be submitted to `bms` by assigning a vector of prior inclusion probabilities via its `mprior.size` argument:

```
R> att_pip <- bms(attitude, mprior = "pip",
+    mprior.size = c(0.01, 0.5, 0.5, 0.5, 0.5, 0.5), user.int = FALSE)
```

This implies a prior model size of $\bar{m} = 0.01 + 5 \times 0.5 = 2.51$.

The results can be obtained with `summary(att_pip)`:

```
R> summary(att_pip)
```

```
Mean no. regressors                Draws              Burnins
          "2.1262"                 "64"                  "0"
               Time   No. models visited       Modelspace 2^K
  "0.02080202 secs"                 "64"                 "64"
         % visited          % Topmodels             Corr PMP
              "100"                "100"                   "NA"
          No. Obs.          Model Prior              g-Prior
              "30"         "pip / 2.51"                "UIP"
    Shrinkage-Stats
        "Av=0.9677"
```

Accordingly, `complaints` still retains its PIP of near 100%. Posterior model size, however, decreases and all other variables obtain a far smaller PIP.

3.3. Binomial-beta model priors

Like the uniform prior, the fixed common θ in the binomial prior centers the mass of its distribution near the prior model size. A look on the prior model distribution with the following command shows that the prior model size distribution is quite concentrated around its mode, which is illustrated in Figure 3, left panel.

```
R> plotModelsize(att_pip)
```

This feature is sometimes criticized, in particular by Ley and Steel (2009): They note that to reflect prior uncertainty about model size, one should rather impose a prior that is less tight around prior expected model size. Therefore, Ley and Steel (2009) propose to put a *hyperprior* on the inclusion probability θ, effectively drawing it from a Beta distribution. In terms of researcher input, this prior again only requires to choose the prior expected model size. However, the resulting prior distribution is considerably less tight and should thus reduce the risk of unintended consequences from imposing a particular prior model size.[12]

[12]Therefore, the binomial-beta model prior with random θ is implemented as the default choice in `bms`.

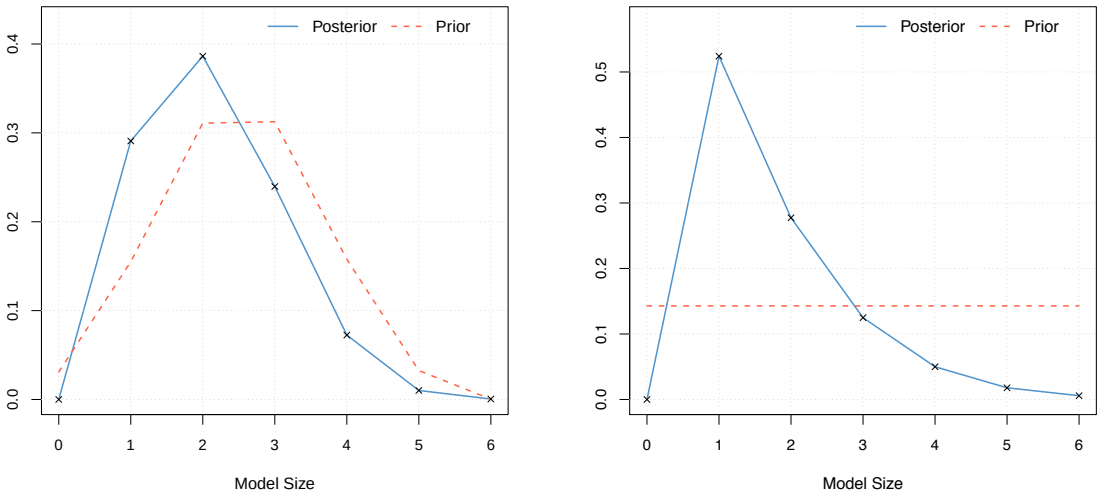

Figure 3: Distribution of posterior model size. Left panel: Custom prior on the model space employed. Right panel: Binomial-beta prior on the model space employed. Expected model size equals $K/2$ regressors.

In the vein of Ley and Steel (2009) a prior on the model space that is completely flat can be invoked by anchoring the binomial-beta prior on an expected model size of $K/2 = 6/2 = 3$ regressors:

```
R> att_random <- bms(attitude, mprior = "random", mprior.size = 3,
+    user.int = FALSE)
R> plotModelsize(att_random)
```

As Figure 3, right panel, illustrates, the prior on the model space is flat, while the posterior model size turns out to be 1.73. In terms of coefficient and posterior model size distribution, the results are very similar to those of `att_fixed`, even though the latter approach involved a tighter model prior. Concluding, a decrease of prior importance by the use of the binomial-beta framework supports the results found in `att_fixed`.

We can compare the PIPs from the four approaches presented so far with the following command:[13]

```
R> plotComp(Uniform = att, Fixed = att_fixed, PIP = att_pip,
+    Random = att_random)
```

Figure 4 illustrates that `att_fixed` (Fixed) and `att_random` (Random) lead to very similar results and are plainly smaller compared to the PIPs under `att` (Uniform).

Note that the Appendix A.1 contains an overview of the built-in model priors available in package **BMS**. Moreover, package **BMS** allows the user to define any custom model prior herself and straightforwardly use it in `bms` – for examples of these extensions, check the web page `http://bms.zeugner.eu/custompriors.php`. Another concept relating to model priors is to keep regressors fixed, i.e., to be included in every sampled model: Appendix A.4 provides some examples.

[13] This is equivalent to the command `plotComp(att, att_fixed, att_pip, att_random)`.

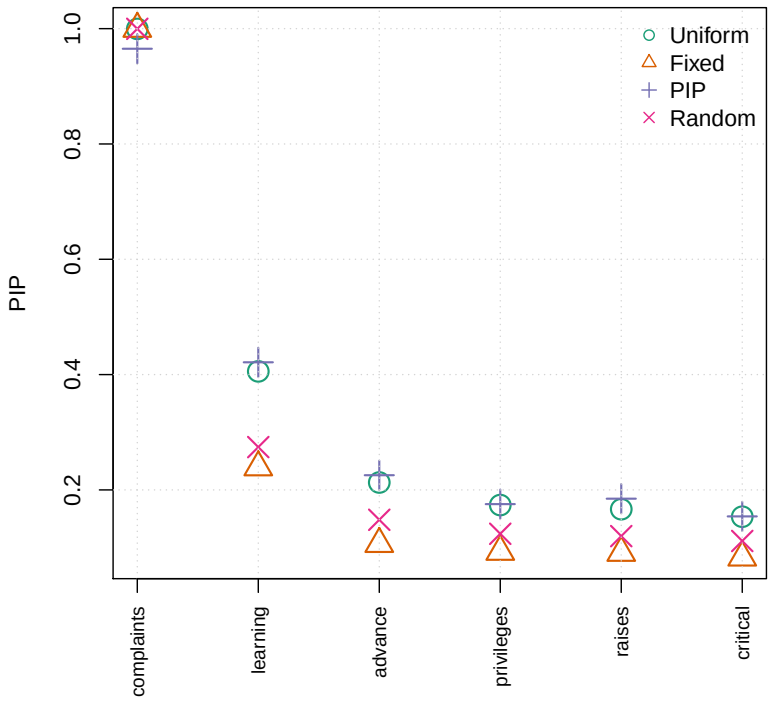

Figure 4: Posterior inclusion probabilities under different prior settings on the model space.

4. MCMC samplers and more variables

4.1. MCMC samplers

With a small number of variables, it is straightforward to enumerate all potential variable combinations to obtain posterior results. For a larger number of covariates, this becomes more time intensive: enumerating all models for 25 covariates takes about 3 hours on a modern PC, and doing a bit more already becomes infeasible: With 50 covariates for instance, there are more than a quadrillion ($\approx 10^{15}$) potential models to consider. In such a case, MCMC samplers gather results on the most important part of the posterior model distribution and thus approximate it as closely as possible. BMA mostly relies on the Metropolis-Hastings algorithm, which "walks" through the model space as follows:

At step i, the sampler stands at a certain "current" model M_i with PMP $p(M_i|y, X)$. In step $i + 1$ a candidate model M_j is proposed. The sampler switches from the current model to model M_j with probability $p_{i,j}$:

$$p_{i,j} = \min(1, p(M_j|y, X)/p(M_i|y, x)).$$

In case model M_j is rejected, the sampler moves to the next step and proposes a new model M_k against M_i. In case model M_j is accepted, it becomes the current model and has to survive against further candidate models in the next step. In this manner, the number of times each model is kept will converge to the distribution of posterior model probabilities $p(M_i|y, X)$.

In addition to enumerating all models, package **BMS** implements two MCMC samplers that differ in the way they propose candidate models:

- *Birth-death sampler* (`"bd"`): This is the standard model sampler used in most BMA routines. One of the K potential covariates is randomly chosen; if the chosen covariate forms already part of the current model M_i, then the candidate model M_j will have the same set of covariates as M_i but for the chosen variable ("dropping" a 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 ("adding" a variable).

- *Reversible-jump sampler* (`"rev.jump"`): Adapted to BMA by Madigan and York (1995) this sampler either draws a candidate by the birth-death method with 50% probability. In the other case (chosen with 50% probability) a "swap" is proposed, i.e., the candidate model M_j randomly drops one covariate with respect to M_i and randomly adds one chosen at random from the potential covariates that were not included in model M_i.

- *Enumeration* (`"enumerate"`): Up to fourteen covariates, complete enumeration of all models is the default option: This means that instead of an approximation by means of the aforementioned MCMC sampling schemes *all* possible models are evaluated. As enumeration becomes quite time-consuming or infeasible for many variables, the default option is `mcmc = "bd"` in case of $K > 14$, though enumeration can still be invoked with the command `mcmc = "enumerate"`.

The quality of an MCMC approximation to the actual posterior distribution depends on the number of draws the MCMC sampler runs for. In particular, the sampler has to start out from some model[14] that might not be a "good" one. Hence the first batch of iterations will typically not draw models with high PMPs as the sampler will only after a while converge to spheres of models with the largest marginal likelihoods. Therefore, this first set of iterations (the "burn-ins") is to be omitted from the computation of results. In `bms`, the argument `burn` specifies the number of burn-ins, and the argument `iter` the number of subsequent iterations to be retained.

4.2. An example: Economic growth

In one of the most prominent applications of BMA, Fernández, Ley, and Steel (2001b) analyze the importance of 41 explanatory variables on long-term term economic growth in 72 countries by the means of BMA. The data set is available in package **BMS**, a short description is available via `help("datafls")`. They employ a uniform model prior and the birth-death MCMC sampler. Their g prior is set to $g = \max(N, K^2)$, a mechanism such that PMPs asymptotically either behave like the Bayesian information criterion (with $g = N$) or the risk inflation criterion ($g = K^2$) – in `bms` this prior is assigned via the argument `g = "BRIC"`.

Moreover Fernández *et al.* (2001b) employ more than 200 million number of iterations after a substantial number of burn-ins. Since this would take quite a time, the following example reenacts their setting with only 50,000 burn-ins and 100,000 draws and will take about 30 seconds on a modern computer:

[14] `bms` has some simple algorithms implemented to choose "good" starting models; consult the option `start.value` under `help("bms")` for more information.

```
R> data("datafls", package = "BMS")
R> fls1 <- bms(datafls, burn = 50000, iter = 1e+05, g = "BRIC",
+    mprior = "uniform", nmodel = 2000, mcmc = "bd", user.int = FALSE)
```

Before looking at the coefficients, we can check for practical convergence of the MCMC chain by invoking the `summary` command:

```
R> summary(fls1)
```

```
Mean no. regressors              Draws             Burnins
          "10.5441"             "1e+05"             "50000"
               Time  No. models visited      Modelspace 2^K
    "21.74757 secs"             "26469"           "2.2e+12"
         % visited        % Topmodels            Corr PMP
          "1.2e-06"               "42"             "0.9169"
           No. Obs.         Model Prior             g-Prior
               "72"    "uniform / 20.5"              "BRIC"
    Shrinkage-Stats
        "Av=0.9994"
```

Note that due to stochastic variation in the MCMC chain your results might slightly differ from those reported here. Under `Corr PMP`, we find the correlation between iteration counts and analytical PMPs for the 2000 best models (the number 2000 was specified with the `nmodel = 2000` argument). At `summary(fls1)["Corr PMP"]`, this correlation is far from perfect but already indicates a good degree of convergence. For a closer look at convergence between analytical and MCMC PMPs, compare the actual distribution of both concepts:

```
R> plotConv(fls1)
```

Figure 5, left panel, presents the best 2,000 models encountered ordered by their analytical PMP (the red line), and plots their MCMC iteration counts (the blue line). For an even closer look, one might just check the corresponding image for just the best 100 models. The results are provided in Figure 5, right panel, and can be achieved with the following command:[15]

```
R> plotConv(fls1[1:100])
```

4.3. Analytical vs. MCMC likelihoods

The example above already achieved a decent level of correlation among analytical likelihoods and iteration counts with a comparatively small number of sampling draws. In general, the more complicated the distribution of marginal likelihoods, the more difficulties the sampler will meet before converging to a good approximation of PMPs. The quality of approximation may be inferred from the number of times a model got drawn vs. their actual marginal

[15] With `bma` objects such as `fls1`, the indexing parentheses `[]` are used to select subsets of the (ranked) best models retained in the object. For instance, while `fls1` contains 2,000 models, `fls1[1:100]` only contains the 100 best models among them. Correspondingly, `fls1[37]` would only contain the 37th best model. Cf., `help("[.bma")`

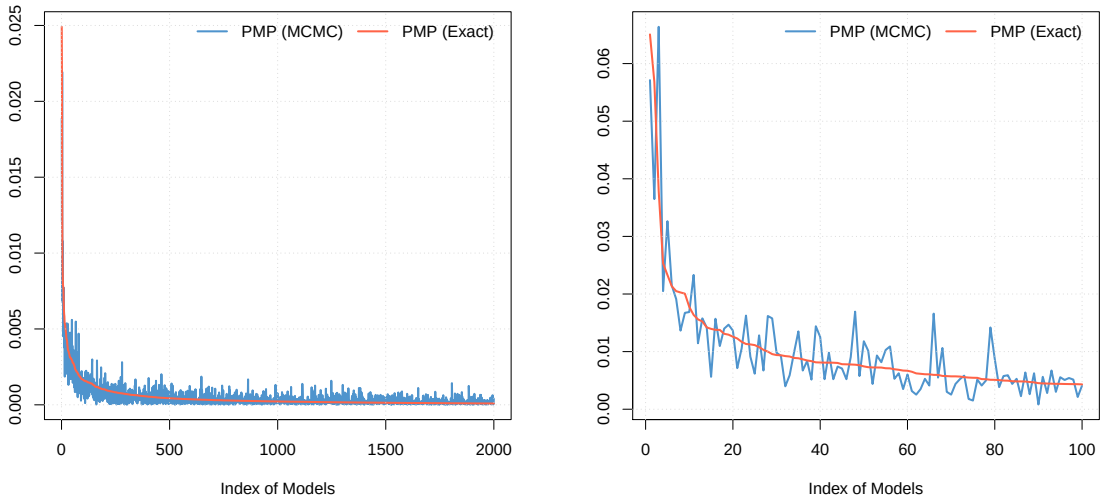

Figure 5: Convergence plot. Left panel illustrates the convergence of the MCMC chain based on 2,000 models, right panel for a subset of 100 models. The blue line corresponds to posterior model probabilities based on sampling frequencies, red line on its analytical counterpart. Convergence of the MCMC chain is achieved if both lines are strongly correlated.

likelihoods. Partly for this reason, `bms` retains a pre-specified number of models with the highest PMPs encountered during MCMC sampling, for which PMPs and draw counts are stored. Their respective distributions and their correlation indicate how well the sampler has converged.

However, due to RAM limits, the sampling chain can hardly retain more than a few 100,000 of these models. Instead, it computes aggregate statistics on-the-fly, taking iteration counts as model weights. For model convergence and some posterior statistics `bms` retains only the "top" (highest PMP) `nmodel` models it encounters during the iterations. Since the time for updating the iteration counts for the "top" models grows in line with their number, the sampler becomes considerably slower the more "top" models are to be kept. Still, if they are sufficiently numerous, those best models can already cover most of the posterior model mass – in this case it is feasible to base posterior statistics on analytical likelihoods instead of MCMC frequencies, just as in the enumeration case from Section 2. From `bms` results, the PMPs of "top" models may be displayed with the command `pmp`. For instance, one could display the PMPs of the best five models for `fls1` as follows:[16]

```
R> pmp(fls1)[1:5, ]
```

```
            PMP (Exact) PMP (MCMC)
0046845800c 0.010585312    0.00863
0046844800c 0.009244863    0.00750
00474440008 0.006160265    0.00650
00064450008 0.004119108    0.00430
00464440008 0.003792446    0.00282
```

[16] `pmp` returns a matrix with two columns and one row for each model. Consequently `pmp(fls1)[1:5, ]` extracts the first five rows and all columns of this matrix.

The numbers in the left-hand column represent analytical PMPs (`PMP (Exact)`) while the right-hand side displays MCMC-based PMPs (`PMP (MCMC)`). Both decline in roughly the same fashion, however sometimes the values for analytical PMPs differ considerably from the MCMC-based ones. This comes from the fact that MCMC-based PMPs derive from the number of iteration counts, while the "exact" PMPs are calculated from comparing the analytical likelihoods of the best models – cf., Equation 2.[17] In order to see the importance of all "top models" with respect to the full model space, we can thus sum up their MCMC-based PMPs as follows:

```
R> colSums(pmp(fls1))
```

```
PMP (Exact)  PMP (MCMC)
    0.42133     0.42133
```

Both columns sum up to the same number and show that in total, the top 2,000 models account for ca. 44% of posterior model mass.[18] This can be verified via:

```
R> round(colSums(pmp(fls1))[[2]], 2) * 100
```

They should thus provide a rough approximation of posterior results that might or might not be better than the MCMC-based results. For this purpose, compare the best 5 covariates in terms of PIP by analytical and MCMC methods: `coef(fls1)` will display the results based on MCMC counts.

```
R> coef(fls1)[1:5, ]
```

```
              PIP     Post Mean      Post SD Cond.Pos.Sign Idx
GDP60     0.99497 -0.0161529327 0.0032782851             0  12
Confucian 0.99054  0.0567793035 0.0144399600             1  19
LifeExp   0.92986  0.0008388346 0.0003466536             1  11
EquipInv  0.92738  0.1593294278 0.0677028844             1  38
SubSahara 0.75162 -0.0115418192 0.0082556040             0   7
```

In contrast, the results based on analytical PMPs will be invoked with the `exact` argument:

```
R> coef(fls1, exact = TRUE)[1:5, ]
```

```
                PIP    Post Mean      Post SD Cond.Pos.Sign Idx
GDP60     1.0000000 -0.016252921 0.0029270331             0  12
Confucian 0.9998987  0.056382088 0.0124665093             1  19
LifeExp   0.9671257  0.000849363 0.0002992502             1  11
EquipInv  0.9641156  0.165775220 0.0592304138             1  38
SubSahara 0.7846298 -0.011906481 0.0077020153             0   7
```

[17] In the call to `topmodels` on page 7, the PMPs under "MCMC" and analytical ("exact") concepts were equal since 1) enumeration bases both "top" model calculation and aggregate on-the-fly results on analytical PMPs and 2) because all possible models were retained in the object `att`.

[18] Note that this share was already provided in column `% Topmodels` resulting from the `summary` command on page 14.

The ordering of covariates in terms of PIP as well as the coefficients are roughly similar. However, the PIPs under `exact = TRUE` are somewhat larger than for the MCMC results. Closer inspection will also show that the analytical results downgrade the PIPs of the worst variables with respect to the PIPs under MCMC. This stems from the fact that analytical results do not take into account the many "bad" models that include "worse" covariates and are factored into MCMC results.

Whether to prefer analytical or MCMC results is a matter of taste – however the literature prefers coefficients the analytical way: Fernández *et al.* (2001b), for instance, retain 5,000 models and report results based on them.

4.4. Combining sampling chains

The MCMC samplers described in Section 4.1 need to discard the first batch of draws (the burn-ins) since they start out from some peculiar starting model and may reach the altitudes of "high" PMPs only after many iterations. Here, choosing an appropriate starting model may help to speed up convergence. By default `bms` selects its starting model as follows: from the full model[19], all covariates with OLS t-statistics > 0.2 are kept and included in the starting model. Other starting models may be assigned outright or chosen according to a similar mechanism (cf., argument `start.value` in `help("bms")`).

However, in order to improve the sampler's convergence to the PMP distribution, one might actually start from several different starting models. This could by particularly helpful if the models with high PMPs are clustered in distant "regions". For instance, one could set up the Fernández *et al.* (2001b) example above to get iteration chains from different starting values and combine them subsequently. Start, e.g., a shorter chain from the null model (the model containing just an intercept), and use the "reversible jump" MCMC sampler:

```
R> fls2 <- bms(datafls, burn = 20000, iter = 50000, g = "BRIC",
+    mprior = "uniform", mcmc = "rev.jump", start.value = 0,
+    user.int = FALSE)
R> summary(fls2)

Mean no. regressors                Draws              Burnins
          "10.5036"              "50000"              "20000"
               Time   No. models visited       Modelspace 2^K
    "10.84385 secs"              "10952"            "2.2e+12"
          % visited          % Topmodels             Corr PMP
            "5e-07"                 "26"             "0.8009"
           No. Obs.           Model Prior              g-Prior
               "72"     "uniform / 20.5"               "BRIC"
    Shrinkage-Stats
        "Av=0.9994"
```

Via:

```
R> round(cor(pmp(fls2))[2, 1], 2)
R> round(cor(pmp(fls1))[2, 1], 2)
```

[19] Actually, a model with randomly drawn $\min(K, N - 3)$ variables.

one can compare the degree of MCMC convergence for both models. Accordingly, with 0.86, the correlation between analytical and MCMC PMPs is a bit smaller than that from the `fls1` example in Section 4.3, which is 0.92. However, the results of this sampling run may be combined to yield more iterations and thus a better representation of the PMP distribution.

```
R> fls_combi <- c(fls1, fls2)
R> summary(fls_combi)
```

```
Mean no. regressors               Draws             Burnins
          "10.5306"            "150000"             "70000"
               Time  No. models visited      Modelspace 2^K
    "32.59142 secs"             "37421"           "2.2e+12"
          % visited         % Topmodels            Corr PMP
          "1.7e-06"                "37"            "0.9392"
           No. Obs.         Model Prior             g-Prior
              "72"    "uniform / 20.5"              "BRIC"
    Shrinkage-Stats
        "Av=0.9994"
```

With 0.95, the PMP correlation from the combined results is broadly better than either of its two constituent chains `fls1` and `fls2`. Still, the PIPs and coefficients do not change much with respect to `fls1` – as evidenced, e.g., by `plotComp(fls1, fls_combi, comp = "Std Mean")`.

5. Alternative formulations for Zellner's g prior

5.1. Alternative fixed g priors

Virtually all BMA applications rely on the presented framework with Zellner's g prior, and the bulk of them relies on specifying a fixed g. As mentioned in Section 1.2, the value of g corresponds to the degree of prior uncertainty: A low g renders the prior coefficient distribution tight around a zero mean, while a large g implies large prior coefficient variance and thus decreases the importance of the coefficient prior.

While some popular default elicitation mechanisms for the g prior (e.g., we have seen UIP and BRIC) are quite popular, they are also subject to severe criticism. Some (e.g Fernández, Ley, and Steel 2001a) advocate a comparatively large g prior to minimize prior impact on the results, stay close to the OLS coefficients, and represent the absolute lack of prior knowledge. Others (e.g., Ciccone and Jarociński 2010) demonstrate that such a large g may not be robust to noise innovations and risks over-fitting – in particular if the noise component plays a substantial role in the data. Again others (Eicher, Papageorgiou, and Raftery 2011) advocate intermediate fixed values for the g priors or present alternative default specifications (Liang, Paulo, Molina, Clyde, and Berger 2008).[20]

[20]Note however, that g should in general be monotonously increasing in N: Fernández *et al.* (2001a) prove that this is sufficient for "consistency", i.e., if there is one single linear model as in Equation 1, than its PMP asymptotically reaches 100% as sample size $N \to \infty$.

In package **BMS**, any fixed g prior may be specified directly by submitting its value to the `bms` function argument `g`. For instance, compare the results for the Fernández *et al.* (2001b) setting when a more conservative prior such as $g = 5$ is employed (and far too few iterations are performed):

```
R> fls_g5 <- bms(datafls, burn = 20000, iter = 50000, g = 5,
+    mprior = "uniform", user.int = FALSE)
R> coef(fls_g5)[1:5, ]

               PIP     Post Mean      Post SD Cond.Pos.Sign Idx
GDP60      0.99644 -0.0138831190 0.0039496898    0.00000000  12
Confucian  0.95382  0.0478522293 0.0202463636    1.00000000  19
LifeExp    0.89468  0.0007017799 0.0003933174    1.00000000  11
EquipInv   0.80270  0.1030272051 0.0744926697    1.00000000  38
SubSahara  0.74282 -0.0103114225 0.0087756753    0.00045772   7

R> summary(fls_g5)

Mean no. regressors              Draws            Burnins
          "20.1228"            "50000"            "20000"
               Time  No. models visited     Modelspace 2^K
    "11.58503 secs"            "43996"          "2.2e+12"
          % visited        % Topmodels           Corr PMP
            "2e-06"                "2"           "0.0856"
           No. Obs.        Model Prior            g-Prior
              "72"   "uniform / 20.5"          "numeric"
    Shrinkage-Stats
        "Av=0.8333"
```

The PIPs and coefficients for the best five covariates are comparable to the results from Section 4.2 but considerably smaller, due to a tight shrinkage factor of $\frac{g}{1+g} = \frac{5}{6}$ (cf., Section 1.2). More important, with 20.4, the posterior expected model size exceeds that of `fls_combi` by a large amount. This stems from the less severe size penalty imposed by eliciting a small g. Finally, with a correlation between analytical and MCMC PMPs of -0.03 we can conclude that the MCMC sampler has not at all converged yet. Feldkircher and Zeugner (2009) show that the smaller the g prior, the less concentrated is the PMP distribution, and therefore the harder it is for the MCMC sampler to provide a reasonable approximation to the actual PMP distribution. Hence the above command should actually be run with many more iterations in order to achieve meaningful results.

5.2. Model-specific g priors

The examples and references above illustrate that eliciting a fixed g prior common to all models can be fraught with difficulties and unintended consequences. Under a shrinkage factor close to 1 (i.e., under a large g), posterior estimates can easily overfit. This not only has implications for the estimated coefficients but also for the PIPs. A too large "overfitting" shrinkage factor leads to tight PMP concentrations and small model sizes, which result into

an unduly skewed PIP distribution. Consequently, an "overfitting" shrinkage factor attributes a relatively high PIP to just a few variables, while all other covariates yield very low PIPs. In contrast a too low shrinkage factor (i.e., low g) does not exploit the data signals, and typically leads to very similar intermediate PIPs for a large share of covariates. In order to address this issue, several authors have proposed to rely on model-specific "flexible" g priors (cf., Liang *et al.* 2008 for an overview). The virtue of such flexible shrinkage factors $\frac{g}{1+g}$ is that they adapt to the data: The better the signal-to-noise ratio, the closer the (expected) posterior shrinkage factor will be to 1, and vice versa. Consequently, the average shrinkage factor over all models can be interpreted as a Bayesian "goodness-of-fit" indicator. Feldkircher and Zeugner (2009) show how the model specific priors implemented in **BMS** can be interpreted in terms of the OLS F-statistic. Overall, there are two flexible g priors that allow for closed-form solutions and are implemented in package **BMS**:

- Empirical Bayes g – local (`"EBL"`): $g_\gamma = \arg\max_g \; p(y|M_\gamma, X, g)$. Authors such as George and Foster (2000) or Hansen and Yu (2001) advocate an "empirical Bayes" approach by using information contained in the data (y, X) to elicit g via maximum likelihood. This amounts to setting $g_\gamma = \max(0, F_\gamma^{OLS} - 1)$ where F_γ^{OLS} is the standard OLS F-statistic for model M_γ. Each single model thus has single shrinkage factor $\frac{g}{1+g}$ estimated in this way, but those shrinkage factors differ over models. The function `gdensity` provides density plots and data on the discrete distribution of the shrinkage factor over all models.

  Note that the local `"EBL"` prior is popular with some for its simplicity and effectiveness, while it is despised by others: It does not constitute a "real"' prior since it involves "peeking" at the data in order to formulate a prior. Moreover, asymptotic "consistency" of BMA is not guaranteed in this case.

- Hyper-g prior (`"hyper"`): Liang *et al.* (2008) propose putting a hyper-prior on g. In order to arrive at closed-form solutions, they suggest a Beta prior on the shrinkage factor of the form $\frac{g}{1+g} \sim B\left(1, \frac{a}{2} - 1\right)$, where a is a parameter in the range of $2 < a \leq 4$. Then, the prior expected value of the shrinkage factor is $\mathsf{E}(\frac{g}{1+g}) = \frac{2}{a}$. Moreover, setting $a = 4$ corresponds to the uniform prior distribution of $\frac{g}{1+g}$ over the interval $[0, 1]$, while $a \to 2$ concentrates prior mass very close to 1 (thus corresponding to $g \to \infty$). (`bms` allows to set a via the argument `g = "hyper = x"`, where `x` denotes the a parameter.) The virtue of the hyper-prior is that it allows for prior assumptions about g, but relies on Bayesian updating to adjust it. This limits the risk of unintended consequences on the posterior results, while retaining the theoretical advantages of a fixed g. Therefore Feldkircher and Zeugner (2009) prefer the use of hyper-g over other available g prior frameworks. In an application Feldkircher and Zeugner (2012) show that the use of the hyper-g prior leads to more robust results. Moreover, the hyper-g prior has the advantage over similar proposals that its closed-form posterior moments allow for computing them at a speed that is only slightly slower than fixed g priors.

Both model-specific g priors adapt to the data: The better the signal-to-noise ratio, the closer the (expected) posterior shrinkage factor will be to one, and vice versa. Therefore average statistics on the shrinkage factor offer the interpretation as a "goodness-of-fit" indicator. See also Feldkircher and Zeugner (2009) who show that both EBL and hyper-g can be interpreted

in terms of the OLS F-statistic.[21]

Consider, for instance, the Fernández *et al.* (2001b) example under an empirical Bayes prior:

```
R> fls_ebl <- bms(datafls, burn = 20000, iter = 50000, g = "EBL",
+    mprior = "uniform", nmodel = 1000, user.int = FALSE)
R> summary(fls_ebl)

Mean no. regressors              Draws              Burnins
          "20.5683"            "50000"            "20000"
               Time  No. models visited      Modelspace 2^K
    "12.32304 secs"            "28833"           "2.2e+12"
          % visited        % Topmodels           Corr PMP
          "1.3e-06"              "7.9"           "0.0651"
           No. Obs.        Model Prior            g-Prior
               "72"  "uniform / 20.5"              "EBL"
    Shrinkage-Stats
        "Av=0.9595"
```

The result `Shrinkage-Stats` reports a posterior average EBL shrinkage factor of 0.96, which corresponds to a shrinkage factor $\frac{g}{1+g}$ under $g \approx 24$. Consequently, posterior model size is considerably larger than under `fls_combi`, and the sampler has had a harder time to converge, as evidenced in a quite low `Corr PMP`. This can also be seen by invoking the `plot(fls_ebl)` command that yields a two-layer plot featuring the `plotModelsize` and `plotConv` plots illustrated previously.

The above results show that using a flexible and model-specific prior on Fernández *et al.* (2001b) data results in rather small posterior estimates of $\frac{g}{1+g}$, thus indicating that the `g = "BRIC"` prior used in `fls_combi` may be set too far from zero. This interacts with the uniform model prior to concentrate posterior model mass on quite large models. However, imposing a uniform model prior means to expect a model size of $K/2 = 20.5$, which may seem overblown. Instead, try to impose a smaller model size through a corresponding model prior – e.g., impose a prior model size of 7 as in Sala-i-Martin, Doppelhofer, and Miller (2004). This can be combined with a hyper-g prior, where the argument `g = "hyper = UIP"` imposes an a parameter such that the prior expected value of g corresponds to the unit information prior ($g = N$).[22]

```
R> fls_hyper <- bms(datafls, burn = 20000, iter = 50000, g = "hyper=UIP",
+    mprior = "random", mprior.size = 7, nmodel = 1000, user.int = FALSE)
R> summary(fls_hyper)
```

[21] For instance, the posterior expected value of the shrinkage factor for each model M_s with k_s parameters and R-squared R_S^2 is

$$\mathsf{E}\left(\frac{g}{1+g}|y, X_s, M_s\right) = \frac{1}{R_s^2(N-3-k_s-a+2)}\left(\frac{k_s+a-2}{{}_2F_1(\frac{N-1}{2}, 1, \frac{k_s+a}{2}, R_s^2)} - (k_s+a-2) + (N-3)R_s^2\right).$$

Its formulation mainly relates to R_s^2 and k_s and thus resembles a goodness-of-fit indicator. Feldkircher and Zeugner (2009) note that this also holds for its average across models: under some fairly standard conditions $\frac{1}{1-\mathsf{E}(\frac{g}{1+g}|X,y)}$ behaves similarly to an adjusted F-statistic of the model average.

[22] This is the default hyper-g prior and may therefore be as well obtained with `g = "hyper"`.

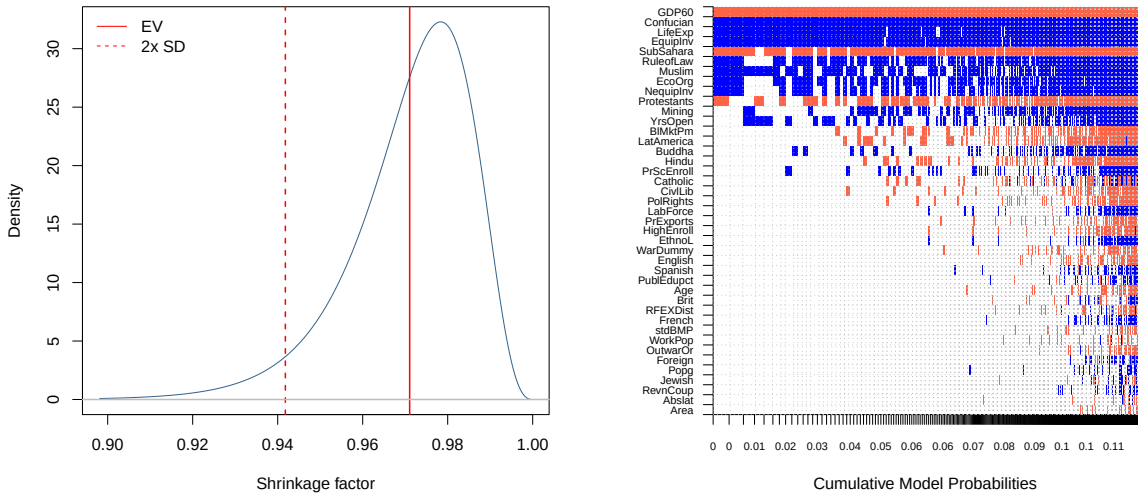

Figure 6: Left panel: Posterior density of the shrinkage factor $(g/(1+g))$. Solid line corresponds to the expected value $\mathsf{E}(g/1+g)|Y)$, dashed line to a $+/-$ 2 standard deviation interval. $\bar{m} = 3$. Right panel: Image plot. Blue color corresponds to a positive coefficient, red to a negative coefficient, and white to non-inclusion of the respective variable. The horizontal axis is scaled by the models' posterior model probabilities. Hyper-g prior employed.

```
      Mean no. regressors                  Draws                Burnins
                "15.0763"              "50000"              "20000"
                    Time      No. models visited      Modelspace 2^K
         "13.49753 secs"              "22745"           "2.2e+12"
               % visited          % Topmodels            Corr PMP
                 "1e-06"                 "14"              "0.3249"
                No. Obs.           Model Prior              g-Prior
                    "72"         "random / 7"  "hyper (a=2.02778)"
          Shrinkage-Stats
"Av=0.9613, Stdev=0.018"
```

From `Shrinkage-stats`, posterior expected shrinkage is 0.96 with rather tight standard deviation bounds. Similar to the EBL case before, the data thus indicates that shrinkage should be rather small (corresponding to a fixed g of $g \approx 24$) and not vary too much from its expected value. Since the hyper-g prior induces a proper posterior distribution for the shrinkage factor, it might be helpful to plot its density with the command below.

```
R> gdensity(fls_hyper)
```

Figure 6, left panel, confirms that posterior shrinkage is tightly concentrated around 0.94, which can be also verified via:

```
R> round(fls_hyper$gprior.info$shrinkage.moments[[1]] -
+    as.numeric(strsplit(summary(fls_hyper)[13], "=")[[1]][[3]]), 2)
```

While the hyper-g prior had an effect similar to the effect in the EBL case `fls_ebl`, the model prior now employed leaves the data more leeway to adjust posterior model size. The

results depart from the expected prior model size and point to an intermediate size of ca. 16. The focus on smaller models is evidenced by charting the best 1,000 models with the `image` command:

```
R> image(fls_hyper)
```

In a broad sense, the coefficient results correspond to those of `fls_combi`, at least in expected values. However, the results from `fls_hyper` were obtained under more sophisticated priors that were specifically designed to avoid unintended influence from prior parameters: By construction, the large shrinkage factor under `fls_combi` induced a quite small posterior model size of 10.4 and concentrated posterior mass tightly on the best models encountered (they make up 39% of the entire model mass). In contrast, the hyper-g prior employed for `fls_hyper` indicated a rather low posterior shrinkage factor and consequently resulted in higher posterior model size 16 and less model mass concentration 13%.

5.3. Posterior coefficient densities

In order to compare more than just coefficient expected values, it might be preferable to look at the entire posterior distribution of coefficients. For instance, the posterior density of the coefficient for `Muslim`, a variable with a PIP of 64%, can be generated via `density`:

```
R> density(fls_combi, reg = "Muslim")
```

The computed marginal posterior densities are a Bayesian model averaging mixture of the marginal posterior densities of the individual models. The accuracy of the result therefore depends on the number of "best" models contained in `fls_combi`. Note that the marginal posterior density can be interpreted as "conditional on inclusion": If the posterior inclusion probability of a variable is smaller than one, then some of its posterior density is Dirac at zero. Therefore the integral of the returned density vector adds up to the posterior inclusion probability, i.e., the probability that the coefficient is not zero.

The marginal densities of posterior coefficient distributions can be plotted with the argument `reg` specifying the variable under scrutiny.

The right panel of Figure 7 illustrates that the coefficient is neatly above zero, but somewhat skewed. The integral of this density will add up to 0.68:

```
R> round(estimates(fls_combi, exact = TRUE)["Muslim", 1], 3)
```

which roughly complies with the analytical PIP of `Muslim`. The vertical bars correspond to the analytical coefficient conditional on inclusion from `fls_combi` as in

```
R> coef(fls_combi, exact = TRUE, condi.coef = TRUE)["Muslim", ]
```

```
          PIP     Post Mean       Post SD Cond.Pos.Sign           Idx
 0.694049809   0.012790551   0.004541992   1.000000000  23.000000000
```

Note that the posterior marginal density is actually a model-weighted mixture of posterior densities for each model and can thus be calculated only for the top models contained in `fls_combi`.

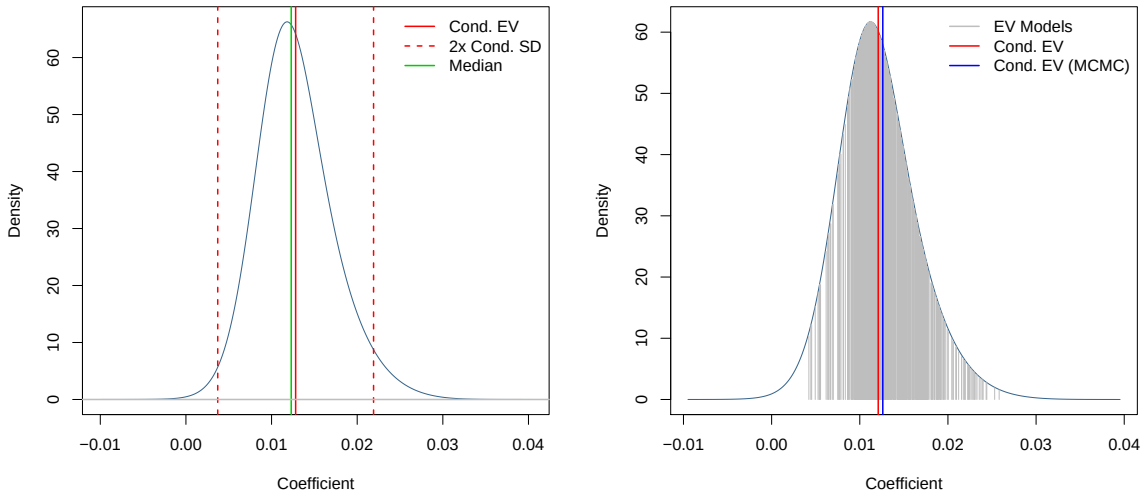

Figure 7: Posterior density attached to coefficient "muslim" conditional on inclusion. Left panel, results based on the "BRIC" prior, right panel based on the hyper-g prior. Cond. EV denotes the posterior expected value Cond. SD the posterior standard deviation, and median the median of the posterior distribution. All statistics based on "best" model likelihoods. The right panel shows on top of that the posterior expected value of the single best models, and the conditional expected value based on MCMC frequencies.

Now let us compare this density with the results under the hyper-g prior:[23]

```
R> dmuslim <- density(fls_hyper, reg = "Muslim", addons = "Eebl")
```

Figure 7, right panel, illustrates the posterior distribution of the coefficient `Muslim`. The `addons` argument assigns the vertical bars to be drawn: The expected conditional coefficient from MCMC (`E`) results should be indicated in contrast to the expected coefficient based on analytical PMPs (`e`). In addition, the expected coefficients under the individual models are plotted (`b`) and a legend is included (`l`). The density seems more symmetric than before and the analytical results a bit smaller than what could be expected from MCMC results.

Nonetheless, even though `fls_hyper` and `fls_combi` applied very different g and model priors, the results for the `Muslim` covariate are broadly similar: It is unanimously positive, with a conditional expected value somewhat above 0.01. In fact 95% of the posterior coefficient mass seems to be concentrated between 0.004 and 0.022.

```
R> quantile(dmuslim, c(0.025, 0.975))
```

```
       2.5%       97.5%
0.004019685 0.021635443
```

[23] Since for the hyper-g prior, the marginal posterior coefficient distribution derives from quite complicated expressions, executing this command could take a few seconds.

6. Predictive densities

Of course, BMA lends itself not only to inference, but also to prediction. The employed "Bayesian regression" models naturally give rise to predictive densities, whose mixture yields the BMA predictive density – a procedure very similar to the coefficient densities explored in the previous section.

Let us, for instance, use the information from the first 70 countries contained in `datafls` to forecast economic growth for the latter two, namely Zambia (identifier `ZM`) and Zimbabwe (identifier `ZW`). Based on their macro-fundamentals (i.e., the explanatory variables contained in `datafls`) we can form predictions invoking the function `pred.density`:

```
R> fcstbma <- bms(datafls[1:70, ], mprior = "uniform", burn = 20000,
+    iter = 50000, user.int = FALSE)
R> pdens <- pred.density(fcstbma, newdata = datafls[71:72, ])
```

The resulting object `pdens` holds the distribution of the forecast for the two countries, conditional on what we know from other countries, and the explanatory data from Zambia and Zimbabwe. The expected value of this growth forecast is very similar to the classical point forecast and can be accessed with `pdens$fit`.[24] Likewise the standard deviations of the predictive distribution correspond to classical standard errors and are returned by `pdens$std.err`. But the predictive density for economic growth in, e.g., Zimbabwe might be as well visualized with the following command:[25]

```
R> plot(pdens, 2)
```

Figure 8, left panel, shows that conditional on Zimbabwe's explanatory data, we expect growth to be concentrated around 0. And the actual value in `datafls[72, 1]` with 0.0046 is not too far off from that prediction. A closer look at both our densities with the function `quantile` shows that for Zimbabwe, any growth rate between -0.01 and 0.01 is quite likely.

```
R> quantile(pdens, c(0.05, 0.95))

             5%        95%
ZM  0.003284431 0.02752649
ZW -0.010804288 0.01154784
```

For Zambia (`ZM`), though, based on our regression model we would expect growth to be positive. Compared to Zimbabwe, however, economic growth over our evaluation period has been even worse.[26] Under the predictive density for Zambia, its realized value (-0.012) seems quite unlikely.

To compare the BMA prediction performance with actual outcomes, we could look, e.g., at the forecast error:

```
R> pdens$fit - datafls[71:72, 1]
```

[24] Note that this is equivalent to `predict(fcstbma, datafls[71:72, ])`.

[25] Here, 2 means to plot for the second forecasted observation, in this case `ZW`, the 72th row of `datafls`.

[26] Note that since `ZM` is the row name of the 71st row of `datafls`, this is equivalent to calling `datafls[71,]`.

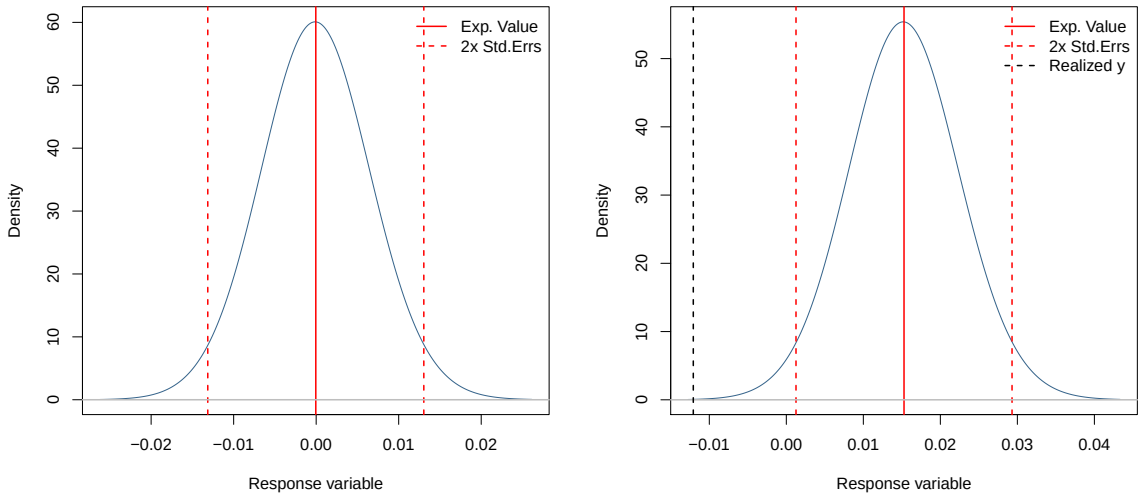

Figure 8: Posterior predictive density for economic growth in Zimbabwe (left panel) and Zambia (right panel). Solid red line denotes the expected value of the forecast, dashed red lines a +/− 2 standard deviations interval. In the right panel, the blue dashed line denotes the realized value for Zambian economic growth.

```
          ZM           ZW
0.027515629 -0.004212923
```

Taking a more Bayesian stance, one can evaluate the predictive distributions via:

```
R> pdens$dyf(datafls[71:72, 1])
```

```
[1]  0.06990264 47.91784502
```

The density for Zimbabwe is quite high (similar to the mode of the predictive density as seen in Figure 8, left panel), whereas the one for Zambia is quite low. In order to visualize how bad the forecast for Zambia was, compare a plot of predictive density to the actual outcome, which is situated far to the left.

```
R> plot(pdens, "ZM", realized.y = datafls["ZM", 1])
```

The results for Zambia might imply that the country can be considered as an outlier. One could also try different prior settings and compare the resulting models by their joint predictions for Zambia and Zimbabwe (or even more countries). Last, as opposed to evaluating the goodness of forecasts via its mean squared errors, we illustrate how to make use of the whole distribution of the forecast. Following Fernández *et al.* 2001a we calculate the "log-predictive score" (LPS), which is defined as follows:

$$-\sum_i \log(p(y_i^f|X, y, X_i^f)),$$

where $p(y_i^f|X, y, X_i^f)$ denotes predictive density for y_i^f (Zambian growth) based on the model information (y, X) (the first 70 countries) and the explanatory variables for the forecast observation (Zambian investment, schooling, etc.).

The log-predictive score can be accessed with `lps`.

```
R> lps(pdens, datafls[71:72, 1])

[1] -0.604418
```

If you compare different forecasting settings, the model that achieves a lower score is to be preferred. Note however, that the LPS is only meaningful when comparing different forecast settings.

7. Concluding remarks

The **BMS** package implements Bayesian model averaging for R. It excels in offering a range of widely used prior structures coupled with efficient MCMC algorithms to sort through the model space. Amini and Parmeter (2011) and Amini and Parmeter (2012) carry out a comparison of R software packages that implement Bayesian model averaging, in particular the packages **BAS** (Clyde 2012) and **BMA** (Raftery, Hoeting, Volinsky, Painter, and Yeung 2014). Amini and Parmeter (2012) conclude that package **BMS** is the only one among its competitors that is able to reproduce empirical results in Fernández *et al.* (2001b); Doppelhofer and Weeks (2009) and the working paper version of Masanjala and Papageorgiou (2008).

The **BMS** package is very flexible regarding the use of prior information. It allows for uniform and binomial-beta priors on the model space as well as informative prior inclusion probabilities. Via these "customized" model priors one can thus fuse prior beliefs into the otherwise purely agnostic analysis, that is prevalent in the applied literature using BMA. The **BMS** package also provides various specifications for Zellner's g prior including the so-called hyper-g priors advocated in Liang *et al.* (2008); Ley and Steel (2012); Feldkircher and Zeugner (2009). The sensitivity of BMA results to the specification of Zellner's g prior is well documented in the literature (Feldkircher and Zeugner 2012). It is thus of ample importance to offer a wide range of prior specifications in order to allow the user to carry out a serious sensitivity analysis.

Finally, the package comes along with numerous graphical tools to analyze posterior coefficient densities, the posterior model size or predictive densities. It also includes a graphical representation of the model space via an image plot. The flexibility and user-friendliness of the package is proved by the fact that various studies have used package **BMS** to perform Bayesian model averaging (among others, see Giannone, Lenza, and Reichlin 2011; Horváth 2011; Amini and Parmeter 2012; Babecký, Havránek, Matěju, Rusnák, Šmídková, and Vašíček 2013; Horváth 2013; Iršová and Havránek 2013; Feldkircher 2014; Feldkircher, Horváth, and Rusnák 2014; Horváth, Rusnák, Šmídková, and Zapal 2014). Future versions of **BMS** might include spatial regression models (Crespo Cuaresma and Feldkircher 2013; Crespo Cuaresma, Doppelhofer, and Feldkircher 2014), model averaging based on the predictive likelihood (Eklund and Karlsson 2007; Feldkircher 2012), so-called "heredity priors" to handle related predictors (Chipman 1996) and dilution priors that penalize models which contain highly collinear regressors (George 2010). For a detailed discussion on dilution and heredity priors see Moser and Hofmarcher (2014). An add-on to package **BMS** programmed also by Moser and Hofmarcher (2014) is available at `https://bitbucket.org/matmo/dilutbms2` featuring among others a dilution prior put forward by Durlauf, Kourtellos, and Tan (2012), the heredity priors proposed in Chipman (1996) and employed in Feldkircher *et al.* (2014); Feldkircher (2014) and the tessellation sampler proposed by George (2010) that accounts for

model space redundancy. The interested reader may be further referred to the "BMS-blog" section at http://bms.zeugner.eu/. The web page contains (video) tutorials on the usage of package **BMS** as well as further **BMS** add-on packages.

Acknowledgments

The opinions in this paper are those of the authors and do not necessarily coincide with those of the Oesterreichische Nationalbank. We would like to thank Jesús Crespo Cuaresma, Gernot Doppelhofer, Paul Hofmarcher, Eduardo Ley and Mark Steel for helpful comments.

References

Amini SM, Parmeter CF (2011). "Bayesian Model Averaging in R." *Journal of Economic and Social Measurement*, **36**(4), 253–287.

Amini SM, Parmeter CF (2012). "Comparisons of Model Averaging Techniques: Assessing Growth Determinants." *Journal of Applied Econometrics*, **27**(5), 870–876.

Babecký J, Havránek T, Matěju J, Rusnák M, Šmídková K, Vašíček B (2013). "Leading Indicators of Crisis Incidence: Evidence from Developed Countries." *Journal of International Money and Finance*, **35**, 1–19.

Chipman HA (1996). "Bayesian Variable Selection with Related Predictors." *Canadian Journal of Statistics*, **24**(1), 17–36.

Ciccone A, Jarociński M (2010). "Determinants of Economic Growth: Will Data Tell?" *American Economic Journal: Macroeconomics*, **2**(4), 222–246.

Clyde M (2012). ***BAS**: Bayesian Adaptive Sampling for Bayesian Model Averaging*. R package version 1.0, URL http://CRAN.R-project.org/package=BAS.

Crespo Cuaresma J, Doppelhofer G, Feldkircher M (2014). "The Determinants of Economic Growth in European Regions." *Regional Studies*, **48**(1), 44–67.

Crespo Cuaresma J, Feldkircher M (2013). "Spatial Filtering, Model Uncertainty and the Speed of Income Convergence in Europe." *Journal of Applied Econometrics*, **28**(4), 720–741.

Doppelhofer G, Weeks M (2009). "Jointness of Growth Determinants." *Journal of Applied Econometrics*, **24**(2), 209–244.

Durlauf SN, Kourtellos A, Tan CM (2012). "Is God in the Details? A Reexamination of the Role of Religion in Economic Growth." *Journal of Applied Econometrics*, **27**(7), 1059–1075.

Eicher TS, Papageorgiou C, Raftery AE (2011). "Default Priors and Predictive Performance in Bayesian Model Averaging, with Application to Growth Determinants." *Journal of Applied Econometrics*, **26**(1), 30–55.

Eklund J, Karlsson S (2007). "Forecast Combination and Model Averaging using Predictive Measures." *Econometric Reviews*, **26**(2–4), 329–362.

Feldkircher M (2012). "Forecast Combination and Bayesian Model Averaging: A Prior Sensitivity Analysis." *Journal of Forecasting*, **31**(4), 361–376.

Feldkircher M (2014). "The Determinants of Vulnerability to the Global Financial Crisis 2008 to 2009: Credit Growth and other Sources of Risk." *Journal of International Money and Finance*, **43**, 19–49.

Feldkircher M, Horváth R, Rusnák M (2014). "Exchange Market Pressures during the Financial Crisis: A Bayesian Model Averaging Evidence." *Journal of International Money and Finance*, **40**, 21–41.

Feldkircher M, Zeugner S (2009). "Benchmark Priors Revisited: On Adaptive Shrinkage and the Supermodel Effect in Bayesian Model Averaging." IMF Working Paper WP/09/202.

Feldkircher M, Zeugner S (2012). "The Impact of Data Revisions on the Robustness of Growth Determinants – A Note on 'Determinants of Economic Growth. Will Data Tell?'." *Journal of Applied Econometrics*, **27**(4), 686–694.

Feldkircher M, Zeugner S (2015). ***BMS**: Bayesian Model Averaging Library*. R package version 0.3.4, URL http://CRAN.R-project.org/package=BMS.

Fernández C, Ley E, Steel MF (2001a). "Benchmark Priors for Bayesian Model Averaging." *Journal of Econometrics*, **100**(2), 381–427.

Fernández C, Ley E, Steel MF (2001b). "Model Uncertainty in Cross-Country Growth Regressions." *Journal of Applied Econometrics*, **16**(5), 563–576.

George EI (2010). "Dilution Priors: Compensating for Model Space Redundancy." In JO Berger, TT Cai, IM Johnstone (eds.), *Borrowing Strength: Theory Powering Applications – Festschrift for Lawrence D. Brown*, volume Volume 6 of *Collections*, pp. 158–165. Institute of Mathematical Statistics, Beachwood, Ohio, USA.

George EI, Foster DP (2000). "Calibration and Empirical Bayes Variable Selection." *Biometrika*, **87**(4), 731–747.

Giannone D, Lenza M, Reichlin L (2011). "Market Freedom and the Global Recession." *IMF Economic Review*, **59**(1), 111–135.

Hansen MH, Yu B (2001). "Model Selection and the Principle of Minimum Description Length." *Journal of the American Statistical Association*, **96**(454), 746–774.

Hoeting JA, Madigan D, Raftery AE, Volinsky CT (1999). "Bayesian Model Averaging: A Tutorial." *Statistical Science*, **14**(4), 382–417.

Horváth R (2011). "Research & Development and Growth: A Bayesian Model Averaging Analysis." *Economic Modelling*, **28**(6), 2669–2673.

Horváth R (2013). "Does Trust Promote Growth?" *Journal of Comparative Economics*, **41**(3), 777–788.

Horváth R, Rusnák M, Šmídková K, Zapal J (2014). "Dissent Voting Behavior of Central Bankers: What Do We Really Know?" *Applied Economics*, **46**(4), 450–461.

Iršová Z, Havránek T (2013). "Determinants of Horizontal Spillovers from FDI: Evidence from a Large Meta-Analysis." *World Development*, **42**, 1–15.

Ley E, Steel MFJ (2009). "On the Effect of Prior Assumptions in Bayesian Model Averaging with Applications to Growth Regressions." *Journal of Applied Econometrics*, **24**(4), 651–674.

Ley E, Steel MFJ (2012). "Mixtures of g-Priors for Bayesian Model Averaging with Economic Applications." *Journal of Econometrics*, **171**(2), 251–266.

Liang F, Paulo R, Molina G, Clyde MA, Berger JO (2008). "Mixtures of g Priors for Bayesian Variable Selection." *Journal of the American Statistical Association*, **103**(481), 410–423.

Madigan D, York J (1995). "Bayesian Graphical Models for Discrete Data." *International Statistical Review*, **63**(2), 215–232.

Masanjala WH, Papageorgiou C (2008). "Rough and Lonely Road to Prosperity: A Reexamination of the Sources of Growth in Africa Using Bayesian Model Averaging." *Journal of Applied Econometrics*, **23**(5), 671–682.

Moser M, Hofmarcher P (2014). "Model Priors Revisited: Interaction Terms in BMA Growth Applications." *Journal of Applied Econometrics*, **29**(2), 344–347.

Raftery A, Hoeting J, Volinsky C, Painter I, Yeung KY (2014). ***BMA**: Bayesian Model Averaging*. R package version 3.18.1, URL http://CRAN.R-project.org/package=BMA.

R Core Team (2015). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Sala-i-Martin X, Doppelhofer G, Miller RI (2004). "Determinants of Long-Term Growth: A Bayesian Averaging of Classical Estimates (BACE) Approach." *American Economic Review*, **94**(4), 813–835.

A. Appendix

A.1. Available model priors – Synopsis

The following provides an overview over the model priors available in `bms`. Default is `mprior = "random"`. For details and examples on built-in priors, consult `help("bms")`. For defining different, custom g priors, consult `help("gprior")` or http://bms.zeugner.eu/custompriors.php.

Uniform model prior

- *Argument*: `mprior = "uniform"`.
- *Parameter*: none.
- *Concept*: $p(M_\gamma) \propto 1$.
- *Reference*: none.

Binomial model prior

- *Argument*: `mprior = "fixed"`.
- *Parameter* (`mprior.size`): prior model size $\bar{m}$ (scalar); default is $\bar{m} = K/2$.
- *Concept*: $p(M_\gamma) \propto \left(\frac{\bar{m}}{K}\right)^{k_\gamma} \left(1 - \frac{\bar{m}}{K}\right)^{K-k_\gamma}$.
- *Reference*: Sala-i-Martin *et al.* (2004).

Binomial-beta model prior

- *Argument*: `mprior = "random"`.
- *Parameter* (`mprior.size`): prior model size $\bar{m}$ (scalar).
- *Concept*: $p(M_\gamma) \propto \Gamma(1 + k_\gamma)\Gamma(\frac{K-m}{m} + K - k_\gamma)$; default is $\bar{m} = K/2$.
- *Reference*: Ley and Steel (2009).

Custom prior inclusion probabilities

- *Argument*: `mprior = "pip"`.
- *Parameter* (`mprior.size`): a vector of size K, detailing K prior inclusion probabilities π_i: $0 < \pi_i < 1 \; \forall i$.

- *Concept*: $p(M_\gamma) \propto \prod_{i\in\gamma} \pi_i \prod_{j\notin\gamma}(1-\pi_j)$.
- *Reference*: none.

Custom model size prior

- *Argument*: `mprior = "customk"`.
- *Parameter* (`mprior.size`): a vector of size $K+1$, detailing prior θ_j for 0 to K size models: any real > 0 admissible.
- *Concept*: $p(M_\gamma) \propto \theta_{k_\gamma}$.
- *Reference*: none.

A.2. Available g priors – Synopsis

The following provides an overview over the g priors available in `bms`. Default is `g = "UIP"`. For implementation details and examples, consult `help("bms")`. For defining different, custom g priors, consult `help("gprior")` or http://bms.zeugner.eu/custompriors.php.

Fixed g

- *Argument*: `g = x` where `x` is a positive real scalar.
- *Concept*: fixed g common to all models.
- *Reference*: Fernández *et al.* (2001a).
- *Sub-options*: Unit information prior `g = "UIP"` sets $g = N$; `g = "BRIC"` sets $g = \max(N, K^2)$, a combination of BIC and RIC. (Note that these two options guarantee asymptotic consistency.) Other options include `g = "RIC"` for $g = K^2$ and `g = "HQ"` for the Hannan-Quinn setting $g = \log(N)^3$.

Empirical Bayes (Local) g

- *Argument*: `g = "EBL"`.
- *Concept*: Model-specific g_γ estimated via maximum likelihood: amounts to $g_\gamma = \max(0, F_\gamma - 1)$, where $F_\gamma \equiv \frac{R^2_\gamma (N-1-k_\gamma)}{(1-R^2_\gamma)k_\gamma}$ and R^2_γ is the OLS R-squared of model M_γ.
- *Reference*: George and Foster (2000); Liang *et al.* (2008).
- *Sub-options*: none.

Hyper-g prior

- *Argument*: `g = "hyper"`.
- *Concept*: A Beta prior on the shrinkage factor with $p(\frac{g}{1+g}) = B(1, \frac{a}{2} - 1)$. Parameter a $(2 < a \leq 4)$ represents prior beliefs: $a = 4$ implies prior shrinkage to be uniformly distributed over $[0, 1]$, $a \to 2$ concentrates mass close to unity. Note that the prior expected value of the shrinkage facor is $\mathsf{E}(\frac{g}{1+g}) = \frac{2}{a}$.
- *Reference*: Liang *et al.* (2008); Feldkircher and Zeugner (2009).
- *Sub-options*: `g = "hyper = x"` with `x` defining the parameter a (e.g., `g = "hyper = 3"` sets $a = 3$). `g = "hyper"` resp. `g = "hyper = UIP"` sets the prior expected shrinkage factor equivalent to the UIP prior $\mathsf{E}(\frac{g}{1+g}) = \frac{N}{1+N}$; `g = "hyper = BRIC"` sets the prior expected shrinkage factor equivalent to the BRIC prior. Note that the latter two options guarantee asymptotic consistency.

A.3. "Bayesian regression" with Zellner's g – Bayesian model selection

The linear model presented in Section 1.2 using Zellner's g prior is implemented under the function `zlm`. For instance, we might consider the `attitude` data from Section 2 and estimate just the full model containing all 6 variables. For this purpose, first load the built-in data set with the command

```
R> data("attitude", package = "datasets")
```

The full model is obtained by applying the function `zlm` to the data set and storing the estimation into `att_full`. Zellner's g prior is estimated by the argument `g` just in the same way as in Section 5.[27]

```
R> att_full <- zlm(attitude, g = "UIP")
```

The results can then be displayed by using, e.g., the `summary` method.

```
R> summary(att_full)

Coefficients
               Exp.Val.   St.Dev.
(Intercept) 12.52405242        NA
complaints   0.59340736 0.1524868
privileges  -0.07069369 0.1285614
learning     0.30999882 0.1596262
raises       0.07909561 0.2097886
critical     0.03714334 0.1392373
advance     -0.21005485 0.1688040

 Log Marginal Likelihood:
-113.7063
 g-Prior: UIP
Shrinkage Factor: 0.968
```

[27] Likewise, most methods applicable to `bms`, such as `density`, `predict` or `coef`, work analogously for `zlm`.

The results are very similar to those resulting from OLS (which can be obtained using `summary(lm(attitude))`). The less conservative, i.e., the larger g becomes, the closer the results get to OLS. But remember that the full model was not the best model from the BMA application in Section 2. In order to extract the best encountered model, use the function `as.zlm` to extract this single model for further analysis (with the argument `model` specifying the rank-order of the model to be extracted). The following command reads the best model from the BMA results into the variable `att_best`.

```
R> att_best <- as.zlm(att, model = 1)
R> summary(att_best)

Coefficients
              Exp.Val.   St.Dev.
(Intercept) 15.9975134        NA
complaints   0.7302676 0.1010205

 Log Marginal Likelihood:
-107.4047
 g-Prior: UIP
Shrinkage Factor: 0.968
```

As suspected, the best model according to BMA is the one including only `complaints` and the intercept, as it has the highest log-marginal likelihood (`logLik(att_best)`). In such a way, the command `as.zlm` can be combined with `bms` for "Bayesian model selection", i.e., using the model prior and posterior framework to focus on the model with highest posterior mass. Via the utility `model.frame`, this best model can be straightforwardly converted into a standard OLS model:

```
R> att_bestlm <- lm(model.frame(as.zlm(att)))
R> summary(att_bestlm)

Call:
lm(formula = model.frame(as.zlm(att)))

Residuals:
     Min       1Q   Median       3Q      Max
-12.8799  -5.9905   0.1783   6.2978   9.6294

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 14.37632    6.61999   2.172   0.0385 *
complaints   0.75461    0.09753   7.737 1.99e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.993 on 28 degrees of freedom
Multiple R-squared: 0.6813,        Adjusted R-squared: 0.6699
F-statistic: 59.86 on 1 and 28 DF,  p-value: 1.988e-08
```

A.4. BMA when keeping a fixed set of regressors

While BMA should usually compare as many models as possible, some considerations might dictate the restriction to a subspace of the 2^K models. For complicated settings one might employ a customary designed model prior (cf., Section A.1). The by far most common setting, though, is to keep some regressors fixed in the model setting, and apply Bayesian model uncertainty only to a subset of regressors.

Suppose, for instance, that prior research tells us that any meaningful model for `attitude` (as in Section 2) must include the variables `complaints` and `learning`. The only question is whether the additional four variables matter (which reduces the potential model space to $2^4 = 16$). We thus sample over these models while keeping `complaints` and `learning` as fixed regressors:

```
R> att_learn <- bms(attitude, mprior = "uniform",
+    fixed.reg = c("complaints", "learning"))

                 PIP    Post Mean    Post SD Cond.Pos.Sign Idx
complaints 1.0000000  0.622480469 0.12718297     1.0000000   1
learning   1.0000000  0.237607970 0.15086061     1.0000000   3
advance    0.2878040 -0.053972968 0.11744640     0.0000000   6
privileges 0.1913388 -0.017789715 0.06764219     0.0000000   2
raises     0.1583504  0.001767835 0.07951209     0.3080239   4
critical   0.1550556  0.002642777 0.05409412     1.0000000   5

Mean no. regressors              Draws             Burnins
          "2.7925"                "16"                "0"
               Time  No. models visited      Modelspace 2^K
 "0.008205175 secs"                "16"                "64"
         % visited         % Topmodels            Corr PMP
              "25"               "100"                "NA"
          No. Obs.         Model Prior             g-Prior
              "30"       "uniform / 4"               "UIP"
    Shrinkage-Stats
        "Av=0.9677"

Time difference of 0.008205175 secs
```

The results show that the PIP and the coefficients for the remaining variables increase a bit compared to `att`. The higher PIPs are related to the fact that the posterior model size (as in `sum(coef(att_learn)[, 1])`) is quite larger as under `att`. This follows naturally from our model prior: putting a uniform prior on all models between parameter size 2 (the base model) and 6 (the full model) implies a prior expected model size of 4 for `att_learn` instead of the 3 for `att`.[28] So to achieve comparable results, one needs to take the number of fixed regressors into account when setting the model prior parameter `mprior.size`. Consider another example: Suppose we would like to sample the importance and coefficients for the

[28]The command `att_learn2 = bms(attitude, mprior = "fixed", mprior.size = 3, fixed.reg = c("complaints", "learning"))` produces coefficients that are much more similar to `att`.

cultural dummies in the dataset `datafls`, conditional on information from the remaining "hard" variables. This implies keeping 27 fixed regressors, while sampling over the 14 cultural dummies. Since model uncertainty thus applies only to $2^{14} = 16,384$ models, we resort to full enumeration of the model space.

```
R> fls_culture <- bms(datafls, fixed.reg = c(1, 8:16, 24, 26:41),
+    mprior = "random", mprior.size = 28, mcmc = "enumerate",
+    user.int = FALSE)
```

Here, the vector `c(1, 8:16, 24, 26:41)` denotes the indices of the regressors in `datafls` to be kept fixed.[29] Moreover, we use the binomial-beta ("random") model prior. The prior model size of 30 embodies our prior expectation that on average 1 out of the 14 cultural dummies should be included in the true model. As we only care about those 14 variables, let us just display the results for the 14 variables with the least PIP:

```
R> coef(fls_culture)[28:41, ]
```

```
                 PIP     Post Mean      Post SD Cond.Pos.Sign Idx
Confucian  0.99950018  6.796387e-02 0.0130198193    1.00000000  19
Hindu      0.94793751 -7.519094e-02 0.0270173174    0.00000000  21
SubSahara  0.84127891 -1.584077e-02 0.0091307736    0.00000000   7
EthnoL     0.70497895  9.238430e-03 0.0070484067    0.99999675  20
Protestants 0.57157577 -6.160916e-03 0.0061672877    0.00001431  25
Muslim     0.53068726  7.908574e-03 0.0086009027    0.99999949  23
LatAmerica 0.52063035 -6.538488e-03 0.0074843453    0.00469905   6
Spanish    0.21738032  2.105990e-03 0.0047683535    0.98325917   2
French     0.17512267  1.065459e-03 0.0027682773    0.99999954   3
Buddha     0.11583307  9.647944e-04 0.0033462133    0.99999992  17
Brit       0.10056773  4.095469e-04 0.0017468022    0.94203569   4
Catholic   0.09790780 -1.072004e-05 0.0019274111    0.45246829  18
WarDummy   0.07478332 -1.578379e-04 0.0007599415    0.00123399   5
Jewish     0.04114852 -5.614758e-05 0.0018626191    0.24834675  22
```

As before, we find that `Confucian` (with positive sign) as well as `Hindu` and `SubSahara` (negative signs) have the most important impact conditional on "hard" information. Moreover, the data seems to attribute more importance to cultural dummies as we expected with our model prior: Comparing prior and posterior model size with the following command shows how much importance is attributed to the dummies.

```
R> plotModelsize(fls_culture, ksubset = 27:41)
```

Figure 9 shows that expected posterior model size is close to 33, which means that 6 out of the cultural dummies should actually be included in a "true" model.

[29] Here, indices start from the first regressor, i.e., they do not take the dependent variable into account. The fixed data used above therefore corresponds to `datafls[, c(1, 8:16, 24, 26:41) + 1]`.

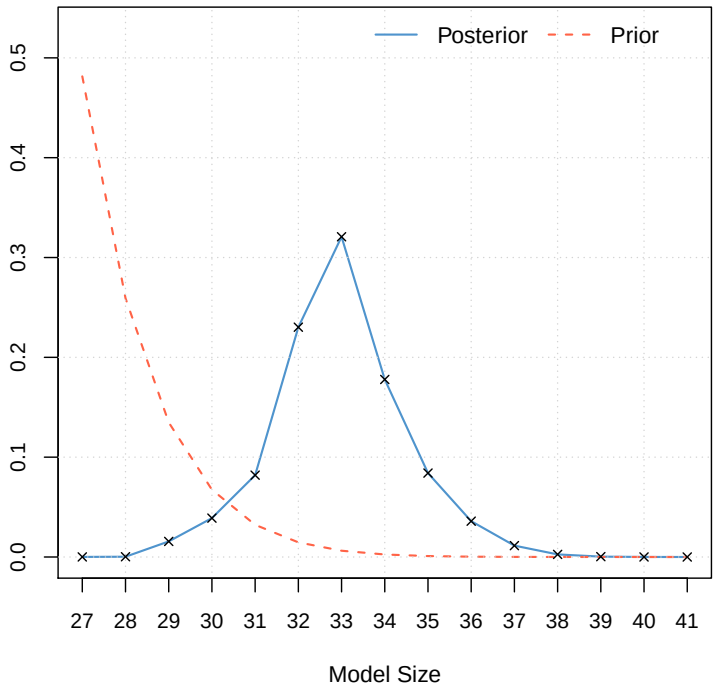

Figure 9: Posterior distribution of model size. Example using fixed regressors.

Affiliation:

Stefan Zeugner
Université Libre de Bruxelles
Avenue F. D. Roosevelt 50
CP 114, Bruxelles 1050, Belgium
E-mail: stefan.zeugner@gmail.com
URL: http://bms.zeugner.eu/

Martin Feldkircher
Oesterreichische Nationalbank
Otto-Wagner Platz 3
A-1090 Wien, Austria
E-mail: martin.feldkircher@oenb.at

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
November 2015, Volume 68, Issue 4 *Submitted:* 2012-07-30
doi:10.18637/jss.v068.i04 *Accepted:* 2014-10-15