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

Marcin Błażejowski†, Jacek Kwiatkowski‡

†University WSB Merito in Toruń, Poland
‡Nicolaus Copernicus University, Toruń, Poland

Gretl Conference 2023
1 Introduction
   • The problem no. 1
   • The problem no. 2

2 BACE package in gretl
   • History
   • A new way
   • New interface

3 Comparison with other software
   • Logit regression model
   • Poisson regression model
   • Run times
Introduction

BACE package in gretl

Comparison with other software

The problem no. 1

The problem no. 2

One of the basic problems in econometric modeling is the identification of the determinants of the dependent variable. When the number of possible explanatory variables is $K = 30$, the number of possible specifications is $2^K = 1,073,741,824$. Building a model with a very large number of explanatory variables can potentially lead to decision-making problems. Moreover, different types of modeling approaches can lead to different estimates and conflicting conclusions.
One of the basic problems in econometric modeling is the identification of the determinants of the dependent variable.
One of the basic problems in econometric modeling is the identification of the determinants of the dependent variable.

When the number of possible explanatory variables is $K = 30$, the number of possible specifications is $2^K = 1,073,741,824$. 
One of the basic problems in econometric modeling is the identification of the determinants of the dependent variable. When the number of possible explanatory variables is $K = 30$, the number of possible specifications is $2^K = 1,073,741,824$. Building a model with a very large number of explanatory variables can potentially lead to decision-making problems.
One of the basic problems in econometric modeling is the identification of the determinants of the dependent variable.

When the number of possible explanatory variables is $K = 30$, the number of possible specifications is $2^K = 1,073,741,824$.

Building a model with a very large number of explanatory variables can potentially lead to decision-making problems.

Moreover, different types of modeling approaches can lead to different estimates and conflicting conclusions.
Bayesian model averaging (BMA) is a useful alternative to other variable selection procedures, because it incorporates \textit{model uncertainty} into conclusions about estimated parameters. It is a standard Bayesian solution, where the inference on parameters is based on a weighted average over all possible models under consideration, rather than on one single regression model. These weights are Bayesian posterior probabilities of the individual models\textsuperscript{1}.

In some relatively simple models, such as linear regression, the marginal data density used for specification comparison can be represented by a closed-form analytic expression.
In some relatively simple models, such as linear regression, the marginal data density used for specification comparison can be represented by a closed-form analytic expression.

However, in many situations, such as limited variable models (e.g., logit and probit), it is difficult to quickly calculate the explanatory power of a very large set of competitive models.
In some relatively simple models, such as linear regression, the marginal data density used for specification comparison can be represented by a closed-form analytic expression.

However, in many situations, such as limited variable models (e.g., logit and probit), it is difficult to quickly calculate the explanatory power of a very large set of competitive models.

Numerical integration is computationally demanding and time-consuming in such cases (Newton & Raftery, 1994; Gelman et al., 2013; Pajor, 2017).
The solutions
The solutions

1. Generalized Linear Models:

2. Frequency approach based on Bayesian Information Criterion (BIC) – BACE
The solutions

1. Generalized Linear Models:
   - \textbf{ParMA} package in \texttt{gretl}— Lucchetti and Pedini (2022)— Bayesian approach

2. Frequency approach based on Bayesian Information Criterion (BIC) — BACE
The solutions

1. Generalized Linear Models:
   - ParMA package in gretl— Lucchetti and Pedini (2022) – Bayesian approach
   - BMA package in R — Raftery et al. (2022)

2. Frequency approach based on Bayesian Information Criterion (BIC) – BACE
**The solutions**

1. **Generalized Linear Models:**
   - **ParMA** package in *gretl*— Lucchetti and Pedini (2022) – Bayesian approach
   - **BMA** package in R — Raftery et al. (2022)

2. **Frequency approach based on Bayesian Information Criterion (BIC)** – BACE
   - **BMA** package in R — Raftery et al. (2022)
The solutions

1. Generalized Linear Models:
   - ParMA package in gretl — Lucchetti and Pedini (2022) — Bayesian approach
   - BMA package in R — Raftery et al. (2022)

2. Frequency approach based on Bayesian Information Criterion (BIC) — BACE
   - BMA package in R — Raftery et al. (2022)
   - mlogitBMA for BMA package in R — Sevcikova and Raftery (2022)
The solutions

1. Generalized Linear Models:
   - **ParMA** package in *gretl* — Lucchetti and Pedini (2022) – Bayesian approach
   - **BMA** package in R — Raftery et al. (2022)

2. Frequency approach based on Bayesian Information Criterion (BIC) — BACE
   - **BMA** package in R — Raftery et al. (2022)
   - **mlogitBMA** for **BMA** package in R — Sevcikova and Raftery (2022)
   - **BAS** package in R — Clyde (2022)
The solutions

1. Generalized Linear Models:
   - **ParMA** package in *gretl*— Lucchetti and Pedini (2022) – Bayesian approach
   - **BMA** package in R — Raftery et al. (2022)

2. Frequency approach based on Bayesian Information Criterion (BIC) – BACE
   - **BMA** package in R — Raftery et al. (2022)
   - **mlogitBMA** for **BMA** package in R — Sevcikova and Raftery (2022)
   - **BAS** package in R — Clyde (2022)
   - **BACE** package in *gretl*— Błażejowski and Kwiatkowski (2023)
BACE package - Main features

In terms of models with a limited dependent variable, our package can be considered as an alternative to the previously mentioned packages because:

1. It offers a larger number of various models: logit (including ordered and multinomial), probit (including ordered), Poisson regression, Tobit model, interval regression, and linear regression,
2. It is fast and provides consistent results,
3. It includes the latest proposals of BIC variants, and the latest jointness measures,
4. It is able to replicate the complex reduction strategy (Błażejowski, Kufel, & Kwiatkowski, 2020),
5. It has a GUI interface.
BACE package - Main features

In terms of models with a limited dependent variable, our package can be considered as an alternative to the previously mentioned packages because:

1. it offers a larger number of various models: logit (including ordered and multinomial), probit (including ordered), Poisson regression, Tobit model, interval regression, and linear regression,
BACE package - Main features

In terms of models with a limited dependent variable, our package can be considered as an alternative to the previously mentioned packages because:

1. it offers a larger number of various models: logit (including ordered and multinomial), probit (including ordered), Poisson regression, Tobit model, interval regression, and linear regression,
2. it is fast and provides consistent results,
BACE package - Main features

In terms of models with a limited dependent variable, our package can be considered as an alternative to the previously mentioned packages because:

1. it offers a larger number of various models: logit (including ordered and multinomial), probit (including ordered), Poisson regression, Tobit model, interval regression, and linear regression,

2. it is fast and provides consistent results,

3. it includes the latest proposals of BIC variants, and the latest jointness measures,
BACE package - Main features

In terms of models with a limited dependent variable, our package can be considered as an alternative to the previously mentioned packages because:

1. it offers a larger number of various models: logit (including ordered and multinomial), probit (including ordered), Poisson regression, Tobit model, interval regression, and linear regression,

2. it is fast and provides consistent results,

3. it includes the latest proposals of BIC variants, and the latest jointness measures,

4. it is able to replicate the complex reduction strategy (Błażejowski, Kufel, & Kwiatkowski, 2020),
In terms of models with a limited dependent variable, our package can be considered as an alternative to the previously mentioned packages because:

1. it offers a larger number of various models: logit (including ordered and multinomial), probit (including ordered), Poisson regression, Tobit model, interval regression, and linear regression,

2. it is fast and provides consistent results,

3. it includes the latest proposals of BIC variants, and the latest jointness measures,

4. it is able to replicate the complex reduction strategy (Błażejowski, Kufel, & Kwiatkowski, 2020),

5. it has a GUI interface.
The **BACE** ver. 3.0 is *improved* an *extended* version of the package presented at GC2017, Athens, Greece. So, our today’s speech may be treated as a continuation of that previous work. The timeline looks as follows:
The **BACE** ver. 3.0 is *improved* an *extended* version of the package presented at GC2017, Athens, Greece. So, our today’s speech may be treated as a continuation of that previous work. The timeline looks as follows:

**2018-07-22**: initial version (1.0)
The **BACE** ver. 3.0 is *improved* an *extended* version of the package presented at GC2017, Athens, Greece. So, our today’s speech may be treated as a continuation of that previous work. The timeline looks as follows:

- **2018-07-22**: initial version (1.0)
- **2019-11-01**: MPI based version (2.0)
The **BACE** ver. 3.0 is *improved* an *extended* version of the package presented at GC2017, Athens, Greece. So, our today’s speech may be treated as a continuation of that previous work. The timeline looks as follows:

**2018-07-22:** initial version (1.0)

**2019-11-01:** MPI based version (2.0)

**2023-03-15:** refactored version with completely new internal strategy for managing (storing and updating) posterior results obtained for already visited specifications + several models added + many more (3.0)
The **BACE** ver. 3.0 is *improved* an *extended* version of the package presented at GC2017, Athens, Greece. So, our today’s speech may be treated as a continuation of that previous work. The timeline looks as follows:

2018-07-22: initial version (1.0)
2019-11-01: MPI based version (2.0)
2023-03-15: refactored version with completely new internal strategy for managing (storing and updating) posterior results obtained for already visited specifications + several models added + many more (3.0)

**BACE** for *gretl* project main page:
https://sourceforge.net/projects/bace-for-gretl
The architecture of **BACE** 1.0 – even used in parallel via MPI interface – turned out to be a dead end due to the following factors:
The architecture of BACE 1.0 – even used in parallel via MPI interface – turned out to be a dead end due to the following factors:

- estimation based on ”manual” algebra operations via mols was effective only for linear regression
The architecture of **BACE 1.0** – even used in parallel via MPI interface – turned out to be a dead end due to the following factors:

- estimation based on ”manual” algebra operations via `mols` was effective only for linear regression
- bundle-based posterior results managing with `hexadecimal keys with multiple hex ↔ bin and bin → int` operations was massively CPU time consuming
The architecture of **BACE 1.0** – even used in parallel via MPI interface – turned out to be a dead end due to the following factors:

- estimation based on ”manual” algebra operations via `mols` was effective only for linear regression
- bundle-based posterior results managing with `hexadecimal` keys with multiple `hex ↔ bin` and `bin → int` operations was massively CPU time consuming
- furthermore, many updates (write back operations) lead to memory fragmentation slowing the package down dramatically
Surprisingly effective trick

We new that we must reduce any write back operations to minimum, i.e. updating only(!) models counter and leaving the rest of posterior results untouched.

✓ . . . and we managed to do it quite easily.

But there was still an inefficient method of mapping hex ↔ bin → int, that was problematic.

✓ . . . and we were thinking of how to speed up the package and – after the number of different trials – we decided to test something very strange:

model_id = sprintf "%d", M. mod_struct_old

where M.mod_struct_old is a binary vector:

[0 1 1 1 . . . 0 1 0]

1 × K
Surprisingly effective trick

- We new that we must reduce any write back operations to minimum, i.e. updating only(!) models counter and leaving the rest of posterior results untouched.
Surprisingly effective trick

- We knew that we must reduce any write back operations to minimum, i.e. updating only(!) models counter and leaving the rest of posterior results untouched.
  - … and we managed to do it quite easily.

Marcin Błazejowski and Jacek Kwiatkowski

BACE: A gretl Package for Model Averaging
Surprisingly effective trick

- We knew that we must reduce any write back operations to minimum, i.e. updating only(!) models counter and leaving the rest of posterior results untouched.
  - ... and we managed to do it quite easily.
- But there was still an inefficient method of mapping hex ↔ bin → int, that was problematic.
Surprising effective trick

- We new that we must reduce any write back operations to minimum, i.e. updating only(!) models counter and leaving the rest of posterior results untouched.
  ✓ ... and we managed to do it quite easily.
- But there was still an inefficient method of mapping hex ↔ bin → int, that was problematic.
  ✓ ... and we were thinking of how to speed up the package and – after the number of different trials – we decided to test something very strange:

```gretl
model_id = sprintf "%d", M.mod_struct_old
```
where `M.mod_struct_old` is a binary vector:
\[
\begin{array}{c}
0 & 1 & 1 & 1 & \cdots & 0 & 1 & 0 \\
\end{array}
\times K
\]
### Surprisingly effective trick

- We new that we must reduce any write back operations to minimum, i.e. updating only(!) models counter and leaving the rest of posterior results untouched.
  - ✓ ... and we managed to do it quite easily.

- But there was still an inefficient method of mapping hex ↔ bin → int, that was problematic.
  - ✓ ... and we were thinking of how to speed up the package and – after the number of different trials – we decided to test something very strange:

  \[
  \text{model_id} = \text{sprintf}("%d", \text{M.mod_struct_old})
  \]

  where \(\text{M.mod_struct_old}\) is a binary vector:

  \[
  \begin{bmatrix}
  0 & 1 & 1 & 1 & \ldots & 0 & 1 & 0 \\
  \end{bmatrix}_{1 \times K}
  \]
Core function that runs and controls the main loop:

```
function bundle BACE (list Y "Dependent variable", list X_list "List of independent variables (X)", bundle Options[null])
```
Core function that runs and controls the main loop:

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

The first two options are mandatory, while the last one is an optional set of additional arguments\(^2\).

\(^2\)Note, that in case of intreg estimator the Y list must contain two series with minvar and maxvar (left- and right- bounds respectively).
Core function that runs and controls the main loop:

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

The first two options are mandatory, while the last one is an optional set of additional arguments\(^2\).

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

\(^2\)Note, that in case of `intreg` estimator the Y list must contain two series with `minvar` and `maxvar` (left- and right- bounds respectively).
**BACE optional parameters**

**string** `Options["model_type"] = "logit"`: 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

**scalar** `Options["model_prior"] = 1`: the prior model size distribution to use; possible values are: 0—none, 1—Binomial, 2—Beta-Binomial
**BACE optional parameters**

**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 is calculated as:

$$avg\_model\_size\_tmp = \begin{cases} 
\frac{K}{2} & \text{if } with\_const \neq 2, \\
\frac{K + 1}{2} & \text{elsewhere}, 
\end{cases}$$

and $K$ stands for the number of independent variables

**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)

**scalar** Options["alpha"] = alpha_tmp: the significance level for the initial model, where **scalar** alpha_tmp takes the following values:

\[
\text{alpha}_\text{tmp} = \begin{cases} 
1.0 & \text{if model_type = "intreg"}, \\
0.6 & \text{elsewhere};
\end{cases}
\]

**scalar** Options["Nrank"] = 5: the number of top-ranked models to be printed

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

**scalar** Options["y_lag"] = 0: defines the lag order of the dependent variable
**Introduction**

**BACE package in Gretl**

**Comparison with other software**

**History**

A new way

New interface

---

**scalar** Options["hyperparameter"] = 0.5: the hyperparameter for prior-based Bayesian information criterion; for now, it is interpreted as $\gamma$ for the extended Bayesian information criterion (EBIC)

**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["verbosity"] = 2: verbosity of the output: 1—basic results, 2—full results; this option controls what is printed as output by BACE: 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["_gui"] = 0: this parameter is for internal use only and must not be modified by the user under any circumstances

Marcin Błażejowski and Jacek Kwiatkowski

**BACE: A Gretl Package for Model Averaging...**
**Introduction**

**BACE package in gretl**

**Comparison with other software**

**History**

A new way

**New interface**

```
scalar Options["Nrep"] = Nrep: the total number of Monte Carlo iterations (model draws) in MC³ algorithm, where scalar Nrep takes the following values:

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

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

do_show_progress = \begin{cases} 
1 & \text{if } \$sysinfo["gui_mode"], \\
0 & \text{elsewhere};
\end{cases}
```

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

do_plot_tmp = \begin{cases} 
1 & \text{if } \$sysinfo["gui_mode"], \\
0 & \text{elsewhere};
\end{cases}
```
**BACE main window**

![BACE GUI window interface.](image)

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

Marcin Błażejowski and Jacek Kwiatkowski

BACE: A gretl Package for Model Averaging...
### Logit regression model


<table>
<thead>
<tr>
<th>Variable</th>
<th>BACE</th>
<th>ParMA</th>
<th>BMA</th>
<th>BAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>const</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>ht</td>
<td>0.842</td>
<td>0.746</td>
<td>0.675</td>
<td>0.388</td>
</tr>
<tr>
<td>lwt</td>
<td>0.809</td>
<td>0.789</td>
<td>0.694</td>
<td>0.411</td>
</tr>
<tr>
<td>smoke</td>
<td>0.633</td>
<td>0.721</td>
<td>0.369</td>
<td>0.289</td>
</tr>
<tr>
<td>Drace_2</td>
<td>0.574</td>
<td>0.569</td>
<td>0.245</td>
<td>0.238</td>
</tr>
<tr>
<td>ptl</td>
<td>0.546</td>
<td>0.555</td>
<td>0.423</td>
<td>0.316</td>
</tr>
<tr>
<td>ui</td>
<td>0.485</td>
<td>0.496</td>
<td>0.296</td>
<td>0.238</td>
</tr>
<tr>
<td>Drace_3</td>
<td>0.443</td>
<td>0.580</td>
<td>0.167</td>
<td>0.197</td>
</tr>
<tr>
<td>age</td>
<td>0.232</td>
<td>0.242</td>
<td>0.089</td>
<td>0.150</td>
</tr>
<tr>
<td>ftv</td>
<td>0.151</td>
<td>0.153</td>
<td>0.014</td>
<td>0.087</td>
</tr>
</tbody>
</table>

**Table:** Comparison of Posterior Inclusion Probabilities (PIPs) among BACE, ParMA, BMA, and BAS packages for the logit regression model. Dataset `bithwt` presented in Hosmer and Lemeshow (2000), which consists of 189 births with eight risk factors associated with a dummy variable for low infant birth weight and 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.
### Table: Comparison of Posterior Inclusion Probabilities (PIPs) among BACE, ParMA, BMA, and BAS packages for the Poisson regression model. Cameron and Trivedi (1986).

Dependent variable: the number of doctor visits. Determinants: 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).
### Run time

<table>
<thead>
<tr>
<th>Model Type</th>
<th>Dataset</th>
<th>$N$</th>
<th>$K$</th>
<th>BACE</th>
<th>ParMA</th>
<th>BMA</th>
<th>BAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logit</td>
<td>bithwt</td>
<td>189</td>
<td>9</td>
<td>0.17</td>
<td>2.04</td>
<td>0.42</td>
<td>0.24</td>
</tr>
<tr>
<td>Ordered logit</td>
<td>nels</td>
<td>6649</td>
<td>13</td>
<td>5.13</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Probit</td>
<td>bithwt</td>
<td>189</td>
<td>9</td>
<td>0.22</td>
<td>2.47</td>
<td>2.56</td>
<td>—</td>
</tr>
<tr>
<td>Ordered probit</td>
<td>nels</td>
<td>6649</td>
<td>13</td>
<td>5.59</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Multinomial logit</td>
<td>hsbdemo</td>
<td>200</td>
<td>10</td>
<td>0.35</td>
<td>—</td>
<td>9.24</td>
<td>—</td>
</tr>
<tr>
<td>Poisson</td>
<td>rac3d</td>
<td>5190</td>
<td>12</td>
<td>4.14</td>
<td>8.88</td>
<td>4.30</td>
<td>224.0</td>
</tr>
<tr>
<td>Tobit (Type I)</td>
<td>tobit</td>
<td>200</td>
<td>4</td>
<td>0.07</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Interval regression</td>
<td>wtp</td>
<td>312</td>
<td>5</td>
<td>0.12</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Linear regression</td>
<td>FLS</td>
<td>72</td>
<td>41</td>
<td>45.38</td>
<td>14041.1</td>
<td>11.18</td>
<td>84.85</td>
</tr>
</tbody>
</table>

**Table:** Run time in seconds of BACE, ParMA, BMA, and BAS packages for selected datasets, $N$ — the number of observations, $K$ — the total number of explanatory variables in estimated model.
Empirical example no. 1: \textit{logit} model

\begin{verbatim}
set verbose off
set seed 1000000

# We load packages
include BACE.gfn

# We load birthwt dataset
open birthwt.gdt --frompkg=BACE --quiet

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

set stopwatch
RES = BACE(low, X, _{(model_type="logit", model_prior="beta-binomial", \bic_type="hbic", do_joint="myq", binary_threshold=0.6)})
printf "Total execution time: \%.2f sec.\n", $stopwatch
\end{verbatim}

Run
Empirical example no. 2: **tobit** model

```gretl
set seed 1000000
# We load packages
include BACE.gfn

# We load tobit dataset [https://stats.oarc.ucla.edu/r/dae/tobit-models/]
open tobit.gdt --frompkg=BACE --quiet

# We specify list of explanatory variables
list X = read math proggeneral progvocational

# BACE
set stopwatch
BACE(apt, X, _(model_type="tobit", high=800))
printf "Total execution time: %.2f sec.\n", $stopwatch
```

**Run**


Thank You for Your attention!