gretl working papers

BACE: A **gretl** Package for Model Averaging in Limited Dependent Variable Models

Marcin Błażejowski
WSB University in Torun
marcin.blazejowski@wsb.torun.pl

Jacek Kwiatkowski
Nicolaus Copernicus University in Toruń
jkwiat@umk.pl

working paper #6

March 14, 2023

Abstract

This paper presents a software package called **BACE** (Bayesian Averaging of Classical Estimates) which offers model-building strategy for various limited dependent variable models, including logit and probit models, ordered logit and probit models, multinomial logistic regression, Poisson regression, Tobit model, and interval regression. **BACE** strategy is a model selection method that incorporates both classical estimation and Bayesian techniques. It solves the problem of computation speed and model uncertainty that arise when dealing with a large number of competing advanced statistical models. Our **BACE** package is both fast and capable of delivering consistent results. The package also provides implementation of the latest proposals of BIC variants, and the latest measures of jointness. We use **gretl**, a popular, free, and open-source software for econometric analysis that features an easy-to-use graphical user interface.

Keywords: BMA, model selection, BIC, **gretl**, `Hansl`

Contents

1 Introduction **2**

2 Methodological background **3**
- 2.1 Model averaging and marginal data density approximation 3
- 2.2 The limited dependent variable models 8
- 2.3 Jointness measures 10

3 Implementation in gretl **12**
- 3.1 The core of the **BACE** code 12
  - 3.1.1 `BACE()` 12
  - 3.1.2 `BACE_GUI()` 15
  - 3.1.3 `BACE_Print()` 16
  - 3.1.4 `BACE_Plot()` 17

4 Usage of the BACE package **17**
- 4.1 Output 20

5 Empirical illustration and speed of calculation **27**
- 5.1 Logistic regression example 27
- 5.2 Poisson regression example 30
- 5.3 Speed of calculation 31

6 Conclusions **33**

1 Introduction

In articles Raftery (1995) and Sala-i-Martin et al. (2004), the authors suggest a technique for averaging parameter estimates called Bayesian Averaging of Classical Estimates (**BACE**). This technique enables the measurement of the importance of potential regressors and is an alternative to the familiar and previously applied Bayesian Model Averaging (BMA) technique. It differs mainly in that it is based on Ordinary Least Squares (OLS) or Maximum Likelihood estimation (MLE) estimates[1]. The averaging of the parameter estimates is performed across all possible combinations of regression models obtained by means of classical frequentist inference, and hence is known as Frequentist Model Averaging (FMA).

A key element in model averaging with Bayesian inference is calculating the marginal data density. However, in many cases, the combination of the likelihood function and a prior distribution leads to posterior distributions that are difficult to integrate analytically. This difficulty arises even when integrating each component is mathematically easy, such as in the case of a Normal likelihood with a Beta prior. While obtaining the marginal data density in Bayesian linear regression is relatively easy, models with the limited dependent variables, such as logit and probit, are much more CPU time consuming, and require complex numerical techniques (see Bollen et al., 2014; Pajor, 2017; Lucchetti and Pedini, 2022). It is a well-known fact that as the number of explanatory variables grows, the number of possible models increases exponentially, which may pose problems for the Bayesian approach in advanced statistical models. The **BACE** method solves this problem by requiring significantly less computational time through the use of marginal data density approximation, such as the Schwarz criterion (Schwarz, 1978). It also provides a reliable results based on a weighted average over all individual models, and in addition we can show that it is able to replicate the complex reduction strategy (Błażejowski et al., 2020).

Our approach follows directly ideas presented in Kass and Raftery (1995); Sala-i-Martin et al. (2004); Bollen et al. (2014), where authors use Schwarz criterion to calculate approximation of the marginal data density and the odds ratio. An example of the application of the **BACE** technique for forecasting macroeconomic indicators can be found, among others, in the articles by van Dijk (2004); Moral-Benito (2012); Albis and Mapa (2014) and Białowolski et al. (2014). A recent overview of the applications of the **BACE** can be found in Steel (2020).

In this paper, we introduce Bayesian Averaging of Classical Estimates package for **gretl**, named **BACE**. In our opinion, the presented package contains several interesting features that are worth mentioning. Our **BACE** package is very fast and accurate set of functions. We provide an averaging technique for several limited de-

---

[1] The technique of model averaging using Maximum Likelihood estimation is sometimes referred to as Bayesian Averaging of Maximum Likelihood Estimates (BAMLE), as described in Moral-Benito (2012). For simplicity in our notation, we use the term **BACE** to refer to both variants of Frequentist Model Averaging, whether using OLS or MLE.

pendent variable models, namely: logit and probit models, ordered logit and probit models, multinomial logistic regression, Poisson regression, Tobit model, and interval regression. To our knowledge, most of them at this moment are only available in the **BACE** package. We use **gretl**, which is a popular, free, and open-source software for econometric analysis with an easy-to-use graphical user interface (GUI) for model averaging.

While there are many packages in the literature dedicated to model averaging for standard regression (Amini and Parmeter, 2011; Clyde et al., 2012; Zeugner and Feldkircher, 2015; Błażejowski and Kwiatkowski, 2015), to our knowledge, only a few packages have considered discrete dependent variable models (Lucchetti and Pedini, 2022; Raftery et al., 2022; Sevcikova and Raftery, 2022). In both recent cases, the authors use generalized linear models to use limited dependent variable models, such as logit, probit, and Poisson specifications. In the first example, Lucchetti and Pedini (2022) introduced the **ParMA** package for **gretl**, where the authors use a Bayesian approach with a reversible jump Markov chain Monte Carlo scheme. In the second case, Raftery et al. (2022) and Sevcikova and Raftery (2022) developed the **BMA** and **mlogitBMA** packages in **R** using a frequency approach based on the Bayesian Information Criterion (BIC). In terms of models with a limited dependent variable, our package can be considered an alternative to the previously mentioned packages because it offers a large number of limited variable specifications, fast computational speed, good accuracy, the latest proposals of BIC variants, and the latest jointness measures.

The remainder of this article is structured as follows: Section 2 briefly outlines the model averaging and marginal data density approximation used in our package, as well as the implemented limited dependent variable models and measures of jointness. Section 3 presents the basic idea of the **BACE** package, including its code, the use of command line and GUI interfaces, and the outputs that are returned. Section 4 includes a usage example for the ordered probit model. Section 5 presents an empirical illustration of logistic regression and Poisson regression, as well as information about the speed of calculation of **BACE** package. The final section concludes.

2 Methodological background

2.1 Model averaging and marginal data density approximation

As we know, probability can be used to describe the degree of belief associated with different specifications when comparing them using the Bayesian approach. For example, based on posterior probability, we can quickly compare two mutually exclusive (non-nested) and jointly exhaustive models with various subsets of variables. Hence, the posterior odds ratio for a model M_l versus model M_n can be obtained using the following formula:

$$\frac{\mathsf{P}(M_l \mid y)}{\mathsf{P}(M_n \mid y)} = \frac{\mathsf{P}(M_l)}{\mathsf{P}(M_n)} \frac{p(y \mid M_l)}{p(y \mid M_n)}, \tag{1}$$

where $\frac{\mathsf{P}(M_l)}{\mathsf{P}(M_n)}$ is the prior odds ratio, $\frac{p(y|M_l)}{p(y|M_n)}$ is the Bayes factor, $p(y \mid M_j)$ and $p(y \mid M_n)$ are marginal data densities.

If the posterior ratio $\frac{\mathsf{P}(M_l|y)}{\mathsf{P}(M_n|y)}$ is greater than one, it indicates that the data supports model M_l over model M_n.

As we can easily see, the key element needed to calculate the Bayes factor is the normalizing constant, which is given by the equation:

$$p(y \mid M_r) = \int p(y \mid \theta_r, M_r) p(\theta_r \mid M_r) d\theta_r, \tag{2}$$

where θ_r is the vector of unknown parameters, $p(y|\theta_r, M_r)$ can be interpreted as likelihood function under model M_r, and $p(\theta_r|M_r)$ is a prior distribution of the parameters θ_r.

In some relatively simple models, such as linear regression, the marginal data density can be represented by a closed-form analytic expression (Zellner, 1971; Fernández et al., 2001; Koop, 2003). However, in many situations, even though it is a simple concept, the marginal data density is difficult to calculate, especially in high-dimensional models where numerical integration is more computationally demanding (see Newton and Raftery, 1994; Gelman et al., 2013; Pajor, 2017). This is particularly problematic in model averaging, where we can have a large number of possible explanatory variables (K) and therefore many potential combinations (2^K) to calculate in a relatively short time. For this reason, one possible solution to overcome this problem is to use an approximation of the integral in Equation 2 by Schwarz criterion (Schwarz, 1978). The **BACE** method with Schwarz criterion refers to this solution by conducting a fast algorithm of model averaging with a reasonably accurate marginal data density approximation. Additionally, it is worth noting that the **BACE** method also provides a good approximation of estimated characteristics and it is capable of replicating the complex reduction strategy (Błażejowski et al., 2020).

In our **BACE** package, we use the latest approximations of $p(y \mid M_r)$, which are popular in the frequentist approach, and we briefly present them further in this paper (Raftery, 1996; Bollen et al., 2012, 2014; Peterson and Cavanaugh, 2019; Bayarri et al., 2019):

1. Standard Bayesian information criterion (BIC) introduced by Schwarz (1978), one of the most popular model selection method, based on the following assumption:

$$\ln p(y \mid M_r) \approx \ln p(y \mid \hat{\theta}_r, M_r) - \frac{k_r}{2} \ln N, \tag{3}$$

   where $\hat{\theta}_k$ is maximum likelihood estimator of parameters θ_k, k_r is a number of estimated parameters within the model M_r and N is the number of observations.

2. Haughton's BIC (HBIC) – variant with additional retained term of Taylor expansion of the $\ln p(y \mid M_k)$ around the MLE estimator (Haughton, 1988; Kass and Raftery, 1995):
$$\ln p(y \mid M_r) \approx \ln p(y \mid \hat{\theta}_r, M_r) - \frac{k_r}{2} \ln N + \frac{k_r}{2} \ln(2\pi). \tag{4}$$
3. The Information Matrix-Based BIC (IBIC) proposed by Bollen et al. (2012), which includes another part of the Taylor expansion, this time Fisher information matrix $I(\hat{\theta})$ evaluated at $\hat{\theta}$:
$$\ln p(y \mid M_r) \approx \ln p(y \mid \hat{\theta}_r, M_r) - \frac{k_r}{2} \ln N + \frac{k_r}{2} \ln(2\pi) - \frac{1}{2} \ln \left| I(\hat{\theta}) \right|. \tag{5}$$
4. Extended Bayesian Information Criteria (EBIC) introduced by Chen and Chen (2008) – standard BIC with additional expression that penalizes a model based on the number of candidate models of the same size:
$$\mathrm{EBIC} = \mathrm{BIC} - \gamma \ln \binom{K}{k_r}. \tag{6}$$

Using **BACE**, we can also easily evaluate the mean and variance of the parameters θ across the entire model space (Leamer, 1978):

$$E(\theta \mid y) \approx \sum_{r=1}^{2^K} \Pr(M_r \mid y) E(\theta_r \mid y, M_r), \tag{7}$$

$$\begin{aligned} Var(\theta \mid y) \approx & \sum_{r=1}^{2^K} \Pr(M_r \mid y) Var(\theta_r \mid y, M_r) + \\ & + \sum_{r=1}^{2^K} \Pr(M_r \mid y) \left(E(\theta_r \mid y, M_r) - E(\theta \mid y)\right)^2, \end{aligned} \tag{8}$$

where $E(\theta_r \mid y, M_r)$ and $Var(\theta_r \mid y, M_r)$ are the OLS (or MLE) estimates of θ_r from model M_r and 2^K denotes total model space.

Following Ley and Steel (2009) let us assume that each regressor enters a model independently of the others with prior probability $\theta \in [0, 1]$ (the prior probability of inclusion). In our **BACE** package we use two popular variants of prior model size (W) distribution (see Ley and Steel, 2009; Steel, 2011):

- Binomial (Forsyth, 2018):
$$\mathsf{P}(W = k_r) = \binom{K}{k_r} \mathsf{P}(M_r), \tag{9}$$
  with corresponding prior model probabilities as
$$\mathsf{P}(M_r) = \theta^{k_r} (1 - \theta)^{K - k_r}, \tag{10}$$
  and expected model size $\mathsf{E}(W) = K\theta$;

- Beta-binomial (Wakefield, 2013):

  $$\mathsf{P}(W = k_r) = \binom{K}{k_r}\mathsf{P}(M_r), \tag{11}$$

  where we assume that θ follows a Beta distribution with hyperparameters a and b, both of which are greater than zero. This results in the following corresponding prior model probabilities[2]:

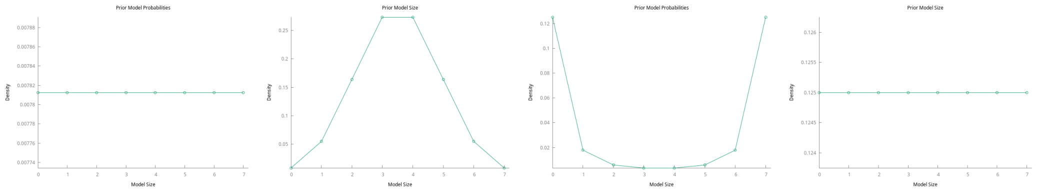
  $$\mathsf{P}(M_r) = \frac{\Gamma(K+1)}{\Gamma(k_r+1)\Gamma(K-k_r+1)} \cdot \frac{\Gamma(a+k_r)\Gamma(K+b-k_r)}{\Gamma(a+b+k_r)} \cdot \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \tag{12}$$

  with expected model size $\mathsf{E}(W) = K\frac{a}{a+b}$. Furthermore, following Ley and Steel (2009), we use fixed value of $a = 1$.

We assume that the user can specify just one hyperparameter for the model prior, i.e., the expected model size $\mathsf{E}(W) = W^\star \in (0, K)$. Our **BACE** package will automatically calculate the prior probabilities for all competitive models. This is a very convenient and flexible solution, similar to approaches known from Fernández et al. (2001) and Sala-i-Martin et al. (2004).

In the case of a Binomial prior, if we specify $W^\star = \frac{K}{2}$, which is the average number of potential independent variables, then we have $\theta = 0.5$, and thus, the prior model probability distribution is uniform. An example for $K = 7$ is shown in Figure 1a, together with the corresponding prior model size distribution shown in Figure 1b.

In the case of a Beta-binomial prior, if we specify $W^\star = \frac{K}{2}$, then we have $b = 1$, which results in a uniform prior distribution for the model size. An example for $K = 7$ is shown in Figure 1d, together with the corresponding prior model probabilities shown in Figure 1c.



(a) (b) (c) (d)

Figure 1: Prior model probabilities with associated prior model size for $K = 7$ and $W^\star = \frac{K}{2}$, when model includes constant term

Sometimes, in a regression model, we do not include a constant term (see Krolzig and Hendry, 2001; Błażejowski et al., 2020). This modelling strategy means that the

---

[2] The reason why we use the gamma function here instead of the beta function is that **gretl** provides the former but not the latter.

model space is reduced to 2^{K-1} possible combinations by excluding one specification that consists of only a constant term, i.e., a model with no explanatory variables. As a consequence, we should reflect this in the **BACE** modeling by using a truncated prior model size distribution.

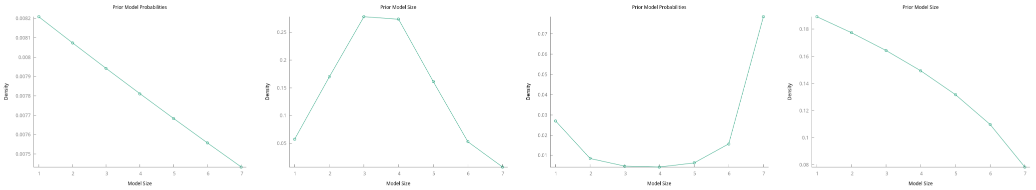
From the definition of conditional probability, we can define the density of a truncated random variable as (see Greene, 1999, pp 864):

$$f\left(x|x>a\right)=\frac{f(x)}{\mathsf{P}(x>0)}. \tag{13}$$

Because in our package, we assume that the user defines only the expected model size $\mathsf{E}(W)=W^{\star}$, we have to rescale the prior model size distribution so that it sums up to 1 and has an expected value equal to $W^{\star}$. To do this, we have to find the root of the following equation (see Atwood, 1980; Tripathi et al., 1994):

$$\frac{\mathsf{E}(W)}{1-\mathsf{P}(W=0)}-W^{\star}=0. \tag{14}$$

We solve Equation (14) numerically using **gretl**'s built-in `fzero` function. Note that for edge cases, i.e., when $W^{\star}\approx 0$ or $W^{\star}\approx K$, the solution to Equation (14) may not exist. Examples for $K=7$ are shown in Figures 2a-2d.



(a) (b) (c) (d)

Figure 2: Prior model probabilities with associated prior model size for $K=7$ and $W^{\star}=\frac{K}{2}$, when model does not include a constant term

On the basis of Bayes' formula, the posterior probability of any model M_j can be computed as follows:

$$\Pr(M_j|y)=\frac{\Pr(M_j)p\left(y\mid M_j\right)}{\sum_{r=1}^{2^K}\Pr(M_r)p\left(y\mid M_r\right)} \tag{15}$$

It is easy to see that Equation 15 allows us to calculate the probability for each model M_r using corresponding prior probabilities $P\left(M_r\right)$ and the density of the marginal distribution $p(y\mid M_j)$, here approximated by expressions 3 – 5.

Another useful characteristic popular in the model averaging approach is the Posterior Inclusion Probability (PIP), which is defined as the posterior probability that the independent variable x_i is relevant in explaining the dependent variable (Leamer (1978); Mitchell and Beauchamp (1988)). In our case, the PIP is calculated using Monte Carlo integration, as the sum of the posterior model probabilities for all models that include the specific independent variable (Forte et al., 2018).

Although posterior characteristics are often of interest, predicting future values of the dependent variable can be valuable in many cases. With the **BACE** approach, we can also perform predictive inference across the entire model space. If we have the matrix X_r^* containing explanatory variables for each of the out-of-sample data points, we can use a weighted average of the individual model forecasts to make predictions about unobserved values of the dependent variable.

To determine the significance of determinants and the most likely models across the model space, we utilize the MC^3 sampling algorithm, which was originally developed by Madigan et al. (1995). The MC^3 algorithm is one of the most widely used MCMC algorithms in many statistical packages for model averaging, for example: Lamnisos et al. (2013); Zeugner and Feldkircher (2015); Raftery et al. (2022); Lucchetti and Pedini (2022).

2.2 The limited dependent variable models

In many practical applications, it is common to encounter phenomena with discrete or censored dependent variables. Our **BACE** package includes several models that can handle this issue, including the most popular ones such as logit and probit models, as well as their ordered specifications and multinomial logistic regression. Additionally, the package provides support for Poisson regression, Tobit model, and interval regression[3]. In all of these cases, the dependent variable is either dichotomous (i.e., taking the values zero or one), categorical, a count, or censored. In the remainder of this subsection, we present the standard and popular models for limited dependent variable included in our **BACE** package, which are also discussed in Koop et al. (2007); Hosmer and Lemeshow (2000); Winkelmann (2008); Verbeek (2018) and Cottrell and Lucchetti (2022c).

First, let us analyze the logit and probit models. The logit and probit models are two types of regression models commonly used to predict a binary dependent variable. Logit models are based on the logistic distribution, while probit models are based on the normal distribution. Let us consider a model of the form:

$$z_i = \sum_{j=1}^{k_r} x_{ij}\beta_j + \epsilon_i, \tag{16}$$

and

$$y_i = \begin{cases} 1 & \text{if } z_i > 0 \\ 0 & \text{if } z_i \leq 0, \end{cases}$$

where z_i, $i = 1, \ldots, N$ is an unobserved variable, while y_i is binary, observed data. x_{ij} and β_j are explanatory variables and regression coefficients in model M_r for $j = 1, \ldots, k_r$. If $\epsilon_i \overset{\text{iid}}{\sim} N(0, 1)$, then we have a probit model. Otherwise, we obtain a logit regression.

---

[3] The **BACE** package also includes linear regression models. However, since the dependent variable is not bounded in these models, linear regression is not the main focus of our interest.

In the next step, let us specify logit and probit models, where we consider the choice between L alternatives, numbered from 1 to L:

$$z_i = \sum_{j=1}^{k_r} x_{ij}\beta_j + \epsilon_i, \tag{17}$$

and

$$y_i = \begin{cases} 1 & \text{if } \alpha_0 < z_i \leq \alpha_1 \\ 2 & \text{if } \alpha_1 < z_i \leq \alpha_2 \\ \vdots & \vdots \\ L & \text{if } \alpha_{L-1} < z_i \leq \alpha_L, \end{cases}$$

where unknown α_is are cutpoints with $\alpha_0 = -\infty, \alpha_1 = 0$ and $\alpha_L = \infty$. Assuming that ϵ_i is iid and standard normal, we obtain the ordered probit model, whereas the logistic distribution gives the ordered logit model.

The latest considered model of this kind is multinomial logistic regression. It s an extension of logistic regression modeling for cases where the dependent variable is not binary but can take the values $0, 1, \ldots, p$ and the categories are nominal, meaning they do not have a natural order.

The model can be written in the following form:

$$\Pr(y_i = k \mid x_i) = \frac{\exp(x_i\beta_k)}{1 + \sum_{j=1}^{p} x_i\beta_j}, k > 0, \tag{18}$$

$$\Pr(y_i = 0 \mid x_i) = \frac{1}{1 + \sum_{j=1}^{p} x_i\beta_j}. \tag{19}$$

Poisson regression is a model used to predict count data occurrences. That is, a dependent variable y takes values $k = 0, 1, 2\ldots$ from the Poisson distribution. Under this assumption, the probability of the outcome variable y conditional upon regressors is given by:

$$\Pr(y_i = k \mid x_i) = \frac{\exp\{-\lambda_i\}\lambda_i^k}{k!}, \tag{20}$$

where $\lambda_i = \exp(\sum_{j=1}^{k_r} x_{ij}\beta_j)$ in model M_r.

The Tobit model is a type of regression model used to analyze censored data, which refers to data with a lower or upper bound on the dependent variable. Consider the following Tobit model (type I) with two thresholds (Cottrell and Lucchetti, 2022c):

$$z_i = \sum_{j=1}^{k_r} x_{ij}\beta_j + \epsilon_i, \tag{21}$$

and

$$y_i = \begin{cases} a & \text{for } z_i \leq a \\ z_i & \text{for } a < z_i < b \\ b & \text{for } z_i \geq b, \end{cases} \tag{22}$$

where $\epsilon \sim N(0, \sigma^2)$. Usually the most common specification is $a = 0$ and $b = \infty$.That is, at zero, observations are censored from below, which means that we do not observe any z_i values that are negative (Verbeek, 2018).

Another model that is available in the **BACE** package is the interval regression. It is similar to the standard linear regression, but it allows for the possibility of censoring some data points due to measurement limitations or other factors (Stewart, 1983; Bettin and Lucchetti, 2012; Cottrell and Lucchetti, 2022c). In this case, we consider a data generating process with an unobserved dependent variable z_i (Equation 21) and an observed interval that contains this variable:

$$m_i \leq z_i \leq M_i. \tag{23}$$

The interval can also be left-unbounded, i.e., $m_i = -\infty$ or right-unbounded, when $M_i = \infty$. In the case of $m_i = M_i$, all observations are point observations and the model reduces to a standard regression model, which is also available in the **BACE** package. Please note that due to the complexity of the censoring problem, the interval regression model is only available in script form.

All of the mentioned models are available in the **BACE** package. Through the use of model averaging analysis, it is possible to estimate parameters, identify the key determinants, and make predictions.

2.3 Jointness measures

The main implementations of model averaging are concerned with the selection of variables when model uncertainty is present. Another relevant issue which arises in this framework is to identify whether different sets of two variables x_i and x_j are substitutes, complements or neither over the model space. For that reason, Ley and Steel (2007) and Doppelhofer and Weeks (2009) define ex-post jointness measures of dependence between different sets of explanatory variables. Following Hofmarcher et al. (2018), we can use a general contingency Table 1 and easily calculate a number of jointness measures considered in the literature. In **BACE** we have:

Table 1: General contingency table for two binary variables.

	B	$\overline{B}$	Total
A	a	b	$a+b$
$\overline{A}$	c	d	$c+d$
Total	$a+c$	$b+d$	$a+b+c+d=N$

- The jointness measure J_{DW1}, first proposed in the Doppelhofer and Weeks

(2005):

$$\begin{aligned} J_{DW1}(A,B) &= \ln\left(\frac{\mathsf{P}(AB \mid y)}{\mathsf{P}(A \mid y)\mathsf{P}(B \mid y)}\right) \\ &= \ln\left(\frac{aN}{(a+b)(a+c)}\right). \end{aligned} \tag{24}$$

- An alternative variant J_{DW2} also proposed by Doppelhofer and Weeks (2009):

$$\begin{aligned} J_{DW2}(A,B) &= \ln\left(\frac{\mathsf{P}(AB \mid y)\mathsf{P}(\overline{A}\overline{B} \mid y)}{\mathsf{P}(A\overline{B} \mid y)\mathsf{P}(\overline{A}B \mid y)}\right) \\ &= \ln\left(\frac{ad}{bc}\right). \end{aligned} \tag{25}$$

- The jointness statistics presented in Ley and Steel (2007):

$$\begin{aligned} J_{LS1}(A,B) &= \frac{\mathsf{P}(AB \mid y)}{\mathsf{P}(A \mid y) + \mathsf{P}(B \mid y) - \mathsf{P}(AB \mid y)} \\ &= \frac{a}{a+b+c}. \end{aligned} \tag{26}$$

- An additional variant of an earlier measure:

$$\begin{aligned} J_{LS2}(A,B) &= \frac{\mathsf{P}(AB \mid y)}{\mathsf{P}(A\overline{B} \mid y) + \mathsf{P}(\overline{A}B \mid y)} \\ &= \frac{a}{b+c}. \end{aligned} \tag{27}$$

- The measure proposed in Strachan (2009):

$$\begin{aligned} J_{St}(A,B) &= \mathsf{P}(A \mid y)\mathsf{P}(B \mid y)\ln\left(\frac{\mathsf{P}(AB \mid y)}{\mathsf{P}(A\overline{B} \mid y)\mathsf{P}(\overline{A}B \mid y)}\right) \\ &= \frac{a+b}{N}\frac{a+c}{N}\ln\left(\frac{aN}{bc}\right). \end{aligned} \tag{28}$$

- Yule's Q measure:

$$\begin{aligned} J_{YQ}(A,B) &= \frac{\mathsf{P}(AB)\mathsf{P}(\overline{A}\overline{B}) - \mathsf{P}(A\overline{B})\mathsf{P}(\overline{A}B)}{\mathsf{P}(AB)\mathsf{P}(\overline{A}\overline{B}) + \mathsf{P}(A\overline{B})\mathsf{P}(\overline{A}B)} \\ &= \frac{ad-bc}{ad+bc}. \end{aligned} \tag{29}$$

- The last measure and also the default one, modified Yule's Q:

$$J_{YQ} = \frac{\left(a+\frac{1}{2}\right)\left(d+\frac{1}{2}\right) - \left(b+\frac{1}{2}\right)\left(c+\frac{1}{2}\right)}{\left(a+\frac{1}{2}\right)\left(d+\frac{1}{2}\right) + \left(b+\frac{1}{2}\right)\left(c+\frac{1}{2}\right) - \frac{1}{2}}. \tag{30}$$

Information on the properties of these listed measures and their interpretations can be found in the publication of Hofmarcher et al. (2018).

3 Implementation in gretl

In this section, we will describe the basic idea of the **BACE** package. First, we will characterize our code and explain the use of both the command line and the GUI interfaces. Next, we will provide some examples of how to use the **BACE**. Finally, we will show the outputs that are returned.

3.1 The core of the BACE code

The **BACE** package was written in the `Hansl`[4] programming language and consists of 4 public and 14 private functions[5]. The name of each function in the **BACE** package starts with the prefix `BACE_`. The public functions are: `BACE()`, `BACE_GUI()`, `BACE_Print()` and `BACE_Plot()`.

In order to install **BACE** one should start **gretl** and go to File → Function packages → On server... and select **BACE** from a list. Another possible option is to use either **gretl's** console or script editor and type:

```
pkg install BACE
```

or – assuming installation from local file:

```
pkg install /path/to/BACE.zip --local --quiet
```

where `/path/to/` is an absolute or relative path to the `BACE.zip` archive.

3.1.1 `BACE()`

This is a core function of the package that runs and controls the main loop. As a result, it is a primary interface to the **BACE** package, although it may also be called by `BACE_GUI()`. The `BACE()` function returns a **gretl** `bundle`[6] containing posterior results and all necessary information to print and/or plot results at any time. To print the results, one can call `BACE_Print()`, and to plot graphs, one can call `BACE_Plot()`. The signature of `BACE()` function is as follows:

```
function bundle BACE (list Y "Dependent variable",
 list X_list "List of indenpendent variables (X)",
 bundle Options[null])
```

---

[4] More information about **BACE** can be found in Cottrell and Lucchetti (2022a).

[5] One note should be added here: **BACE** provides analysis for ordered logit/probit models, including prediction, and as such, depends on one external gretl package called `oprobit_predict`.

[6] A **gretl** bundle is a collection of **gretl**'s regular objects, including other bundles. For more information, please refer to Cottrell and Lucchetti (2022c).

The first two options are mandatory, while the last one is an optional set of additional arguments[7]. These arguments are defined with the following default values[8]:

`string Options["model_type"] = "linear"`: the type of model that should be applied to the data; possible options are: 1—linear, 2—logit, 3—probit, 4—poisson, 5–tobit, 6–intreg, 7–mlogit;

`scalar Options["bic_type"] = 2`: the version of Bayesian Information Criterion that should be used; possible values are: 1—bic, 2—hbic, 3—ibic, 4—ebic (see Equations (3-6));

`scalar Options["model_prior"] = 1`: the prior model size distribution to use; possible values are: 0—none, 1—Binomial, 2—Beta-Binomial (see Equations (9) and (11));

`scalar Options["with_const"] = 1`: the inclusion of a constant term in each estimated model; possible values are: 0—Never (without constant), 1—Always (constant is always included), 2—Can be dropped (constant may be dropped);

`scalar Options["avg_model_size"] = avg_model_size_tmp`: the expected prior model size $W^\star$ as defined on page 6, where `scalar avg_model_size_tmp` is calculated according to the following formula:

$$\texttt{avg\_model\_size\_tmp} = \begin{cases} \texttt{K / 2} & \text{if} \quad \texttt{with\_const != 2}, \\ \texttt{(K + 1)/ 2} & \text{elsewhere}, \end{cases} \tag{31}$$

and K stands for the number of independent variables[9];

`scalar Options["data_trans"] = 0`: the transformation of explanatory variables; possible options are: 0—none, 1—centering (i.e., subtracting the mean from each value of a variable), 2—standarizing (centering a variable and additional dividing by its standard deviation);

`scalar Options["do_joint"] = 0`: the type of jointness analysis; possible values are: 0—none, 1—DW1 (Doppelhofer-Weeks Measure 1), 2—DW2 (Doppelhofer-Weeks Measure 2), 3—LS1 (Ley-Steel Measure 1), 4—LS2 (Ley-Steel Measure 2), 5—SM (Strachan Measure), 6—YQ (Yule's Q Measure), 7—YQM (Modified Yule's Q Measure) (see Section 2.3);

`scalar Options["alpha"] = alpha_tmp`: the significance level for the initial model,

---

[7] Note, that in case of `intreg` estimator the Y list must contain two series with *minvar* and *maxvar* (left- and right- bounds respectively).

[8] Some of the above-mentioned parameters are over-loadable, which means that their value may be passed either as an integer or as a string (case-insensitive). These parameters are: `bic_type`, `model_prior`, `with_const`, `do_joint`, `data_trans`, and `model_type`.

[9] If model includes lagged depended variable then `avg_model_size_tmp` is increased by `y_lag/2`.

where `scalar alpha_tmp` takes the following values:

$$\texttt{alpha\_tmp} = \begin{cases} 1.0 & \text{if} \quad \texttt{model\_type = "intreg"}, \\ 0.6 & \text{elsewhere}; \end{cases} \tag{32}$$

`scalar Options["Nrank"] = 5`: the number of top-ranked models to be printed, as described on page 23;

`scalar Options["Npredict"] = 0`: the number of out-of-sample forecasts[10];

`scalar Options["y_lag"] = 0`: defines the lag order of the dependent variable used in dynamic models;

`scalar Options["hyperparameter"] = 0.5`: the hyperparameter for prior-based Bayesian information criterion; for now, it is interpreted as γ for the extended Bayesian information criterion (EBIC) (see Equation (6));

`scalar Options["binary_threshold"] = 0.5`: this hyper-parameter is applicable only in models for binary-dependent variables; it indicates the lowest probability for which $\hat{y}_i$ should be considered as 1;

`scalar Options["low"] = 0`: this option is applicable only for the `tobit` model; it controls the `--llimit` flag of the tobit model, which sets the left bound in censoring;

`scalar Options["high"] = NA`: applicable only for `tobit` model; this option controls tobit's `--rlimit` flag for setting right bound in truncation;

`scalar Options["Nrep"] = Nrep`: the total number of Monte Carlo iterations (model draws) in MC^3 algorithm, where `scalar Nrep` takes the following values:

$$\texttt{Nrep} = \begin{cases} 10^4 & \text{if} \quad K \leq 15, \\ 10^5 & \text{if} \quad K \in [16; 19], \\ 10^6 & \text{if} \quad K \geq 20; \end{cases} \tag{33}$$

`scalar Options["burn"] = 10`: percentage of burn-in draws in MC^3;

`scalar Options["verbosity"] = 2`: verbosity of the output: 1—basic results, 2—full results; this option controls what is printed as output by BACE (see Section 4.1 for details): if set to 2, all possible results are printed, and if set to 1, printing of the initial model and models in ranking is suppressed;

`scalar Options["show_progress"] = do_show_progress`: this controls whether the overall progress is shown, where `scalar do_show_progress` takes the follow-

---

[10] Note, that using this option makes **BACE** to perform an analysis on subsample.

ing values:

$$\texttt{do\_show\_progress} = \begin{cases} 1 & \text{if} \qquad\qquad \texttt{\$sysinfo["gui\_mode"]}, \\ 0 & \text{elsewhere;} \end{cases} \tag{34}$$

`scalar Options["do_plot"] = do_plot_tmp`: this option controls whether to produce graphs as described on page 26, where `scalar do_plot_tmp` takes the following values:

$$\texttt{do\_plot\_tmp} = \begin{cases} 1 & \text{if} \qquad\qquad \texttt{\$sysinfo["gui\_mode"]}, \\ 0 & \text{elsewhere;} \end{cases} \tag{35}$$

`scalar Options["_gui"] = 0`: this parameter is for internal use only and must not be modified by the user under any circumstances.

Most of the optional parameters listed above are of a numerical type and can be grouped into the following categories based on the type of numbers they can take:

1. Integers: `y_lag`, `with_const`, `model_prior`, `Nrank`, `do_joint`, `Npredict`, `Nrep`, `burn`, `verbosity`, `data_trans`, `bic_type`.
2. Booleans: `show_progress`, `do_plot`.
3. Scalars: `avg_model_size`, `alpha`, `binary_threshold`.

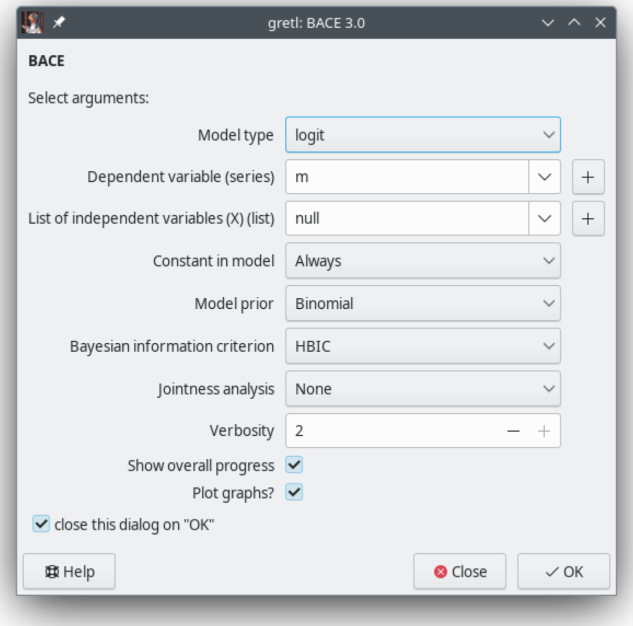
3.1.2 `BACE_GUI()`

Instead of using the script method, another option to utilize the **BACE** package is through its GUI interface. Assuming that the package is already installed, to access the GUI, one should navigate to File → Function packages → On local machine, and select **BACE**. Figure 3 presents **BACE** GUI window with default settings.

The `BACE_GUI()` function is a wrapper for the `BACE()` function, and its signature is as follows:

```
function bundle BACE_GUI (
 int model_type[1:5:1] "Model type" {"logit", "probit",
       "mlogit", "poisson", "tobit", "linear"},
 series Y "Dependent variable",
 list X_list[null] "List of independent variables (X)",
 int with_const[0:2:1] "Constant in model" {"Never",
       "Always", "Can be dropped"},
 int model_prior[0:2:2] "Model prior" {"none",
       "Binomial", "Beta-Binomial"},
 int bic_type[1:4:2] "Bayesian information criterion"
       {"BIC", "HBIC", "IBIC", "EBIC"},
```

```
int do_joint[0:7:0] "Jointness analysis" {"None",
       "Doppelhofer-Weeks Measure (DW1)",
       "Doppelhofer-Weeks Measure (DW2)",
       "Ley-Steel Measure (LS1)", "Ley-Steel
       Measure (LS2)", "Strachan Measure (SM)",
       "Yule's Q Measure (YQ)", "Modified Yule's Q
       Measure (MYQ)"},
int verbosity[1:2:2] "Verbosity",
bool show_progress[1] "Show overall progress",
bool do_plot[1] "Plot graphs?")
```



BACE

Select arguments:

Model type: logit

Dependent variable (series): m

List of independent variables (X) (list): null

Constant in model: Always

Model prior: Binomial

Bayesian information criterion: HBIC

Jointness analysis: None

Verbosity: 2

Show overall progress ☑

Plot graphs? ☑

☑ close this dialog on "OK"

Figure 3: The **BACE** GUI window interface.

Please note that, since `BACE_GUI()` is designed to be a wrapper to `BACE()`, it does not offer an option to set all possible package parameters. This is partially because some of them are mutually exclusive, and partially because we wanted to design a GUI interface that is not too complicated.

3.1.3 `BACE_Print()`

`BACE_Print()` is a function for printing the results of the analysis performed by the **BACE** package, and its signature is as follows:

```
function void BACE_Print (bundle *MODEL)
```

This function takes as an argument a reference to the bundle `MODEL` with **BACE** results and prints information according to the `verbosity` settings. A detailed description of what is printed as output can be found in Section 4.1.

3.1.4 `BACE_Plot()`

`BACE_Plot()` is a function for producing graphs, and its signature is as follows:

```
function void BACE_Plot (const bundle *MODEL)
```

This function takes as an argument a reference to the `MODEL` bundle with **BACE** results and plots graphs according to the `do_plot` setting, which includes the Prior model probabilities, model size distributions (Prior and Posterior), CDF of model size, predictive results (if forecasts are generated), and two heatmaps with variables inclusions and jointness results. In non-GUI mode (when `$sysinfo["gui_mode"]` == 0), plots are generated as scripts for `gnuplot`[11] program and saved on a disk in path specified under `$workdir` entry. Names of generated files have their base set to `strftime($now[1], "%Y%m%d%H%M%S")`. Example graphs are presented on Figures 4b-6b.

4 Usage of the BACE package

The main interface to the **BACE** package is `BACE()` function described in 3.1.1. Calling this function is available from both: script and command line window, just after loading the package via `include` command. As an example we use so-called NELS dataset used in Hill et al. (2017) which concerns longitudinal study on eight-grade students in USA[12]. Table 2 includes basic characteristics of this dataset.

We assume analysis for ordered probit model for `psechoice` with default values of **BACE** parameters. The only non-default setting will be jointness analysis by modified Yule's Q Measure (Equation 30). A very minimal code would be as follows[13]:

```
open nels.gdt --quiet
include BACE.gfn
set seed 1000000
list X = 2..14

```

---

[11] See `http://www.gnuplot.info` for additional information.

[12] This dataset is available at `http://www.principlesofeconometrics.com/poe5/poe5data.html`. It is also available for **gretl** thanks to the work of Lee Adkins: `http://www.learneconometrics.com/gretl/index.html`.

[13] The reported chain (10000 replications with 10% burn-in draws) took ≈ 8 sec. of CPU time on a PC with 4 Intel Core i7-8550U CPU @1.80GHz and 16 GB RAM running under Debian bookworm/sid GNU/Linux. We used **gretl** 2023a compiled by gcc 12.2.0 and linked against openblas 0.3.21. The seed for RNG was set to 1000000.

Table 2: NELS dataset variables definition

$$\text{psechoice} = \begin{cases} \text{1 if first postsecondary education was no college} \\ \text{2 if first postsecondary education was a 2-year college} \\ \text{3 if first postsecondary education was a 4-year college} \end{cases}$$

hscath = 1 if catholic high school graduate

hsrural = 1 if high school rural

grades = average grade in math, english and social studies on 13 point scale with 1 = highest

faminc = gross 1991 family income (in $1000)

famsiz = number of family members

parsome = 1 if most educated parent had some college, but not a 4-year degree

parcoll = 1 if most educated parent graduated from college or had an advanced degree

female = 1 if female

asian = 1 if asian

hispan = 1 if hispan

black = 1 if black

grants = 1 if student had a grant/scholarship or fellowship when attending college

loans = 1 if student received a loan while attending college

Source: `http://www.principlesofeconometrics.com/poe5/data/def/nels.def`

```
res = BACE(psechoice, X, _(model_type="probit",
   do_joint="myq"))
```

The above example consists of 5 lines and is rather self-explained. In the first line we call `open` command to load NELS dataset from `nels.gdt` file. In the second line we load **BACE** package (using `include` command). The third line contains definition of seed for pseudo-random number generator. In the fourth line we define gretl's `list X` which includes independent variables with ID's between 2 and 14. Finally, the fifth line defines Bayesian Averaging of Classical Estimates analysis of student's college type choice in USA with two parameters set explicitly (i.e., values of rest of parameters are set to their defaults as described on page 13). Note, that switching between binary and ordered probit is done by gretl automatically according to type of the depended variable. The **BACE** results will be saved in `res` bundle.

The above-mentioned way of calling `BACE()` function is not the only one possible in **gretl**. Alternatively, we can use an option bundle to define **BACE** parameters[14]:

```
bundle opts
opts["model_type"] = "probit"
```

---

[14] We assume that the first four lines from the initial script remain unchanged in consecutive examples.

```
opts["do_joint"] = "myq"

res = BACE(psechoice, X, opts)
```

On contrary, the corresponding "big call", i.e. setting all possible parameters manually, would be as follows:

```
bundle opts = defbundle("model_prior", 1, \
  "avg_model_size", 6.5, "alpha", 0.6, "Nrank", 5, \
  "do_joint", 7, "Npredict", 0, "Nrep", 10000, \
  "burn", 10, "verbosity", 2, "show_progress", 1, \
  "do_plot", 1, "model_type", 3, "with_const", 0, \
  "bic_type", 2, "data_trans", 0, \
  "hyperparameter", 0.5, "binary_threshold", 0.5, \
  "low", NA, "high", 0)

res = BACE(psechoice, X, opts)
```

or alternatively with the new syntax for creating bundle on-the-fly:

```
res = BACE(psechoice, X, _(model_prior=1, \
  avg_model_size=6.5, alpha=0.6, Nrank=5, \
  do_joint=7, Npredict=0, Nrep=10^4, burn=10, \
  verbosity=2, show_progress=1, do_plot=1,\
  model_type=3, with_const=0, bic_type=2, \
  data_trans=0, hyperparameter=0.5, \
  binary_threshold=0.5, low=NA, high=0))
```

Since some parameters in option bundle are over-loadable (see information in 8), the equivalent call looks like this:

```
res = BACE(psechoice, X, \
  _(model_prior="binomial", avg_model_size=6.5, \
  alpha=0.6, Nrank=5, do_joint="myq", Npredict=0, \
  Nrep=10^4, burn=10, verbosity=2, show_progress=1, \
  do_plot=1, model_type="probit", with_const="never", \
  bic_type="hbic", data_trans="none", \
  hyperparameter=0.5, binary_threshold=0.5, low=NA, \
  high=0))
```

Note that since we are using an ordered probit model here, some of the options mentioned above are redundant and as such are not even considered. Those are:

- `hyperparameter`, used only for EBIC (Equation 6),
- `binary_threshold`, used only for binary `logit` or `probit` models,
- `low` and `high`, used only for `tobit` model.

Results from **BACE** are described in Section 4.1.

Suppose we want to calculate Ley-Steel's LS2 Measure instead of already computed Modified Yule's Q Measure (MYQ). Since the type of jointness measure is controlled by `do_join` field in the result bundle, the only thing we have to do is to set the value of `do_join` to 4 (or "ls2") and produce **BACE** results again. The code should be as follows:

```
res["do_joint"] = 4
BACE_Print(&res)
BACE_Plot(&res)
```

Note, that all above mentioned analysis may be performed via GUI interface. But - since it is only an option method - it does not offer an option to set all possible package parameters. See Section 3.1.2 for details.

4.1 Output

The output from the **BACE** package can be split into two parts: printed text produced by the `BACE_Print()` function, where summary and posterior results are presented, and plotted graphs returned by the `BACE_Plot()` function. Both functions are described in Sections 3.1.3 and 3.1.4. The printed output window consists of several important sections, such as the initial settings and basic summary of random sampling, an initial (starting) model estimates, posterior results—summary of the most common model averaging metrics, jointness measures, and estimates for the top models.

When you run the script with the initial set (a complete script is described on page 18):

```
res = BACE(psechoice, X, _(model_type="probit",
    do_joint="myq"))
```

one of the first pieces of information returned is a short summary of the **BACE** settings, along with basic information about the results of the Monte Carlo experiment. This information includes the number of visited and accepted models. The first part of the printed results is as follows:

```
Model type: ordered probit
Constant type: Never (with cut points)
Prior: Binomial
Prior average model size: 6.500000
Significance level for the initial model: 0.600000
Bayesian information criterion used: HBIC
Data transformation: none
Total number of iterations: 10000
Number of burn-in draws: 1000
Total number of possible models: 8192
```

```
Number of models visited: 114
Number of models accepted: 15
```

The results printed above are fairly self-explanatory, but a few additional comments should be added:

- In case of `tobit` model two additional lines are printed below `Model type`: `left bound` and `right bound` with values of `low` and `high` limits respectively.
- If we use EBIC as a model selection criterion, an additional line will be printed below the `Bayesian information criterion used`, which is the `Hyperparameter for EBIC (gamma)` with the value of the γ parameter used in the simulation.
- If we have binary `logit` or `probit` models, an additional line is printed below the `Data transformation`, i.e. the `Threshold for binary dependent variable`, with the minimum value for which the dependent variable takes a value of 1.
- If we run **BACE** with either ,"random" or "without" a constant term, two additional lines may be printed below the `Total number of iterations`: `Number of rejected draws` and `Effective number of iterations`. This is due to the fact that, in these cases, an empty model does not exist, so if we have such a specification drawn in an MC^3 simulation, it must be rejected (we cannot implicitly exclude empty specifications from the model space in our drawing schema).
- If **BACE** encounters missing values in a dataset, a line `Original number of observations (with missing)` is printed at the bottom of the summary.

If `verbosity` option is set to 2, the results of the initial model estimation are printed. For the NELS example, the initial model looks as follows:

```
The initial model: ordered probit, dependent variable: psechoice

            coefficient   std. error       z        p-value
-----------------------------------------------------------
hscath       0.348616     0.0664749       5.244   1.57e-07   ***
hsrural     -0.187616     0.0352009      -5.330   9.83e-08   ***
grades      -0.229958     0.00825797    -27.85    1.18e-170 ***
faminc       0.00782622   0.000519981    15.05    3.40e-51   ***
famsiz      -0.0510152    0.0120154      -4.246   2.18e-05   ***
parsome      0.323895     0.0401926       8.059   7.72e-16   ***
parcoll      0.684446     0.0487883      14.03    1.04e-44   ***
asian        0.102519     0.0623079       1.645   0.0999     *
hispan       0.153188     0.0545347       2.809   0.0050     ***
```

```
black        0.261885     0.0575335      4.552   5.32e-06  ***
grants       0.792064     0.0403978     19.61    1.36e-85  ***
loans        1.09411      0.0524522     20.86    1.26e-96  ***
cut1        -1.67948      0.0921330    -18.23    3.05e-74  ***
cut2        -0.523737     0.0902467     -5.803   6.50e-09  ***
```

The output contains typical information regarding model estimates, including coefficient estimates, standard errors for the coefficient estimates, z-statistics that test whether the coefficients are equal to zero, and corresponding p-values. In some cases, the results for the initial model may be extended with additional information:

- If case of a `tobit` model, lines indicating the number of left– and/or right–censored observations will be printed.
- If we have an `intreg` model, lines indicating the number of left– and right–unbounded, bounded, and point observations will be printed.
- When it comes to models with a censored dependent variable, the results of the normality test are displayed.
- Note that in case of a multinomial `logit`, the output in starting model is multiplied as many times as the number of cases of dependent variable minus 1.

One of the first pieces of information displayed about posterior results is prior and posterior model size distributions, for NELS example that will be:

```
Prior:
        average model size: 6.500000
        standard deviation: 1.802776
Posterior:
        average model size: 10.548111
        standard deviation: 0.638328
```

Note that if we use EBIC, an additional lines labelled `Effective Prior` are displayed, showing the moments of the effective prior model size distribution calculated as a combination of the `model_prior` specified by the user and the model prior implicitly included in EBIC (Equation 6).

The next step displays the basic characteristics associated with evaluating the parameters and related variables in the model averaging approach.

```
  Posterior moments (unconditional and conditional on inclusion):
                 PIP         Mean      Std.Dev.    Cond.Mean Cond.Std.Dev.
   loans     1.000000    1.089366    0.052477     1.089366     0.052477
  grants     1.000000    0.796474    0.040560     0.796474     0.040560
 parcoll     1.000000    0.689982    0.048644     0.689982     0.048644
```

```
parsome    1.000000    0.326197    0.040197    0.326197    0.040197
 faminc    1.000000    0.007738    0.000521    0.007738    0.000521
 grades    1.000000   -0.229508    0.008326   -0.229508    0.008326
hsrural    1.000000   -0.199767    0.035077   -0.199767    0.035077
 hscath    1.000000    0.341419    0.066410    0.341419    0.066410
  black    0.990222    0.240438    0.062522    0.242812    0.058061
 famsiz    0.980889   -0.047048    0.013558   -0.047965    0.011976
 hispan    0.460667    0.066270    0.080632    0.143856    0.054333
  asian    0.094000    0.008674    0.033129    0.092273    0.062943
 female    0.022333   -0.000169    0.004942   -0.007560    0.032215
```

The main output of **BACE** includes typical model averaging metrics: `PIP` (Posterior Inclusion Probabilities), `Mean` and `Std.Dev.` (the average of the posterior means and standard deviations of parameters), `Cond.Mean` and `Cond.Std.Dev.` (the posterior mean and standard deviation of each coefficient, conditional on the variable being included in the model). And again, in case of a multinomial `logit`, the posterior results are printed as many times as the number of cases of dependent variable minus 1.

In ordered probit and logit models, the values of the cut points are used to measure the effects of the regressors on the dependent variable. In the **BACE** approach, we have their average estimates based on the total model space.

```
                Mean        Std.Dev.
cut1       -1.682247        0.095886
cut2       -0.527395        0.093964
```

Next, we print a CDF-like table showing the probabilities associated with the models.

```
Posterior probability of models:
Best 5 models account for 0.9630 of mass
Best 6 models account for 0.9732 of mass
Best 7 models account for 0.9828 of mass
Best 8 models account for 0.9916 of mass
Best 9 models account for 0.9963 of mass
Best 15 models account for 1.0000 of mass
Best 14 models account for 1.0000 of mass
```

We also print the rankings of the models, along with the Pearson coefficient calculated for the list based on both the analytical and numerical probabilities of the models. In the case of NELS, this would be:

```
Best models ranking:
Model 1:        0.459889
Model 2:        0.398444
Model 3:        0.046111
```

```
Model 4:        0.045444
Model 5:        0.013111
Total probability of the models in ranking (numerical): 0.963000

Correlation coefficient between the analytical
and numerical probabilities of the above models: 0.917611
```

We return the importance of the "best models" in relation to the entire model space by ranking them based on MCMC sampling. Additionally, the program provides information on Pearson's correlation between the analytical and MC^3 posterior probabilities of the models. High correlation indicates the convergence of the Monte Carlo chain (Koop, 2003; Fernández et al., 2001). The "exact" posterior model probabilities are calculated from approximations 3–6.

```
Contingency table (actual in rows, predicted in columns):
             1            2            3


1           876          404          115
2           355          660          786
3           123          534         2796

Number of cases 'correctly predicted' = 4332 (accuracy 65.2%)
```

For a logit and probit models, we also obtain the so-called contingency matrix, or more specifically, its variant – the confusion matrix. As we know, the confusion matrix is a tool for evaluating the performance of a classification model. In this case, it offers a concise summary, enabling us to evaluate the accuracy of our model's prediction of `psechoice`. We collect individual results and calculate all combinations of predicted and actual values based on the total model space.

The joint probability and jointness analysis for the current example should appear as follows:

```
Posterior joint probability of variables:
             hscath        hsrural         grades         faminc
hscath     0.000000       1.000000       1.000000       1.000000      ...
hsrural    1.000000       0.000000       1.000000       1.000000      ...
grades     1.000000       1.000000       0.000000       1.000000      ...
faminc     1.000000       1.000000       1.000000       0.000000      ...
famsiz     0.980889       0.980889       0.980889       0.980889      ...
parsome    1.000000       1.000000       1.000000       1.000000      ...
parcoll    1.000000       1.000000       1.000000       1.000000      ...
female     0.022333       0.022333       0.022333       0.022333      ...
asian      0.094000       0.094000       0.094000       0.094000      ...
hispan     0.460667       0.460667       0.460667       0.460667      ...
black      0.990222       0.990222       0.990222       0.990222      ...
```

```
grants     1.000000      1.000000      1.000000      1.000000     ...
loans      1.000000      1.000000      1.000000      1.000000     ...

Jointness statistics (Modified Yule’s Q Measure (MYQ)):
             hscath       hsrural        grades        faminc     ...
hscath          nan      1.000000      1.000000      1.000000     ...
hsrural    1.000000           nan      1.000000      1.000000     ...
grades     1.000000      1.000000           nan      1.000000     ...
faminc     1.000000      1.000000      1.000000           nan     ...
famsiz     0.961778      0.961778      0.961778      0.961778     ...
parsome    1.000000      1.000000      1.000000      1.000000     ...
parcoll    1.000000      1.000000      1.000000      1.000000     ...
female    -0.955333     -0.955333     -0.955333     -0.955333     ...
asian     -0.812000     -0.812000     -0.812000     -0.812000     ...
hispan    -0.078667     -0.078667     -0.078667     -0.078667     ...
black      0.980444      0.980444      0.980444      0.980444     ...
grants     1.000000      1.000000      1.000000      1.000000     ...
loans      1.000000      1.000000      1.000000      1.000000     ...
```

In the binary dependent variable models, by default, we use a threshold of 0.5. It is also worth noting that this threshold value (*binary_threshold* = 0.5) can be adjusted in **BACE** package to achieve a desired recall and precision in the model setup.

In the final step, the output provides maximum likelihood estimates (MLE) for the top estimated models. It provides inferences for the model coefficients in the columns labeled `coefficient`, `std. error`, `z` statistics, and `p-value`. Additionally, it provides the type of information criterion used (in this case, `HBIC`) and the explanatory power (represented by the `posterior probability`). Note that in case of a multinomial `logit`, the output of model estimates is multiplied as many times as the number of cases of dependent variable minus 1.

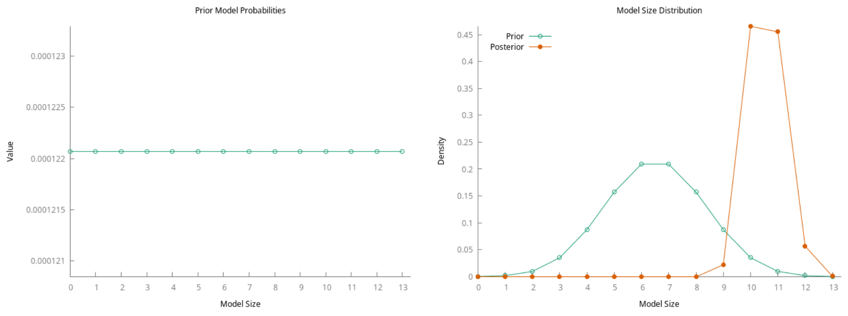
```
Best specifications (MLE estimates for ordered probit model):
----------------------------------

Model 1: HBIC: 9951.131386, posterior probability: 0.459889
         coefficient    std. error       z       p-value
----------------------------------------------------------
hscath      0.339743     0.0663548       5.120   3.05e-07  ***
hsrural    -0.204022     0.0346748      -5.884   4.01e-09  ***
grades     -0.228665     0.00819836    -27.89    3.38e-171 ***
faminc      0.00769592   0.000517451    14.87    4.95e-50  ***
famsiz     -0.0469671    0.0119120      -3.943   8.05e-05  ***
parsome     0.327804     0.0401469       8.165   3.21e-16  ***
parcoll     0.689699     0.0485704      14.20    9.16e-46  ***
```

```
black        0.232157     0.0567165      4.093   4.25e-05  ***
grants       0.799516     0.0403133     19.83    1.56e-87  ***
loans        1.08691      0.0523804     20.75    1.22e-95  ***
cut1        -1.68606      0.0919803    -18.33    4.71e-75  ***
cut2        -0.531627     0.0900860     -5.901   3.61e-09  ***
```

The structure of the results shown may vary slightly depending on the type of model and the settings of the parameters in **BACE**.

As stated at the beginning of this section, the **BACE** package not only provides a printed summary of the results, but also includes graphs that serve as supplementary visual summaries of the results obtained. As the first step, we return the graph of the prior model probabilities i.e. probabilities assigned to each of the candidate models before the data is observed and the model size – prior and posterior number of independent variables in each candidate model being considered. Figure 4 shows prior model probabilities and model size distributions for NELS example.



(a) (b)

Figure 4: Model probabilities and model size distribution for NELS example.

In this case, we employed a Binomial distribution (`"model_prior"=1`) with a default prior average model size ($W^\star = \frac{13}{2} = 6.5$), which, as noted in Section 2.1, leads to a Uniform distribution for all combinations of explanatory variables (graph on the left panel (4a)). It is easy to see that the assumption of a Uniform distribution focuses on combinations consisting of 6 or 7 explanatory variables as being the most likely a priori (panel (4b)). The analysis also shows that, under these assumptions, the data indicate that the most likely models consist of 10 and, further down, 11 independent variables, additionally, combinations consisting of 8 or less elements are highly improbable.

The Figure 5 shows the Cumulative Model Probabilities (CMP) for a NELS example, where the output represents the accumulated probability of all models up to a certain point in the ranking. It is evident that the top 5 ranked models account for over 90% of the total probability in the model space.

The final two figures depict a heatmap of jointness analysis and highlight the explanatory variables identified in the top 5 most probable models. In the current example, the jointness analysis employs the default measure, modified Yule's Q

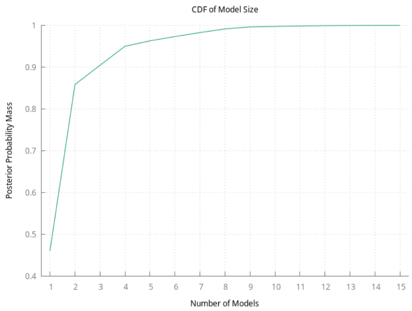
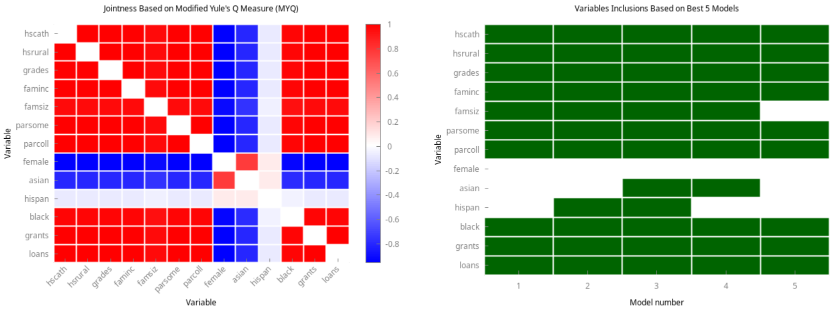


Figure 5: Cumulative Model Probabilities.



(a) (b)

Figure 6: Jointness analysis and variables inclusions in the top models.

(Equation 30). It is easily observed that there is a high complementarity between the most relevant variables. On the other hand, two variables (`female` and `asian`) are strongly substitutable for the others and rarely appear in the most likely models.

5 Empirical illustration and speed of calculation

In this section, we will demonstrate the **BACE** package through two popular empirical applications: logistic regression and Poisson regression. To compare the accuracy and computational speed of the **BACE** results, we consider open and well-known datasets. Additionally, as a benchmark, we investigate the results of two model averaging packages, where some specifications with a limited dependent variable are considered: **ParMA** by Lucchetti and Pedini (2022), **BMA** by Raftery et al. (2022) and **mlogitBMA** by Sevcikova and Raftery (2022).

5.1 Logistic regression example

For the first case, we consider logistic regression for binary data. In this example, we use the *bithwt* dataset presented in Hosmer and Lemeshow (2000), which consists of 189 births with eight risk factors associated with a dummy variable *low* for low infant birth weight. We consider the following risk determinants: *age* – mother's age, *lwt* – mother's weight, *race* – mother's race, *smoke* – smoking status, *ptl* – number of previous premature labours, *ht* – history of hypertension, *ui* – presence of uterine irritability, *ftv* – number of physician visits[15].

Suppose that we want to perform logit model analysis using **BACE** for *bithwt* dataset. The **gretl** script is as follows[16]:

```
set verbose off
set seed 1000000
include BACE.gfn
open birthwt.gdt --quiet

# We specify list of explanatory variables
list X = age lwt Drace_2 Drace_3 smoke ptl ht ui ftv
```

We convert categorical variable *race* into dummy variables and drop *Drace_1* i.e. $race = 1$ due to the collinearity problems. That means we have nine determinants, which leads to $2^9 = 512$ possible model combinations. In the final step, we perform model sampling via the main function **BACE** with `hbic` criterion and without any explicit assumptions on the model prior:

```
BACE(low, X, _(model_type="logit", \
do_plot=0, model_prior="none", bic_type="hbic"))
```

The **BACE** gives a result as below.

```
Posterior moments (unconditional and conditional on inclusion):
              PIP         Mean      Std.Dev.    Cond.Mean Cond.Std.Dev.
const      1.000000     0.449658     1.327281     0.449658     1.327281
ht         0.842333     1.496410     0.921035     1.776505     0.713791
lwt        0.809222    -0.013590     0.009192    -0.016794     0.007114
smoke      0.632556     0.517846     0.513218     0.818657     0.412473
Drace_2    0.573889     0.612028     0.664410     1.066458     0.533460
ptl        0.545889     0.359846     0.415329     0.659193     0.344482
ui         0.485111     0.412825     0.529460     0.850990     0.452758
Drace_3    0.442667     0.345444     0.497936     0.780370     0.469791
age        0.232111    -0.008951     0.023739    -0.038562     0.035862
ftv        0.151333    -0.004575     0.067230    -0.030231     0.170562
```

Based on the presented results, the considered determinants can be divided into three groups: high-probability factors (*ht*, *lwt*) with $\text{PIP} \geq 2/3$, medium-probability factors (*smoke*, *Drace_2*, *ptl*, *ui*, *Drace_3*) with $1/3 \leq \text{PIP} < 2/3$, and the remaining variables (*age*, *ftv*) with low probability ($\text{PIP} < 1/3$).

---

[15] This dataset is available on the R data archive, `https://rdrr.io/cran/MASS/man/birthwt.html`.

[16] When typing `open` without full path, **gretl** searches the file in interest in some relevant paths starting with what is defined under `$workdir`. See Cottrell and Lucchetti (2022b) for details.

The contingency table below indicates an estimation accuracy of 74.1%.

```
Contingency table (actual in rows, predicted in columns):
             0            1


0          126            4
1           45           14

Number of cases 'correctly predicted' = 140 (accuracy 74.1%)
```

In the last step, we compare the results obtained from **BACE** with those obtained from the other two packages that also provide estimation of the logistic regression model. The results are presented in Table 3, which reports the PIPs of the regressors calculated in **BACE**, **ParMA**, and **BMA** packages[17].

	PIP		
Variable	**BACE**	**ParMA**	**BMA**
const	1.000	1.000	1.000
ht	0.842	0.746	0.675
lwt	0.809	0.789	0.694
smoke	0.633	0.721	0.369
Drace_2	0.574	0.569	0.245
ptl	0.546	0.555	0.423
ui	0.485	0.496	0.296
Drace_3	0.443	0.580	0.167
age	0.232	0.242	0.089
ftv	0.151	0.153	0.014

Table 3: Comparison of Posterior Inclusion Probabilities (PIPs) among **BACE**, **ParMA**, and **BMA** packages for the logit regression model.

Based on the presented results, we can conclude that the top three most probable variables are the same as those selected by **ParMA** which are *ht*, *lwt*, and *smoke*. The results obtained from **BACE** and **ParMA** are strongly similar despite being based on different statistical frameworks and model assumptions. The situation is slightly different in the **BMA** case, where the three most likely variables are *ht*, *lwt*, and *ptl*. In this case, the results are more conservative, producing significantly lower PIP values for all explanatory variables.

---

[17] We specify the following entries in the **ParMA** function: `strings vname = varnames(X)`, `string modeltype = "logit"`, `bundle param = null`, `param.mpi = 4`, `param.seed = 1000000`, `param.center = 0`, `n_iter = 9000`, `burn_in = 1000`. In case of the **BMA** we have: `set.seed(1000000)`, `strict=FALSE`, `OR=20`, `glm.family="binomial"`, `factor.type=FALSE`, glm.family="binomial", factor.type=FALSE.

5.2 Poisson regression example

In the second example, we consider the Poisson regression for the *rac3d* dataset, originally used in Cameron and Trivedi (1986, 2013). We use the Poisson model to predict the number of doctor visits (*DVISITS*) based on selected explanatory variables, which include: *SEX* (1 if female, 0 otherwise), *AGE* (in years divided by 100), *AGESQ* (*AGE* squared), *INCOME* (annual, in tens of thousands of dollars), *LEVYPLUS* (1 if private insurance, 0 otherwise), *FREEPOOR* (1 if free government insurance due to low income, 0 otherwise), *FREEREPA* (1 if free government insurance due to old-age, etc., 0 otherwise), *ILLNESS* (number of illnesses), *ACTDAYS* (number of days of reduced activity), *HSCORE* (health questionnaire score, high score means bad health), *CHCOND1* (1 if chronic condition(s) but not limited in activity, 0 otherwise), *CHCOND2* (1 if chronic condition(s) and limited in activity, 0 otherwise). The *rac3d* dataset used in this example consists of $n = 5190$ observations and 12 explanatory variables, resulting in $2^{12} = 4096$ possible model combinations[18]. A similar example was previously considered by Lucchetti and Pedini (2022) using the **ParMA** package.

Here is the minimal code for performing a Poisson analysis using the **BACE** package:

```
set verbose off
set seed 1000000
include BACE.gfn
open rac3d.gdt --quiet

list X = SEX AGE AGESQ INCOME LEVYPLUS FREEPOOR \
        FREEREPA ILLNESS ACTDAYS HSCORE CHCOND1 CHCOND2

BACE(DVISITS, X, _(model_type="poisson", \
        do_plot=0, model_prior="betabinomial", \
        bic_type="hbic"))
```

The main output displays variable names and statistics.

```
 ----------------------------------

Posterior moments (unconditional and conditional on inclusion):
               PIP          Mean      Std.Dev.    Cond.Mean Cond.Std.Dev.
const     1.000000     -2.136763     0.130805     -2.136763     0.130805
ACTDAYS   1.000000      0.128816     0.005349      0.128816     0.005349
ILLNESS   1.000000      0.201279     0.018797      0.201279     0.018797
SEX       0.931222      0.183421     0.073598      0.196968     0.056111
HSCORE    0.818333      0.026606     0.015462      0.032513     0.010005
```

---

[18] The dataset and description are available on the **gretl** repository. To load it, go to File > Open data > Sample file and choose "Gretl" tab.

```
AGE       0.631667   0.349814   0.371404   0.553795   0.324672
FREEPOOR  0.614222  -0.284506   0.265396  -0.463198   0.178618
AGESQ     0.355222   0.142069   0.382779   0.399943   0.556183
INCOME    0.302000  -0.056792   0.099414  -0.188052   0.089677
LEVYPLUS  0.126000   0.012773   0.041862   0.101373   0.070188
CHCOND2   0.105444   0.012172   0.045303   0.115432   0.086857
CHCOND1   0.075000   0.006744   0.031455   0.089922   0.075582
FREEREPA  0.064000   0.005446   0.036704   0.085099   0.119463
```

According to the findings, the determinants of the number of doctor visits can be classified into: high-probable factors (*ACTDAYS*, *ILLNESS*, *SEX*, *HSCORE*), medium-probable (*AGE*, *FREEPOOR*, *AGESQ*) and the others (with low-probability). The second column, `Mean`, and the third column, `Std.Dev.`, display the estimates of the Poisson regression coefficients averaged over all models. As we can see, all coefficients of the most likely variables have a positive sign and relatively small error, which indicates that their increase is significantly related to an increase in the number of doctor visits.

Table 4 displays the Posterior Inclusion Probabilities for the Poisson model estimated by **BACE**, **ParMA**, and **BMA** packages[19].

The results from **BACE** in Table 4 align closely with those of **ParMA**. In all cases, the PIPs are reasonably close, and the same variables have been identified as being relevant. The **BACE** results also appear to be in line with the others. The ranking of the most likely variables in **BMA** is almost identical to the previous two cases. The only major difference is the stronger penalization of less significant variables.

5.3 Speed of calculation

In this section, we will briefly present the calculation speed of the **BACE** package and compare it with the computational speed of two other packages, namely **ParMA** and **BMA**[20], as all three packages support some models for a limited dependent variable. We have implemented our experiment using the **ParMA** package with parallelization across 4 threads, as we believe that this scenario can be implemented on an average personal computer.

Table 5 provides information on the calculation speed of models available in the **BACE** package. As we can see, the **BACE** package allows for the estimation of various models, such as logit and probit regression, ordered logit and probit models, Poisson regression, Tobit model (Type I), and interval regression. Furthermore, we also provide information about the computation speed for the standard (unrestricted) regression model. This study uses well-known and popular datasets in the

---

[19] The following settings are used in the **ParMA** package: `strings vname = varnames(X)`, `string modeltype = "poisson"`, `bundle param = null`, `param.mpi = 4`, `param.seed = 1000000`, `param.center = 0`, `n_iter = 9000`, `burn_in = 1000`. In turn, for the **BMA** we have: `set.seed(1000000)`, `strict=FALSE`, `OR=20`, `glm.family="poisson"`, `factor.type=FALSE`.

[20] The **mlogitBMA** package was used for multinomial logit regression.

	PIP		
Variable	**BACE**	**ParMA**	**BMA**
const	1.000	1.000	1.000
ILLNESS	1.000	1.000	1.000
ACTDAYS	1.000	1.000	1.000
SEX	0.931	0.912	1.000
HSCORE	0.818	0.763	0.723
AGE	0.632	0.688	0.642
FREEPOOR	0.614	0.616	0.350
AGESQ	0.355	0.316	0.358
INCOME	0.302	0.338	0.020
LEVYPLUS	0.126	0.078	<0.01
CHCOND2	0.105	0.042	<0.01
CHCOND1	0.075	0.054	<0.01
FREEREPA	0.064	0.030	<0.01

Table 4: Comparison of Posterior Inclusion Probabilities (PIPs) among **BACE**, **ParMA**, and **BMA** packages for the Poisson regression model.

literature, some of which have already been used in previous sections of this paper. The second column of Table 5 lists the names of the datasets underlying the calculations. These are: *bithwt* presented in Hosmer and Lemeshow (2000)[21], *nels* used in Hill et al. (2017)[22], *rac3d* originally used in Cameron and Trivedi (1986, 2013) and available directly in **gretl** repository, *tobit* example[23], *wtp* – interval regression example of "willingness to pay" in Verbeek (2018), also available in **gretl** repository and *FLS* well–known dataset for model averaging used by Fernández et al. (2001) for standard regression model[24]. The datasets listed differ in their number of observations, explanatory variables, and potential models. As we can see, the FLS data case is the most demanding with over 2 trillion potential combinations, so we can expect significant differences in the speed of computation among the compared packages. According to the data presented in Table 5, it can be concluded that the **BACE** package demonstrated the shortest computation time among the alternatives. The biggest differences in the run time are seen for *FLS* data, where, due to the number of explanatory variables, we consider a very large total model space.

---

[21] MASS: Birth Weights of Babies, `https://rdrr.io/cran/MASS/man/birthwt.html`.

[22] Data Sets for Econometrics, Fifth Edition, `http://www.principlesofeconometrics.com/poe5/poe5data.html`.

[23] Tobit Models in R, `https://stats.oarc.ucla.edu/r/dae/tobit-models/`.

[24] Journal of Applied Econometrics Data Archive, `http://qed.econ.queensu.ca/jae/2001-v16.5/fernandez-ley-steel/`.

Run time	Dataset	N	2^K	**BACE**	**ParMA**	**BMA**
Logit	*bithwt*	189	512	0.17	2.04	0.42
Ordered logit	*nels*	6649	8192	5.13	—	—
Probit	*bithwt*	189	512	0.22	2.47	2.56
Ordered probit	*nels*	6649	8192	5.59	—	—
Multinomial logit	*hsbdemo*	200	1024	0.35	—	9.24
Poisson	*rac3d*	5190	4096	4.14	8.88	4.30
Tobit (Type I)	*tobit*	200	16	0.07	—	—
Interval regression	*wtp*	312	32	0.12	—	—
Linear regression	*FLS*	72	2,199,023,255,552	45.38	14041.1	11.18

Table 5: Run time in seconds of **BACE**, **ParMA** and **BMA** packages for selected datasets, N – the number of observations, K – the total number of explanatory variables in estimated model, and 2^K – total model space.

6 Conclusions

In the article presented here, we propose the **BACE** package, whose main purpose is to provide a model-building strategy for models with a limited dependent variable. We use the **BACE** technique to obtain the average parameter estimates and the significance of determinants in a regression analysis, considering estimates based on Maximum Likelihood Estimation (MLE) across the entire model space.

Overall, we have provided solutions for nine different models, which include logit and probit, ordered logit and probit, multinomial logit, Poisson regression, Tobit, interval regression, and additionally linear regression. Our package utilizes readily available open-source software **gretl**, which offers users the flexibility to perform computations using scripted commands or through a simple and convenient GUI.

The examples presented for the applications of logit and Poisson models confirm the accuracy of the **BACE** results and their consistency with the **ParMA** package. Furthermore, runtime tests have shown that **BACE** produces results in the shortest amount of time, even when compared to the **ParMA** package, which used parallelization across 4 threads in this simulation.

Although the **BACE** package is already fast, consistent, and offers a variety of model types, there is still potential for further development and improvement to expand its functionality. One promising direction for enhancing the package would be to implement two-equation models, such as Heckman-type selection and Zero-Inflated Poisson regression. Both models require a separate drawing schema for each specific equation in the sampling algorithm, which means that they require a different solution approach than the models already implemented in the current version of the package.

References

Albis MLF, Mapa DS. 2014. Bayesian Averaging of Classical Estimates in Asymmetric Vector Autoregressive (AVAR) Models. MPRA Paper 55902, University Library of Munich, Germany.
URL https://ideas.repec.org/p/pra/mprapa/55902.html

Amini S, Parmeter C. 2011. Bayesian Model Averaging in `R`. *Journal of Economic and Social Measurement* **36**: 253–287.

Atwood CL. 1980. Estimators for the truncated beta-binomial distribution .
URL https://www.osti.gov/biblio/5359803

Bayarri MJ, Berger JO, Jang W, Ray S, Pericchi LR, Visser I. 2019. Prior-based Bayesian information criterion. *Statistical Theory and Related Fields* **3**: 2–13.
URL https://doi.org/10.1080/24754269.2019.1582126

Bettin G, Lucchetti R. 2012. Interval regression models with endogenous explanatory variables. *Empirical Economics* **43**: 475–498.
URL https://doi.org/10.1007/s00181-011-0493-9

Białowolski P, Kuszewski T, Witkowski B. 2014. Bayesian averaging of classical estimates in forecasting macroeconomic indicators with application of business survey data. *Empirica* **41**: 53–68.
URL https://ideas.repec.org/a/kap/empiri/v41y2014i1p53-68.html

Bollen KA, Harden JJ, Ray S, Zavisca J. 2014. BIC and Alternative Bayesian Information Criteria in the Selection of Structural Equation Models. *Structural Equation Modeling: A Multidisciplinary Journal* **21**: 1–19. PMID: 31360054.
URL https://doi.org/10.1080/10705511.2014.856691

Bollen KA, Ray S, Zavisca J, Harden JJ. 2012. A Comparison of Bayes Factor Approximation Methods Including Two New Methods. *Sociological Methods & Research* **41**: 294–324.
URL https://doi.org/10.1177/0049124112452393

Błażejowski M, Kufel P, Kwiatkowski J. 2020. Model simplification and variable selection: A replication of the UK inflation model by Hendry (2001). *Journal of Applied Econometrics* **35**: 645–652.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/jae.2760

Błażejowski M, Kwiatkowski J. 2015. Bayesian Model Averaging and Jointness Measures for gretl. *Journal of Statistical Software* **68**: 1–24.

Cameron AC, Trivedi PK. 1986. Econometric models based on count data. Comparisons and applications of some estimators and tests. *Journal of Applied Econometrics* **1**: 29–53.
URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/jae.3950010104`

Cameron AC, Trivedi PK. 2013. *Regression Analysis of Count Data*. Econometric Society Monographs. Cambridge University Press, 2 edition.

Chen J, Chen Z. 2008. Extended Bayesian information criteria for model selection with large model spaces. *Biometrika* **95**: 759–771. ISSN 0006-3444.
URL `https://doi.org/10.1093/biomet/asn034`

Clyde M, Littman M, Ghosh J. 2012. ***BAS**: Bayesian Model Averaging Using Bayesian Adaptive Sampling*. `R` package version 1.0.
URL `http://CRAN.R-project.org/package=BAS`

Cottrell A, Lucchetti R. 2022a. *A Hansl Primer*.
URL `https://netcologne.dl.sourceforge.net/project/gretl/manual/hansl-primer.pdf`

Cottrell A, Lucchetti R. 2022b. *Gretl Command Reference*.
URL `https://netcologne.dl.sourceforge.net/project/gretl/manual/hansl-primer.pdf`

Cottrell A, Lucchetti R. 2022c. ***gretl** User's Guide – Gnu Regression, Econometrics and Time-Series Library*.
URL `https://master.dl.sourceforge.net/project/gretl/manual/gretl-guide.pdf`

Doppelhofer G, Weeks M. 2005. Jointness of Growth Determinants. Cambridge Working Papers in Economics 0542, Faculty of Economics, University of Cambridge.

Doppelhofer G, Weeks M. 2009. Jointness of Growth Determinants. *Journal of Applied Econometrics* **24**: 209–244.

Fernández C, Ley E, Steel M. 2001. Model Uncertainty in Cross-Country Growth Regressions. *Journal of Applied Econometrics* **16**: 563–576.

Forsyth D. 2018. *Probability and Statistics for Computer Science*. Springer Cham. ISBN 978-3-319-64409-7.

Forte A, Garcia-Donato G, Steel M. 2018. Methods and Tools for Bayesian Variable Selection and Model Averaging in Normal Linear Regression. *International Statistical Review* **86**: 237–258.
URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/insr.12249`

Gelman A, Carlin J, Stern H, Dunson D, Vehtari A, Rubin D. 2013. *Bayesian Data Analysis, Third Edition*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis. ISBN 9781439840955.

Greene W. 1999. *Econometric Analysis*. Upper Saddle River: Prentice Hall, 4th edition.

Haughton DMA. 1988. On the Choice of a Model to Fit Data from an Exponential Family. *The Annals of Statistics* **16**: 342 – 355.
URL `https://doi.org/10.1214/aos/1176350709`

Hill R, Griffiths W, Lim G. 2017. *Principles of Econometrics*. Wiley, 5 edition. ISBN 9781119320951.

Hofmarcher P, Crespo Cuaresma J, Grün B, Humer S, Moser M. 2018. Bivariate jointness measures in Bayesian Model Averaging: Solving the conundrum. *Journal of Macroeconomics* **57**: 150–165. ISSN 0164-0704.
URL `https://www.sciencedirect.com/science/article/pii/S0164070418300314`

Hosmer DW, Lemeshow S. 2000. *Applied logistic regression (Wiley Series in probability and statistics)*. Wiley-Interscience Publication, 2 edition. ISBN 0471356328.

Kass RE, Raftery AE. 1995. Bayes Factors. *Journal of the American Statistical Association* **90**: 773–795. ISSN 0162-1459.

Koop G. 2003. *Bayesian Econometrics*. New Jersey: John Wiley & Sons.

Koop G, Poirier D, Tobias J. 2007. *Bayesian Econometric Methods*. Econometric Exercises. Cambridge University Press. ISBN 9780521671736.

Krolzig HM, Hendry DF. 2001. Computer automation of general-to-specific model selection procedures. *Journal of Economic Dynamics and Control* **25**: 831–866. ISSN 0165-1889.

Lamnisos D, Griffin JE, Steel MFJ. 2013. Adaptive Monte Carlo for Bayesian Variable Selection in Regression Models. *Journal of Computational and Graphical Statistics* **22**: 729–748. ISSN 10618600.
URL `http://www.jstor.org/stable/43304857`

Leamer E. 1978. *Specification Searches*. New Jersey, USA: John Wiley & Sons. ISBN 0471015202.

Ley E, Steel M. 2007. Jointness in Bayesian Variable Selection with Applications to Growth Regression. *Journal of Macroeconomics* **29**: 476–493.

Ley E, Steel MF. 2009. On the effect of prior assumptions in Bayesian model averaging with applications to growth regression. *Journal of Applied Econometrics* **24**:

651–674.
URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/jae.1057`

Lucchetti RJ, Pedini L. 2022. ParMA: Parallelized Bayesian Model Averaging for Generalized Linear Models. *Journal of Statistical Software* **104**: 1–39.
URL `https://www.jstatsoft.org/index.php/jss/article/view/v104i02`

Madigan D, York J, Allard D. 1995. Bayesian Graphical Models for Discrete Data. *International Statistical Review* **63**: 215–232.

Mitchell TJ, Beauchamp JJ. 1988. Bayesian Variable Selection in Linear Regression. *Journal of the American Statistical Association* **83**: 1023–1032.
URL `https://www.tandfonline.com/doi/abs/10.1080/01621459.1988.10478694`

Moral-Benito E. 2012. Determinants of Economic Growth: A Bayesian Panel Data Approach. *The Review of Economics and Statistics* **94**: 566–579.
URL `https://ideas.repec.org/a/tpr/restat/v94y2012i2p566-579.html`

Newton MA, Raftery AE. 1994. Approximate Bayesian Inference with the Weighted Likelihood Bootstrap. *Journal of the Royal Statistical Society. Series B (Methodological)* **56**: 3–48. ISSN 00359246.
URL `http://www.jstor.org/stable/2346025`

Pajor A. 2017. Estimating the Marginal Likelihood Using the Arithmetic Mean Identity. *Bayesian Analysis* **12**: 261 – 287.
URL `https://doi.org/10.1214/16-BA1001`

Peterson RA, Cavanaugh JE. 2019. Discussion of prior-based Bayesian information criterion (PBIC) by M.J. Bayarria, James O. Berger, Woncheol Jang, Surajit Ray, Luis R. Pericchi, and Ingmar Visser. *Statistical Theory and Related Fields* **3**: 32–34.
URL `https://doi.org/10.1080/24754269.2019.1611145`

Raftery A, Hoeting J, Volinsky C, Painter I, Yeung K. 2022. *Bayesian Model Averaging – Package BMA*.
URL `http://CRAN.R-project.org/package=BMA`

Raftery AE. 1995. Bayesian Model Selection in Social Research. *Sociological Methodology* **25**: 111–163. ISSN 00811750, 14679531.
URL `http://www.jstor.org/stable/271063`

Raftery AE. 1996. Approximate Bayes factors and accounting for model uncertainty in generalised linear models. *Biometrika* **83**: 251–266. ISSN 0006-3444.
URL `https://doi.org/10.1093/biomet/83.2.251`

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**: 813–835.
URL `https://ideas.repec.org/a/aea/aecrev/v94y2004i4p813-835.html`

Schwarz G. 1978. Estimating the Dimension of a Model. *The Annals of Statistics* **6**: 461–464. ISSN 00905364.
URL `http://www.jstor.org/stable/2958889`

Sevcikova H, Raftery A. 2022. *mlogitBMA: Bayesian Model Averaging for Multinomial Logit Models*.
URL `https://cran.r-project.org/package=mlogitBMA`

Steel M. 2011. Bayesian Model Averaging and Forecasting. *Bulletin of E.U. and U.S. Inflation and Macroeconomic Analysis* **200**: 30–41.

Steel MFJ. 2020. Model Averaging and Its Use in Economics. *Journal of Economic Literature* **58**: 644–719.
URL `https://www.aeaweb.org/articles?id=10.1257/jel.20191385`

Stewart MB. 1983. On Least Squares Estimation when the Dependent Variable is Grouped. *The Review of Economic Studies* **50**: 737–753. ISSN 00346527, 1467937X.
URL `http://www.jstor.org/stable/2297773`

Strachan RW. 2009. Comment on 'Jointness of growth determinants' by Gernot Doppelhofer and Melvyn Weeks. *Journal of Applied Econometrics* **24**: 245–247.
URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/jae.1050`

Tripathi RC, Gupta RC, Gurland J. 1994. Estimation of parameters in the beta binomial model. *Annals of the Institute of Statistical Mathematics* **46**: 317–331. ISSN 0020-3157.
URL `http://link.springer.com/10.1007/BF01720588`

van Dijk D. 2004. Forecasting US Inflation Using Model Averaging. Econometric Society 2004 Australasian Meetings 143, Econometric Society.
URL `https://ideas.repec.org/p/ecm/ausm04/143.html`

Verbeek M. 2018. *A Guide to Modern Econometrics*. Wiley. ISBN 9781119472117.

Wakefield J. 2013. *Bayesian and Frequentist Regression Methods*. Springer Series in Statistics. Springer New York. ISBN 9781441909251.

Winkelmann R. 2008. *Econometric Analysis of Count Data*. Springer Berlin, Heidelberg. ISBN 978-3-540-77648-2.

Zellner A. 1971. *An Introduction Bayesian Inference in Econometrics*. New York: John Wiley & Sons.

Zeugner S, Feldkircher M. 2015. Bayesian Model Averaging Employing Fixed and Flexible Priors: The BMS Package for R. *Journal of Statistical Software* **68**: 1–37. URL `https://www.jstatsoft.org/index.php/jss/article/view/v068i04`