The Stata Journal (2015)
15, Number 2, pp. 325–349

Global search regression: A new automatic model-selection technique for cross-section, time-series, and panel-data regressions

Pablo Gluzmann
Center for Distributive, Labor and Social Studies
Argentine National Council of Scientific and Technological Research
and National University of La Plata
La Plata, Argentina
gluzmann@yahoo.com

Demian Panigo
Center for Worker Innovation
Argentine National Council of Scientific and Technological Research
National University of Moreno
and National University of La Plata
La Plata, Argentina
panigo@gmail.com

Abstract. In this article, we present `gsreg`, a new automatic model-selection technique for cross-section, time-series, and panel-data regressions. Like other exhaustive search algorithms (for example, `vselect`), `gsreg` avoids characteristic path-dependence traps of standard approaches as well as backward- and forward-looking approaches (like PcGets or relevant transformation of the inputs network approach). However, `gsreg` is the first code that 1) guarantees optimality with out-of-sample selection criteria; 2) allows residual testing for each alternative; and 3) provides (depending on user specifications) a full-information dataset with outcome statistics for every alternative model.

Keywords: st0383, gsreg, automatic model selection, vselect, PcGets, RETINA

1 Introduction

Econometric practitioners are commonly faced with global optimization issues. Identifying the real data-generating process (DGP) from a myriad of econometric models is analogous to looking for a global minimum in a highly nonlinear optimization problem. In both cases, some broadly accepted procedures lead to wrong or improvable results.[1]

While global optimization methods in mathematics have evolved, for example, from Newton–Raphson to genetic algorithms (and related search strategies), econometric

1. In econometrics, Leamer (1978) and Lovell (1983) document the low success rates of many widely used model-selection techniques, while Forrest and Mitchell (1991) stress the limitations of new standards (for example, genetic algorithms) in the numerical optimization.

model-selection techniques have changed from rudimentary (backward- or forward-stepwise) sequential regressions to more sophisticated approaches (PcGets, relevant transformation of the inputs network approach [RETINA], least angle regression, and least absolute shrinkage and selection operator; see Castle [2006]).

However, suboptimal path-dependent results still frequently emerge. Like genetic algorithms in global optimization problems, most automatic model-selection techniques (AMSTs) cannot guarantee a global optimum (the best DGP from available alternatives) in model selection. Outcomes can be affected by both search parameters (particularly test parameters) and search starting points (see Derksen and Keselman [1992]).

Newer AMSTs, like PcGets or RETINA, aim to avoid this problem by using alternative multipath–multisample backward- and forward-looking approaches, respectively. While these strategies significantly improve AMST outcomes (Marinucci 2008), they still fail to guarantee global optima because of unexplored reduction paths; the size–power trade-off; and cumulative type I errors of sequential testing, especially in small-sample problems.

The combination of nonexhaustive search (like single- or multiple-path search strategies) and sequential testing (either forward- or backward-looking) frequently affords some cost in terms of statistical inference (depending on test size and selected paths, it will take the form of model under- or overfitting), and the terminal model will coincide with the best DGP.

These weaknesses together with increasing computational capabilities explain the widening use of alternative exhaustive search methods. Unlike global optimum search in mathematics,[2] a model-selection problem in econometrics is always self-constrained. The number of points (models) to be evaluated will never be infinite—it will be a certain integer defined by 2^n, where n is the number of initially admissible covariates. This quantity, while exponentially increasing in n, is far more manageable than any unconstrained nonlinear global optimization problem (see figure 1).

2. The meaning of exhaustive search in mathematics (for example, in nonlinear optimization problems) is not completely satisfactory. Algorithms like Pattern Search in Matlab provide a useful example of exhaustive search in a global optimization context. Indeed, the iterative Pattern Search algorithm looks for a global minimum in variable-size mesh until a threshold level is attained. However, without constraints, the problem must be evaluated at an infinite number of points. Using polling method options, the Pattern Search algorithm reduces the number of iterations to a convenient dimension. Nevertheless, the stronger the constraint, the higher the loss of the global minimum accuracy.

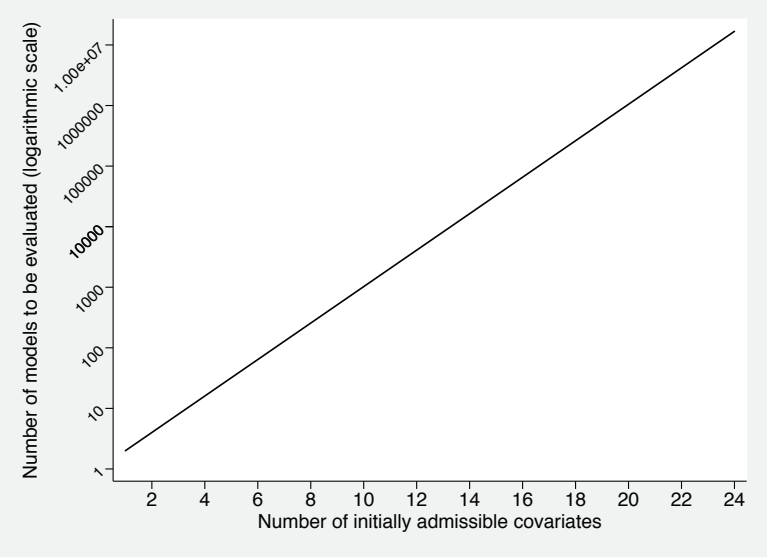

Figure 1. Exhaustive search: Alternative models to be evaluated at different numbers of initially admissible covariates

All in all, the choice between exhaustive and nonexhaustive search is determined by the trade-off between time and accuracy. Current-generation AMSTs try to account for both dimensions, standing somewhere between pure time-saving techniques (for example, first-generation AMSTs like stepwise regressions) and pure accuracy-improving methods (exhaustive search). AMSTs evolve from speed to goodness-of-fit, as long as processing-power innovation increases computational capabilities.

With the release of Stata 13, StataCorp stated that running a linear regression on 10 covariates and 10,000 observations takes 0.034 seconds on an Intel 2.4 GHz Core 2 Quad with Stata/SE for Windows 7 (http://www.stata.com/why-use-stata/fast). Exhaustive search of the best DGP in the same example (10 covariates and 10,000 observations) will involve 1,024 linear regressions in about 34 seconds. Moreover, when using one of the latest Intel Xeon processors (Xeon X5698, 2011, 4.4 GHz), the same procedure takes 19.3 seconds.

Forty years ago, running 1,000,000 regressions (for example, the number of equations to be estimated for an exhaustive search on a general model of 19–20 initially accepted covariates and 10,000 observations) would have taken about 25 years (using the Intel 4004 processor of 108 KHz). Today, it takes only about 5 hours (using the Intel Xeon X5698 processor of 4.4 GHz) or less (for example, only about 4 hours by overclocking the last AMD FX-9590 processor to obtain up to 5.3 GHz).

This exponential increase in hardware computational capabilities (figure 2) has been complemented by newer software codes to implement exhaustive search in econometric model-selection problems (like `vselect` in Stata). However, none of the codes provides

exhaustive outcomes for sensitivity analysis (for example, coefficient- or test-probability distributions for any alternative model structure) or high accuracy when out-of-sample selection criteria are used (or when hypothesis testing is necessary, for example, when testing white-noise residuals).

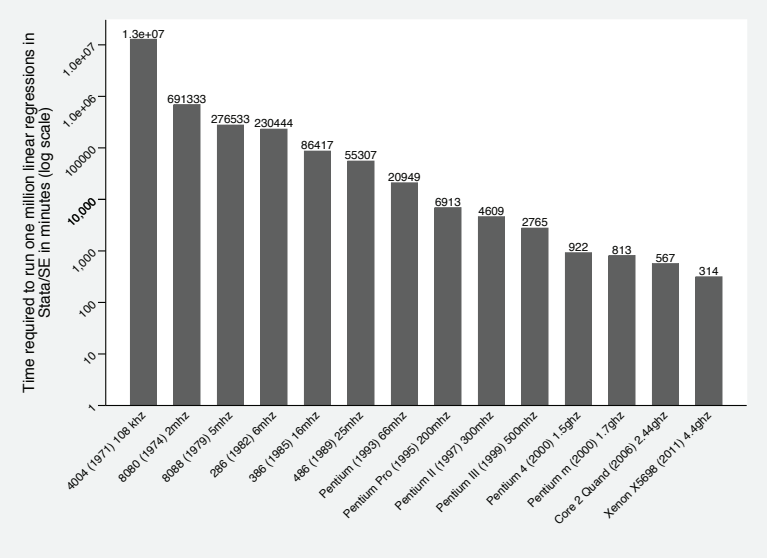

Figure 2. Exhaustive search: Alternative models to be evaluated at different numbers of initially admissible covariates

To fill this gap, we developed `gsreg`—the first code for exhaustive search in AMST that 1) guarantees optimality with out-of-sample selection criteria; 2) allows residual behavior testing for each alternative; and 3) provides (depending on user specifications) a full-information dataset with outcome statistics for every alternative model.[3]

We structure this article in six sections. In the following section, we discuss strengths and weaknesses of main automatic model-selection approaches. In section 3, we introduce the `gsreg` command, including its algorithm, stages, and uses. In section 4, we give the syntax and options. We then present some examples of the application of `gsreg` in section 5, and we describe the features of stored results in section 6.

3. `gsreg` will initially be used for small-size problems in standard personal computers (for example, to find the best DGP over different combinations of 20 or fewer potential covariates, which can be solved in a couple of hours). However, larger calculations will soon be manageable, because a "parallelization revolution" is coming soon. A few years from now, it will be easy to solve a one-billion regression problem with `gsreg` in two hours, using general-purpose computing on graphics processing units and CUDA or OpenCL-like reengineering to improve `gsreg` parallelization capabilities (for example, to fully exploit the `part()` option potential).

2 Distinctive features of main AMSTs

By combining and extending Hendry (1980), Miller (1984), Gatu and Kontoghiorghes (2006), and Duarte Silva (2009) categorizations, we can generate the conceptual tree of model-selection techniques shown in figure 3.

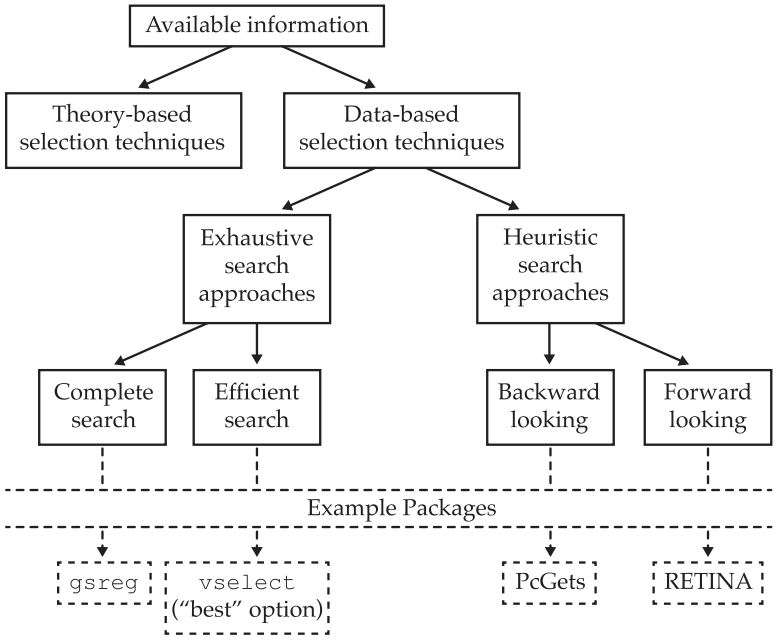

Figure 3. Conceptual tree of model-selection techniques

The first-level choice is related to data mining—one of the most important path-breaking controversies in applied economics—which starts in the 1930s and continues throughout the twentieth century with seminal contributions of Frisch (1934), Haavelmo (1944), Leamer (1978), Lovell (1983), Gilbert (1986), Hendry (1995), and many others.

Recent econometric developments tend to advocate for data-based model-selection techniques, especially the automated ones.[4] Within this family, however, an internal consensus has yet to be achieved. At the end of the 1960s, both exhaustive (also known as exact) and heuristic approaches were very popular. Heuristic subset selection was pioneered by the stepwise regression algorithm of Efroymson (1960), while exhaustive search was initially associated with the optimal or complete regression strategy of Coen, Gomme, and Kendall (1969).

4. It is useful to examine the econometric "zeitgeist" evolution by comparing Miller's (1984) discussions against recent debates in *Econometric Theory* (vol. 21, 2005, devoted to "Automated inference and the future of econometrics").

Box and Newbold's (1971) criticisms on exhaustive search techniques (for example, they are unfeasible for large-size problems) and Berk's (1978) objections to stepwise algorithms (for example, they do not guarantee optimality) have highlighted the need for better alternatives.

The number of exhaustive and heuristic model-selection techniques has grown exponentially in the past 40 years. Alternative algorithms arose, such as nonnegative Garrote (Breiman 1995); least absolute shrinkage and selection operator (Tibshirani 1996); least angle regression (Efron et al. 2004); `vselect`—leaps and bounds (Furnival and Wilson 1974; Lindsey and Sheather 2010); PcGets and Autometrics (Krolzig and Hendry 2001; Doornik 2009); and RETINA (Pérez-Amaral, Gallo, and White 2003).[5] However, most of them have important limitations for robustness analysis or out-of-sample (and in-sample) optimality.

Regarding the general-to-specific approach (PcGets), we must note that path dependence (Pagan 1987) has not been completely eliminated, small-sample problems persist (Marinucci 2008), and out-of-sample results are relatively poor (Herwartz 2007). Even the designers (Krolzig and Hendry, 2001; 839) concluded that "the empirical success of PcGets must depend crucially on the creativity of the researcher".

As for the specific-to-general (RETINA) strategy, the software is still unable to guarantee model-selection optimality because neither path dependence nor cumulative type I errors were fully removed with its multiple-sample or multiple-path methodology. Moreover, RETINA's usual under-parameterization (which may be useful for forecasting purposes) could have some negative effects on in-sample fitting and explanation properties. According to Castle (2006, 46), "The specific-to-general methodology tends to have an ad hoc termination point for the search, and alternative path searches are unbounded, implying that the approach could miss the local DGP. Moreover, the null rejection frequency will not be controlled as the number of tests conducted will depend on the termination point, and failure of misspecification tests is likely at the initial stages, invalidating conventional tests. This does mean that there is no guarantee that the final model selected by RETINA is congruent, which may or may not be relevant for forecasting models."

5. For further details about other selection techniques (PcGets/Autometrics and RETINA), see Davidson and Hendry (1981); Pagan (1987); Derksen and Keselman (1992); Krolzig and Hendry (2001); Pérez-Amaral, Gallo, and White (2003); Castle (2006); Herwartz (2007); Doornik (2009); and Marinucci (2008). For a better explanation of `vselect`, see Lindsey and Sheather (2010), Draper and Smith (1998), and Furnival and Wilson (1974).

Finally, although the efficient exhaustive search package `vselect` overcomes in-sample optimality issues of the alternatives mentioned above, it still faces the following limitations:[6] 1) the algorithm's main property does not apply for out-of-sample model-selection problems, and 2) while more efficient than complete exhaustive methods, the "`vselect, best`" approach becomes unfeasible and time consuming for large-size problems (because the success reduction rate will not compensate the exponential increase of the problem size with the number of potential covariates).

3 The global search regression (gsreg) procedure

Despite the documented increase in computational capabilities, our complete exhaustive algorithm is particularly recommended for small-size (fewer than 30 variables) model-selection problems, where 1) out-of-sample selection criteria will be used to select the optimal choice or 2) the main objective is parameter stability across different model specifications.[7] However, its options are encompassing enough to transform `gsreg` into a flexible device for many other uses. In what follows, we present its features.

The `gsreg` command has two major stages. In the first stage, it creates a set of lists, wherein each list contains one of the possible sets of dependent variables, and therefore, the full set of lists contains all possible combinations of candidate covariates. In the second stage, the command performs a regression for each of the lists previously created.

In the first stage, the set of lists is determined according to the following steps:

1. The algorithm determines an inventory containing the total set of candidate variables, L_vc, according to the list of user-specified original variables and the additional covariates to be included as candidates if the option `dlags()`, `ilags()`, or `lags()` is specified.

2. If the `ncomb()` option is not specified, a first set of lists, SL, is created by taking all possible combinations without repetition of candidate variables (which include all combinations taken from 1 to the total number of variables in L_vc). Otherwise, SL is created by taking all combinations without repetition of candidate variables taken from *#1* to *#2* defined in `ncomb()`. So, SL $= (L_{int1}, \ldots, L_{int2})$, where each L_i is a particular subset of the set of candidate variables L_vc.

6. As noted by Lindsey and Sheather (2010), model-selection methodologies usually face multiple inference issues (for example, different significance levels). Following Sheather (2009), `vselect` authors propose to implement cross-validation techniques by splitting "the data into two parts, performing variable selection on one part (train) and using the other (test) for evaluating the resulting model" (Lindsey and Sheather 2010, 651).
7. Cross-validation techniques are recommended for `gsreg` as well as for `vselect`. Indeed, both commands could be complementary for this purpose: `vselect` could be used to efficiently find the best in-sample model, and `gsreg` could be used to check out-of-sample consistency.

3. If the `squares` option is specified, an additional list is created from each L_i of point 2 (SqL_i), and it includes all covariates in L_i plus all their squares. Then, the whole set of SqL_i lists (SqL) is added to the SL. If the `cubic` option is additionally specified, another group of lists (Cub$L$) is created from the SqL set by generating a CubL_i list for each SqL_i list, in which SqL_i covariates are complemented by L_i cubes. After that, the Cub$L$ set is added to SL.[8]

4. If the `interactions` option is specified, an additional IntL_i list is created from each L_i, which includes all L_i variables plus all possible combinations without repetition of the interactions of these variables. Then, IntL_i lists are added to SL.

5. If the `fixinteractions` option is specified, users can create a FintL_i list from each L_i, which not only includes all L_i variables but also all possible combinations without repetition of the interactions between L_i variables and `fixvar()` variables (see below).

6. If the `schange()` option is specified, a new set of lists (SC) is created from SL (already modified, if specified, by `ilags()`, `dlags()`, `ncomb()`, `squares`, `cubic`, `interactions`, and `fixinteractions`) to test for structural change in every bivariate relationship, including all possible combinations without repetition of the interactions between variables within SL and the user-defined variable of structural change (for example, a step- or point-dummy variable). Then, SC is added to SL.

In the second stage, `gsreg` exhaustively performs one regression per SL, saving coefficients and different statistics (default and user-defined) in a Stata `.dta` file. For each SL, `gsreg` outcomes include

a. coefficients and t statistics of each covariate;

b. regression number (regression ID), number of covariates, and number of observations; and

c. default additional statistics (adjusted-R^2, root mean squared error), optional additional statistics (such as residual test p-values or out-of-sample root mean squared error), and other user-defined statistics that the user specifies in the `cmdstat()` option.

8. Notice that this procedure dismisses all lists (regressions) that include squares of a certain variable but do not include the original variable (for example, in levels), thereby reducing the number of estimations to be performed. If users would like to estimate the cases where a given variable appears only in quadratic terms, they should include the square of that variable (or all variables desired) as an independent variable in the original L_vc set. Also notice that for the `cubic` option, the algorithm generates lists with only the cubes of the variables for which the square was included. Similar criteria were applied to the `interactions`, `fixinteractions`, and `schange()` options.

Figure 4 summarizes the `gsreg` procedure:

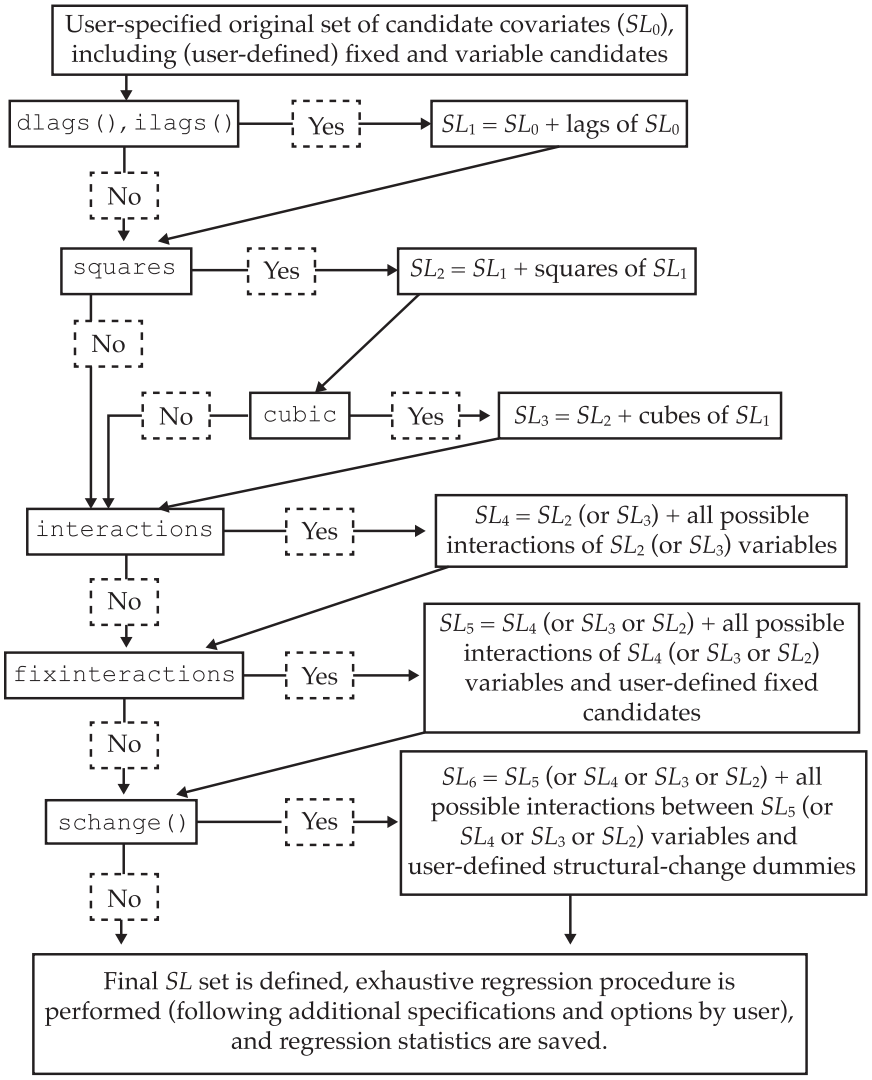

Figure 4. Schematic diagram summarizing the `gsreg` procedure

4 The gsreg command

4.1 Syntax

The syntax for the gsreg command is

gsreg *depvar varlist_ocand* [*if*] [*in*] [*weight*] [, ncomb(*#1*, *#2*) samesample
nocount backup(#) part(*#1*, *#2*) dlags(*numlist*) ilags(*numlist*)
lags(*numlist*) fixvar(*varlist_fix*) schange(*varname_sc*) interactions
squares cubic fixinteractions outsample(#) cmdest(*commandname*)
cmdoptions(*commandoptions*) cmdstat(*commandstats*)
cmdiveq(*varlist_end* = *varlist_inst*) aicbic hettest hettest_o(*hettestoptions*)
archlm archlm_o(*archlmoptions*) bgodfrey bgodfrey_o(*bgodfreyoptions*)
durbinalt durbinalt_o(*durbinaltoptions*) dwatson sktest
sktest_o(*sktestoptions*) swilk swilk_o(*swilkoptions*) sfrancia testpass(#)
resultsdta(*newbasename*) replace double compact nindex(*lcimplist*)
mindex(*lcimplist*) best(#)]

4.2 Options

General options

ncomb(*#1*, *#2*) specifies the minimum[9] and maximum number of variable (instead of user-specified fixed) covariates to be included in the procedure. gsreg will perform all possible combinations (regressions) between candidate variables taken from *#1* to *#2*. *#1* must be less than or equal to *#2* and, additionally, the number of candidates must be greater than or equal to *#2*. The default is to run all possible combinations without repetition of size 1 to n (the total number of candidates).

samesample specifies that all regressions be performed over the same sample of observations, defined as the largest common sample. By default, gsreg performs each regression with the maximum number of common observations available for the covariate subset used in each case.

nocount hides the number of the regression being estimated from the screen. The default is to show the regression number (used for identification purposes) and the total number of regressions to be estimated for each model.

backup(#) creates # backup files during the execution of gsreg. Each backup will contain approximately 1/# regressions. Each file will be saved in the ongoing working directory and will be named with the name of the results database (gsreg by default) followed by the number of partition and the total number of backups spec-

9. ncomb() allows 0 to be included as the minimum value only if the option fixvar() is specified.

ified in # (for example, `gsreg_part_1_of_#.dta`). All partitions will be deleted at the end of the execution of `gsreg`, and the results database will be stored. If the number of regressions is lower than #, the number of partitions will be reset to the number of regressions.

`part(`*#1*`, `*#2*`)` runs a specific partition of all regressions. The partition will contain approximately 1/*#2* regressions. `gsreg` will save the partition (instead of the results database) in the ongoing working directory. If the total number of regressions is lower than *#2*, the number of partitions (*#2*) will be reset to the number of regressions.

Lag-structure options

`dlags(`*numlist*`)` allows dependent variable lags (*depvar*) to be included among candidate covariates. `tsset` must be specified when using this option.

`dlags(`*#*`)` includes the # dependent variable lag among candidates.

`dlags(`*#1*`/`*#2*`)` includes all dependent variable lags from *#1* to *#2* considering one-unit intervals among candidates.

`dlags(`*#1 #2 #3*`)` includes the *#1*, the *#2*, and the *#3* dependent variable lags among candidates.

`dlags(`*#1* `(#`*d*`)` *#2*`)` includes all dependent variable lags from *#1* to *#2* considering #*d*-unit intervals among candidates.

`dlags(`*#1 #2 #3*`...`*#4* `(#`*d*`)` *#5*`)` includes the dependent variable lags *#1*, *#2*, and *#3*, plus all dependent variable lags from *#4* to *#5* considering #*d*-unit intervals among candidates.

`ilags(`*numlist*`)`[10] allows independent variable lags to be included among original candidates. The syntax is flexible and identical to that used in `dlags()`. `tsset` must be specified when using this option.

`lags(`*numlist*`)` allows dependent and independent variable lags to be jointly included among original candidates. It replaces `dlags()` and `ilags()` when the argument is identical. `tsset` must be specified when using this option. `lags()` cannot be specified together with `dlags()` or `ilags()`.

Fixed-variable options

`fixvar(`*varlist_fix*`)` allows users to specify a subset of covariates to be included in all regressions. Variables defined in *varlist_fix* must not be included among the standard candidates (*varlist_ocand*).

10. Using `ilags()` and `dlags()` generates independent and dependent variable lags (respectively) before using `gsreg` and includes them among original candidates. Users looking for different candidate lag structures for each covariate should not specify the option `ilags()`; users should instead create desired candidate lag structures before using `gsreg` and include them in the whole set of original candidates.

Options for transformations and interactions

`schange(`*varname_sc*`)` tests structural change of slopes (using dummy *varname_sc* as interaction with all candidates) or dependent variable levels (alternatively allowing *varname_sc* to interact with the intercept). Interactions of *varname_sc* with any candidate will be included only if this candidate is in the equation. *varname_sc* must not be included among original candidates (*varlist_ocand*) because it will be used only for structural change.

`interactions` includes additional covariate candidates to evaluate all possible interactions without repetition among original candidates (*varlist_ocand*) and lags, if specified in `dlags()`, `ilags()`, or `lags()`. Interactions between any two candidates will be allowed only if both of them are in the equation. When this option is used together with `schange()`, the structural change of interactions will be used only if these interactions are included in the estimated specification.

`squares` adds the squares of each variable in *varlist_ocand* (and lags, if specified in `dlags()`, `ilags()`, or `lags()`) as new candidates. Each square will be accepted as a regression covariate only if its level (original variable) is present in the equation. Similarly, when this option is used together with `schange()`, the structural change of the squares will be allowed only if these squares are in the equation.

`cubic` is similar to `squares`. It includes cubes of each variable in *varlist_ocand* (and lags, if specified in `dlags()`, `ilags()`, or `lags()`) as new candidates. These cubes will be accepted as covariates only if level and square of the same variable are included in the equation. When this option is used together with `schange()`, the structural change of the cubes will be allowed only if these cubes are in the equation.

`fixinteractions` is similar to `interactions`, but it includes all possible interactions without repetition among *varlist_ocand* (and lags, if specified in `dlags()`, `ilags()`, or `lags()`) as well as each fixed variable in *varlist_fix*.

Options for time-series and panel-data forecasts

`outsample(#)` is used in time-series and panel-data models. It splits the sample into two. The first subsample is used for regression purposes, and the second one is applied to evaluate forecast accuracy. `outsample(#)` leaves the last # periods to make forecasts (so that regressions are performed over the first $T-\#$ periods, where T is the total number of available time-series observations). When this option is specified, `gsreg` calculates and stores the `rmse_in` (in-sample root mean squared error) between period 1 and $N-\#$ and the `rmse_out` (out-of-sample root mean squared error) between period $N-\#$ and N. `tsset` must be specified when using this option.

Regression command options

`cmdest(`*commandname*`)`[11] allows the user to implement alternative regression commands. The default is `cmdest(regress)`. *commandname* may be `regress`, `xtreg`, `probit`, `logit`, `areg`, `qreg`, or `plreg`, but the option additionally accepts any regression command that respects the syntax of `regress` and stores results (matrices `e(b)` and `e(V)`) in the same way. `ivregress` is also accepted using option `cmdiveq(`*varlist_end* `=` *varlist_inst*`)`.

`cmdoptions(`*commandoptions*`)` allows additional options supported by *commandname* to be added for each regression.

`cmdstat(`*commandstats*`)` enables `gsreg` to save additional regression statistics[12] as scalars `e()` by the regression command (*commandname*).

`cmdiveq(`*varlist_end* `=` *varlist_inst*`)` includes a list of endogenous variables (*varlist_end*) and a list of instruments (*varlist_inst*) when the estimator command is `ivregress`. When using this option, `cmdest(ivregress 2sls)`, `cmdest(ivregress liml)`, or `cmdest(ivregress gmm)` must be specified. The endogenous variables must be included in *varlist_fix* (see option `fixvar()`) or in *varlist_ocand*.

Postestimation options

`aicbic` calculates `estat ic` after each regression to obtain Akaike information criteria (AIC) and Bayesian information criteria (BIC).

`hettest` calculates default `estat hettest` after each regression and saves p-values.

`hettest_o(`*hettestoptions*`)` allows options to be added to `hettest`.

`archlm` runs default `estat archlm` after each regression and saves p-values. `tsset` must be specified when using this option.

`archlm_o(`*archlmoptions*`)` allows options to be added to `archlm`.

`bgodfrey` computes default `estat bgodfrey` after each regression and saves p-values. `tsset` must be specified when using this option.

`bgodfrey_o(`*bgodfreyoptions*`)` allows options to be added to `bgodfrey`.

`durbinalt` calculates `estat durbinalt` after each regression and saves p-values. `tsset` must be specified when using this option.

`durbinalt_o(`*durbinaltoptions*`)` allows options to be added to `durbinalt`.

`dwatson` runs `estat dwatson` after each regression and saves the Durbin–Watson statistic. `tsset` must be specified when using this option.

11. Not all `gsreg` options can be used in any regression command. For regression commands with required options, options must be specified in `cmdoptions()`.
12. `gsreg` automatically saves the number of observations, `obs`; the number of covariates, `nvar`; the adjusted-R^2, `r_sqr_a`; and the root mean squared error, `rmse_in`.

`sktest` computes `sktest` after each regression and saves the p-value of the joint probability of skewness and kurtosis for normality. `tsset` must be specified when using this option.

`sktest_o(`*sktestoptions*`)` allows options to be added to `sktest`.

`swilk` calculates `swilk` after each regression and saves the p-value of the Shapiro–Wilk normality test. `tsset` must be specified when using this option.

`swilk_o(`*swilkoptions*`)` allows options to be added to `swilk`.

`sfrancia` runs `sfrancia` after each regression and saves the p-value of the Shapiro–Francia normality test. `tsset` must be specified when using this option.

`testpass(#)` reduces the size of the outcome database by saving only those regression results that fulfilled all user-specified residual tests (at a # significance level).

Output options

`resultsdta(`*newbasename*`)` allows the results-database name to be user-defined in *newbasename*. By default, the name will be `gsreg.dta`.

`replace` replaces the results database (with the same name) if it already exists in the ongoing working directory.

`double` forces results to be created and saved in double format, that is, with double precision.

`compact` reduces the size of the results database by deleting all coefficients and t statistics. In their place, `gsreg` creates a string variable called `regressors` that describes which candidate variables are included in each regression. This variable takes value `1` in position # if the candidate variable with position # is included in the equation, and it takes value `.` if it is not. Variable positions are kept in a small database called *newbasename*`_labels.dta` (where *newbasename* is the results database's user-defined name).

`nindex(`*lcimplist*`)` allows an index of normalized accuracy to be specified. Regressions will be ordered from highest to lowest in the results database, the best regression according to `nindex()` will be shown on the screen, and `e()` results of this regression will be saved in memory at the end of the `gsreg` execution. The default is based on the adjusted-R^2 (`r_sqr_a`). User choices about goodness-of-fit or forecast accuracy criteria on `nindex()` can flexibly be specified in *lcimplist*. Using user-selected weights and ranking variables, *lcimplist* allows complex arguments to create multinomial ordering criteria. Any results-database variable can be used as a ranking variable in the *lcimplist* argument (for example, `r_sqr_a`, `rmse_in`, `rmse_out`, `aic`, or `bic`), but it must be preceded by a user-defined real number weight (for example, `nindex(0.3 r_sqr_a -0.3 aic -0.4 bic)`). Each variable included in *lcimplist* is normalized using the whole sample average (across all regressions) of the same variable.

`mindex(`*lcimplist*`)` and `best(#)` must be specified together. `mindex()` generates a normalized ranking index like `nindex()` and has the same syntax as `nindex()`, but the normalization of its arguments is developed using averages obtained from the best #+1 regressions. Therefore, `mindex()` is updated with each additional regression and only the best (in terms of *lcimplist*) # regression results are stored. The joint use of `mindex()` and `best()` can strongly reduce database size (and RAM requirements), making larger model-selection problems feasible. However, because `mindex()` must be recalculated with every regression, `gsreg` could run slower when using `mindex()` (particularly for small model-selection problems).

5 Examples

`gsreg` can be used for many purposes. In this section, we introduce three illustrations of different `gsreg` applications. For brevity, option specifications are not fully discussed here (see the `gsreg` help file for further details).

In the first example, we use artificial data to show how `gsreg` can be used to obtain the best model in terms of in-sample goodness-of-fit, provided that regression residuals fulfill some desirable property. In the second example, we show that `gsreg` could be indispensable when a user's main concern for model selection is out-of-sample accuracy. In our third example, we illustrate another valuable `gsreg` application: parameter stability analysis across different control variable models.

5.1 Model selection and residual tests

The leaps-and-bounds efficient model-selection methodology (introduced by the `vselect` command) has the following two salient characteristics: 1) by using an exhaustive search method (see sections 2 and 2.3), it ensures optimality for any in-sample model selection criterion; and 2) the embedded Furnival and Wilson (1974) efficient algorithm allows exhaustive search to be performed over a larger number of covariates than is feasible for complete search algorithms.

However, the best models in terms of some in-sample information criteria do not necessarily fulfill required residual properties (something left aside by `vselect` and other model-selection Stata commands like `stepwise`). The following example shows why `gsreg`-like algorithms can be essential to solve this problem. Suppose we wish to obtain the best model to explain y, using some combination of two covariates, x and z, and assuming the following DGP:

$$y_t = \beta_0 + \beta_{1t} x_t + \beta_{2t} z_t + u_t$$

with $t = 1, \ldots, 1000$, $\beta_0 = 1$, $\beta_{1t} = 0.9$ if $t < 600$, $\beta_{1t} = 0$ if $t \geq 600$, $\beta_{2t} = 0$ if $t \leq 800$, $\beta_{2t} = 0.1$ if $t > 800$, $z \sim U[0,1]$, $x \sim U[0,2]$ if $t < 600$, $x \sim U[0,2.4]$ if $t \geq 600$, and $u \sim N(0,1)$.

By construction, covariate x has a higher explanatory power than z, but it tends to generate heteroskedasticity problems.

We will use `gsreg` to estimate all possible combinations. With two candidate covariates, there will be only three possible models. For each regression, we will generate and save information (in the `res1.dta` file) about 1) the AIC and the BIC (using the `aicbic` option) and 2) the p-value of the standard heteroskedasticity test (using the `hettest` option). Finally, we will ask `gsreg` to display the best regression for a multinomial normalized `nindex()` based on the adjusted-R^2, the AIC, and the BIC by using the following command statement:

```
. gsreg y x z, resultsdta(res1) replace hettest aicbic
> nindex(0.3 r_sqr_a -0.3 aic -0.4 bic)
-----------------------------------------------------
Total Number of Estimations: 3
-----------------------------------------------------
Computing combinations...
Preparing regression list...
Doing regressions...
Estimation number 1 of 3
Estimation number 2 of 3
Estimation number 3 of 3
Saving results...
file res1.dta saved
-----------------------------------------------------
Best estimation in terms of 0.3 r_sqr_a -0.3 aic -0.4 bic
Estimation number 3
-----------------------------------------------------
```

Source	SS	df	MS	Number of obs	=	1,000
				F(2, 997)	=	35.02
Model	86.5026807	2	43.2513403	Prob > F	=	0.0000
Residual	1231.34003	997	1.23504517	R-squared	=	0.0656
				Adj R-squared	=	0.0638
Total	1317.84271	999	1.31916187	Root MSE	=	1.1113

y	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
x	.4360614	.0545238	8.00	0.000	.3290669	.5430559
z	.2705071	.1242505	2.18	0.030	.0266845	.5143296
_cons	.9651791	.0912015	10.58	0.000	.7862103	1.144148

The best model in terms of `nindex()` includes both x and z covariates. However, our `res1.dta` (partially reproduced in table 1 below) shows some interesting results.[13]

In table 1, we can see that the first model, with only x as covariate, is the best one in terms of the BIC, while the best model in terms of both the AIC and the adjusted-R^2 is that using x and z as covariates. However, both models fail to fulfill the residual homoskedasticity requirement (with `hettest` p-values lower than 0.01). The z model (model 2), although suboptimal under any selection criterion, is the only one for which the null hypothesis of homoskedasticity cannot be rejected.

13. Many `res1.dta` columns have been omitted to reduce the size of table 1 (such as number of observations, number of variables, the root mean squared error, regression coefficients, and t statistics).

Table 1. Example 1: Main stored results

Order	Model	r_sqr_a	aic	bic	hettest	nindex()
1	x	0.0602569	3054.723	3064.538	0.0005750	0.5621497
2	z	0.0046997	3112.161	3121.976	0.2120479	1.1536771
3	x z	0.0637653	3051.980	3066.703	0.0005339	0.5915275

A similar exercise can be simulated for related problems of serial correlation or nonnormal residuals, where best models in terms of some information criteria do not fulfill residuals requirements while suboptimal models surprisingly do.

When user concern is focused on estimation robustness, residuals requirements become crucial and `gsreg` provides a better alternative than other model-selection commands (like `vselect` or `stepwise`) to ensure optimality among admissible models (for example, to find the optimal model among those with white-noise residuals).

5.2 Out-of-sample prediction

Friedman's (1953) contribution still generates a vigorous debate among epistemologists confronting "instrumentalism" and "realism" (see Mäki [1986] or Caldwell [1992]). Some still blame Friedman for generalizing the misleading idea that forecast accuracy (even using models with "false" assumptions) is the only valid mechanism to choose among competing theories.

In econometrics, there is some parallelism with the "measurement without theory" debate associated with Koopmans's (1947) work from almost 70 years ago (reviewed by Hendry and Morgan [1995]) and more recent methodological discussions about in-sample versus out-of-sample model-selection mechanisms. Renowned econometricians like Ashley, Granger, and Schmalensee (1980, 1149) assert that "a sound and natural approach [to testing predictive ability] must rely primarily on the out-of-sample forecasting performance".

It is not surprising that many colleagues increasingly try to overcome this last controversy by examining both in-sample and out-of-sample model outcomes.

In this context, `gsreg` can ensure in-sample as well as out-of-sample model-selection optimality, reducing user concerns about structural breaks in multivariate relationships.

To illustrate this point, suppose that we wish to get the best model of y (in terms of some out-of-sample criteria) using x or z, with 100 time-series observations (using the last 20 for out-of-sample model evaluation) and assuming the following DGP:

$$y_t = \beta_0 + \beta_1 x_t + \beta_{2t} z_t + u_t$$

with $\beta_0 = 1$, $\beta_1 = 1$, $\beta_{2t} = 1$ if $t \leq 70$, $\beta_{2t} = 0$ if $t > 70$, and $x, z, u \sim N(0, 1)$.

By construction, both covariates have a high in-sample explanatory power, but z becomes nonsignificant for out-of-sample evaluation purposes.

If the structural change is unknown (and therefore disregarded) and we do not use `gsreg` to evaluate forecast accuracy, the best representation of y will include x and z as covariates.

Alternatively, users concerned about the dangerous effects of potential structural breaks will exploit some database subsample to check parameter stability (for example, the last 20 observations) and use `gsreg` to examine both explanatory power and forecast accuracy of each model. For this example, the simplest command could be

```
. gsreg y x z, outsample(20) replace
-----------------------------------------------------
Total Number of Estimations: 3
-----------------------------------------------------
Computing combinations...
Preparing regression list...
Doing regressions...
Estimation number 1 of 3
Estimation number 2 of 3
Estimation number 3 of 3
Saving results...
file gsreg.dta saved
-----------------------------------------------------
Best estimation in terms of r_sqr_a
Estimation number 3
-----------------------------------------------------
```

Source	SS	df	MS		
Model	189.57001	2	94.7850049	Number of obs	= 80
Residual	107.654365	77	1.39810864	F(2, 77)	= 67.80
				Prob > F	= 0.0000
				R-squared	= 0.6378
				Adj R-squared	= 0.6284
Total	297.224375	79	3.76233386	Root MSE	= 1.1824

y	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
x	1.158722	.1163415	9.96	0.000	.9270567	1.390388
z	.9722231	.1362012	7.14	0.000	.7010119	1.243434
_cons	-.1121836	.1341773	-0.84	0.406	-.3793648	.1549976

By default, `gsreg` outcomes were saved as `gsreg.dta`, and the "best" model was selected based on the adjusted-R^2. The `outsample(20)` option keeps the last 20 observations to forecast evaluations. It also calculates, by default, the in-sample and the out-of-sample root mean squared errors.

Following table 2 below, the best model for in-sample criteria (adjusted-R^2 or root mean squared error) is the worst in terms of the out-of-sample root mean squared error criterion (model 3, with x and z as covariates). On the contrary, model 1 (which only includes x as a covariate) has a relatively poor in-sample performance but ensures the highest forecast accuracy. By alternatively selecting, for example, `rmse_out` or `rmse_in` as ranking variables, `gsreg` users can exhaustively cross-check model optimality.

Table 2. Example 2: Main stored results

Order	Model	r_sqr_a	rmse_in	rmse_out
1	x	0.3904075	1.5144274	0.5720016
2	z	0.1605745	1.7771323	0.7596292
3	x z	0.6283932	1.1824164	0.7670928

5.3 Parameter stability analysis

By generating a database with exhaustive information about all regression alternatives, `gsreg` is a unique tool for parameter stability analysis. In this example, we will use `crisis_fr.dta` of Gluzmann and Guzman (2011) (containing information on financial crisis, financial reforms, and a set of controls for 89 countries from 1973–2005) to evaluate interest-parameter stability under alternative control variable structures. As a first step, we run a pooled-data linear regression (for Latin American countries and emerging Asian economies in transition) of the probability of future financial crisis over the next five years (`fc5`) on a financial reform index (`ifr`) and its recent change (`d_ifr`).

```
. use crisis_fr.dta, clear
. regress fc5 ifr d_ifr if EA_LA_TR==1
```

Source	SS	df	MS		
Model	5.08852674	2	2.54426337	Number of obs	= 928
Residual	168.410396	925	.182065293	F(2, 925)	= 13.97
				Prob > F	= 0.0000
Total	173.498922	927	.187161729	R-squared	= 0.0293
				Adj R-squared	= 0.0272
				Root MSE	= .42669

fc5	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
ifr	-.00711	.0025616	-2.78	0.006	-.0121372	-.0020829
d_ifr	.0549173	.0110602	4.97	0.000	.0332113	.0766233
_cons	.2793652	.0255525	10.93	0.000	.2292176	.3295128

From the above regression, we obtain a negative and significant relationship between `fc5` and `ifr` and a positive (and even more significant) regression coefficient for `d_ifr`.

In their article, Gluzmann and Guzman (2011) also identify 23 theoretically relevant control variables to consider (`v1` to `v23`). Unlike previous examples, we will not use `gsreg` here to obtain the best model (for example, best control variable structure) in terms of some in-sample or out-of-sample information criteria (or some linear combination of many information criteria); we will instead examine the whole set of results to evaluate `ifr` and `d_ifr` regression coefficient and t statistic distributions.

With this purpose, we follow the Levine and Renelt (1992) and Sala-I-Martin (1997) approach and run all possible regressions using available information in `crisis_fr.dta`, taking `ifr` and `d_ifr` as fixed variables and forcing `gsreg` to use three control variables for each alternative.

```
. use crisis_fr.dta, clear
. gsreg fc5 v1-v23 if EA_LA_TR ==1, ncomb(3) fixvar(ifr d_ifr) replace nocount
-------------------------------------------------
Total Number of Estimations: 1771
-------------------------------------------------
Computing combinations...
Preparing regression list...
Doing regressions...
Saving results...
file gsreg.dta saved
```

(*output omitted*)

`gsreg` execution takes less than a minute using Stata/MP 12.1 for Windows (64 bit) in a laptop with an Intel i7-3520m processor and 4 GB of DDR3 RAM memory. The `fixvar(ifr d_ifr)` option ensures that `ifr` and `d_ifr` will be used as covariates in all regressions. The `ncomb(3)` option reduces the search space to all possible combinations (without repetition) of 23 control variables, taken 3 at time. Main command outcomes can easily be described using the following kernel density plot:

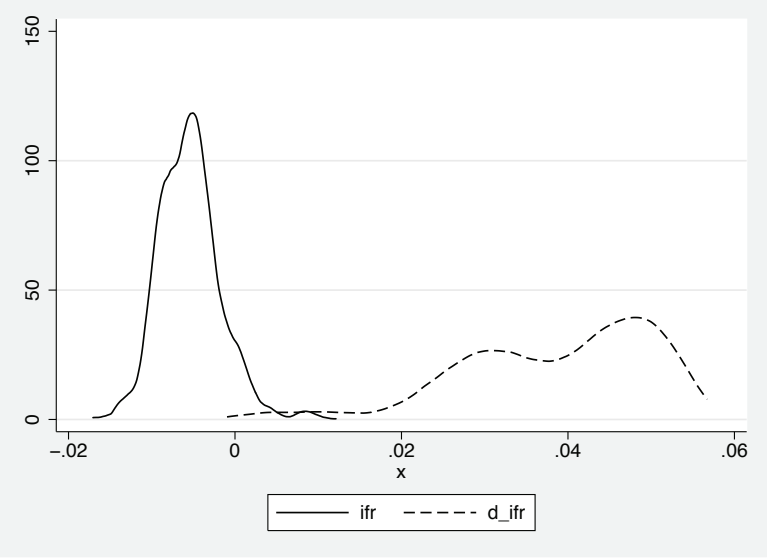

Figure 5. Example 3: `ifr` and `d_ifr` coefficient distribution

From figure 5, we can see that a large share of the `ifr` coefficient distribution is concentrated around 0, while `d_ifr` coefficients are almost exclusively distributed over positive (nonzero) values. To provide users with an enlarged example, we can replicate the analysis using alternative estimation methods, such as

```
. gsreg fc5 v1-v23 if EA_LA_TR ==1, ncomb(3) fixvar(ifr d_ifr) replace nocount
> cmdest(probit) cmdstat(r2_p ll)
```

(*output omitted*)

or

```
. gsreg fc5 v1-v23 if EA_LA_TR ==1, ncomb(3) fixvar(ifr d_ifr) replace nocount
> cmdest(xtreg) cmdoptions(fe robust)
```

(*output omitted*)

For the `probit` (pooled) case, `gsreg` additionally computes and saves the pseudo-R^2 and the log likelihood of each regression. Here the execution time (on the same software and hardware) rose to 13 minutes.

Finally, the `xtreg` version was used to estimate the same relationship using fixed effects and robust standard errors. The execution time of the same exercise was about 11 minutes.

6 Stored results

The `gsreg` command creates a `.dta` file with outcome information for all estimated alternatives. By default, it includes the following columns for each regression:

1. regression ID (variable `order`),
2. covariate regression coefficients (named `v_1_b`, `v_2_b`, ... and labeled with the full covariate name plus the word `coeff.`),
3. coefficient t statistics (named `v_1_t`, `v_2_t`, ... and labeled with the full covariate name plus the word `tstat.`),
4. number of observations (variable `obs`),
5. number of covariates (variable `nvar`),
6. adjusted-R^2 (variable `r_sqr_a`),
7. in-sample root mean squared error (variable `rmse_in`),
8. normalized linear combination of user-selected and weighted model-selection criteria (as `nindex()` or `mindex()` if this option is specified)
9. additional user-specified statistics (if option `cmdstat()` is specified),
10. out-of-sample root mean squared error (if option `outsample()` is specified), and
11. residual test statistics (if specified).

When the `compact` option is specified, regression coefficients and t statistics are omitted and replaced by a unique summary string variable as described in section 4.2.

Also `gsreg` displays the best regression in terms of the user-specified `nindex()` or `mindex()` (or, if these options are not specified, the adjusted-R^2). Therefore, all the "best model" results are also stored in memory (as scalars, macros, matrices, and functions).

7 Acknowledgments

`gsreg` is based on `FUERZA_BRUTA.do`, a former Stata do-file originally developed by Demian Panigo and subsequently enhanced by Diego Herrero (University of Buenos Aires, Argentina) and Pablo Gluzmann. Usual disclaimer applies.

This work was supported by the National Agency for Science and Technology Promotion in Argentina (PICT 2010/2719) and the Argentine National Council of Scientific and Technological Research.

We thank Amalia Torija-Zane, Diego Herrero, Fernando Toledo, and Martín Guzmán for their valuable suggestions that helped us to improve the quality of `gsreg`.

8 References

Ashley, R., C. W. J. Granger, and R. Schmalensee. 1980. Advertising and aggregate consumption: An analysis of causality. *Econometrica* 48: 1149–1167.

Berk, K. N. 1978. Comparing subset regression procedures. *Technometrics* 20: 1–6.

Box, G. E. P., and P. Newbold. 1971. Some comments on a paper of Coen, Gomme and Kendall. *Journal of the Royal Statistical Society, Series A* 134: 229–240.

Breiman, L. 1995. Better subset regression using the nonnegative garrote. *Technometrics* 37: 373–384.

Caldwell, B. J. 1992. Friedman's predictivist instrumentalism: A modification. *Research in the History of Economic Thought and Methodology* 10: 119–128.

Castle, J. L. 2006. Empirical modelling and model selection for forecasting inflation in a non-stationary world. PhD thesis, Nuffield College, University of Oxford.

Coen, P. J., E. D. Gomme, and M. G. Kendall. 1969. Lagged relationships in economic forecasting. *Journal of the Royal Statistical Society, Series A* 132: 133–163.

Davidson, J. E., and D. F. Hendry. 1981. Interpreting econometric evidence: The behaviour of consumers' expenditure in the UK. *European Economic Review* 16: 177–192.

Derksen, S., and H. J. Keselman. 1992. Backward, forward and stepwise automated subset selection algorithms: Frequency of obtaining authentic and noise variables. *British Journal of Mathematical and Statistical Psychology* 45: 265–282.

Doornik, J. A. 2009. Autometrics. In *The Methodology and Practice of Econometrics: A Festschrift in Honour of David F. Hendry*, ed. J. L. Castle and N. Shephard, 88–121. Oxford: Oxford University Press.

Draper, N. R., and H. Smith. 1998. *Applied Regression Analysis*. 3rd ed. New York: Wiley.

Duarte Silva, A. P. 2009. Exact and heuristic algorithms for variable selection: Extended leaps and bounds. Working Paper No. 01/2009, Facultade de Economia e Gestão, Universidade Católica Portuguesa.

Efron, B., T. Hastie, I. Johnstone, and R. Tibshirani. 2004. Least angle regression. *Annals of Statistics* 32: 407–499.

Efroymson, M. A. 1960. Multiple regression analysis. In *Mathematical Methods for Digital Computers*, ed. A. Ralston and H. S. Wilf, 191–203. New York: Wiley.

Forrest, S., and M. Mitchell. 1991. The performance of genetic algorithms on Walsh polynomials: Some anomalous results and their explanation. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, ed. R. K. Belew and L. B. Booker, 182–189. San Francisco: Morgan Kaufmann.

Friedman, M. 1953. *Essays in Positive Economics*. Chicago: Chicago University Press.

Frisch, R. 1934. *Statistical Confluence Analysis by Means of Complete Regression Systems*. Oslo: University Institute of Economics.

Furnival, G. M., and R. W. Wilson. 1974. Regressions by leaps and bounds. *Technometrics* 16: 499–511.

Gatu, C., and E. J. Kontoghiorghes. 2006. Branch-and-bound algorithms for computing the best-subset regression models. *Journal of Computational and Graphical Statistics* 15: 139–156.

Gilbert, C. L. 1986. Practitioners' corner: Professor Hendry's econometric methodology. *Oxford Bulletin of Economics and Statistics* 48: 283–307.

Gluzmann, P., and M. Guzman. 2011. Reformas financieras e inestabilidad financiera. *Ensayos Económicos, BCRA* 61-62: 35–73.

Haavelmo, T. 1944. The probability approach in econometrics. *Econometrica* 12: iii–vi and 1–115.

Hendry, D. F. 1980. Econometrics—alchemy or science? *Economica* 47: 387–406.

———. 1995. *Dynamic Econometrics*. Oxford: Oxford University Press.

Hendry, D. F., and M. S. Morgan. 1995. *The Foundations of Econometric Analysis*. Cambridge: Cambridge University Press.

Herwartz, H. 2007. A note on model selection in (time series) regression models—general-to-specific or specific-to-general? Working Paper No. 09/2007, Department of Economics, Christian-Albrechts-Univesität Kiel.

Koopmans, T. C. 1947. Measurement without theory. *Review of Economics and Statistics* 29: 161–172.

Krolzig, H.-M., and D. F. Hendry. 2001. Computer automation of general-to-specific model selection procedures. *Journal of Economic Dynamics and Control* 25: 831–866.

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

Levine, R., and D. Renelt. 1992. A sensitivity analysis of cross-country growth regressions. *American Economic Review* 82: 942–963.

Lindsey, C., and S. Sheather. 2010. Variable selection in linear regression. *Stata Journal* 10: 650–669.

Lovell, M. C. 1983. Data mining. *Review of Economics and Statistics* 65: 1–12.

Mäki, U. 1986. Rhetoric at the expense of coherence: A reinterpretation of Milton Friedman's methodology. *Research in the History of Economic Thought and Methodology* 4: 127–143.

Marinucci, M. 2008. Automatic prediction and model selection. PhD thesis, Facultad de Ciencias Económicas y Empresariales, Universidad Complutense de Madrid.

Miller, A. J. 1984. Selection of subsets of regression variables. *Journal of the Royal Statistical Society, Series A* 147: 389–425.

Pagan, A. 1987. Three econometric methodologies: A critical appraisal. *Journal of Economic Surveys* 1: 3–23.

Pérez-Amaral, T., G. M. Gallo, and H. White. 2003. A flexible tool for model building: The relevant transformation of the inputs network approach (RETINA). *Oxford Bulletin of Economics and Statistics* 65: 821–838.

Sala-I-Martin, X. X. 1997. I just ran two million regressions. *American Economic Review* 87: 178–183.

Sheather, S. J. 2009. *A Modern Approach to Regression with R*. New York: Springer.

Tibshirani, R. 1996. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society, Series B* 58: 267–288.

About the authors

Pablo Gluzmann has a PhD in economics (National University of La Plata) and is a researcher at the Argentine National Council of Scientific and Technological Research. He is a senior researcher at the Center for Distributive, Labor and Social Studies, and he is an assistant professor of advanced macroeconomics in the Department of Economics in the Faculty of Economic Sciences at the National University of La Plata in Argentina.

Demian Panigo has a PhD in economics (EHESS in Paris) and is Director of the Center of Workers Innovation at the Argentine National Council of Scientific and Technological Research. He is a professor of advanced macroeconomics at the Department of Economics in the Faculty of Economic Sciences at the National University of La Plata and the National University of Moreno, both in Argentina.