

General-to-specific modeling in Stata

Damian Clarke
Department of Economics
University of Oxford
Oxford, UK
damian.clarke@economics.ox.ac.uk

Abstract. Empirical researchers are frequently confronted with issues regarding which explanatory variables to include in their models. This article describes the application of a well-known model-selection algorithm to Stata: general-to-specific (GETS) modeling. This process provides a prescriptive and defendable way of selecting a few relevant variables from a large list of potentially important variables when fitting a regression model. Several empirical issues in GETS modeling are then discussed, specifically, how such an algorithm can be applied to estimations based upon cross-sectional, time-series, and panel data. A command is presented, written in Stata and Mata, that implements this algorithm for various data types in a flexible way. This command is based on Stata’s `regress` or `xtreg` command, so it is suitable for researchers in the broad range of fields where regression analysis is used. Finally, the `genspec` command is illustrated using data from applied studies of GETS modeling with Monte Carlo simulation. It is shown to perform as empirically predicted and to have good size and power (or gauge and potency) properties under simulation.

Keywords: st0365, `genspec`, model selection, general to specific, statistical analysis, specification tests

1 Introduction

A common problem facing the applied statistical researcher is that of restricting her or his models to include the appropriate subset of variables from the real world. This is particularly the case in regression analysis, where the researcher has a determined dependent variable y but can (theoretically) include any number of explanatory variables X in the analysis of y . Sometimes, the researcher can invoke a theory that provides guidance about what an appropriate set of X variables may be. However, at other times, an overarching theory may be absent or may fail to prescribe a parsimonious set of variables. In this situation, the researcher is confronted by issues of model selection: Of all the variables that could be important, which should be included in the final regression model?

Econometric theory expounds on this and can offer useful guidance to all classes of applied statistical researchers—both economists and noneconomists alike. One example of such guidance concerns the general-to-specific or general-to-simple (GETS) modeling procedure. GETS is a prescriptive way to select a parsimonious and instructive final model from a large set of real-world variables and enables the researcher to avoid unnecessary ambiguity or ad hoc decisions. This process involves the definition of a general

model that contains all potentially important variables and then, via a series of step-wise statistical tests, the removal of empirically “unimportant” variables to arrive at the proposed specific or final model.

There is a considerable amount of literature on the theoretical merits and drawbacks of such a process of model selection. Hendry and coauthors (see, for example, [Krolzig and Hendry \[2001\]](#); [Campos, Ericsson, and Hendry \[2005\]](#); [Hendry and Krolzig \[2005\]](#); and references therein) have various articles defining aspects of the GETS estimation procedure and its properties. Applications of GETS are common in analyses of economic growth ([Hendry and Krolzig 2004](#)), consumption ([Hoover and Perez 1999](#); [Campos and Ericsson 1999](#)), and various phenomena in the noneconomic literature ([Succarat and Escribano 2012](#); [Cairns et al. 2011](#)).

GETS modeling is driven by a large group of variables¹ and a series of statistical tests based on subsets of these models. The outcome of a GETS search process is a specific model that is consistent with necessary properties for valid inference and that contains all the statistically significant variables from the initial large set. In this sense, model selection is based upon the observed data and the results of the tests on these data. Such “data-driven” model selection is not without its critics. Both philosophical ([Kennedy 2002a,b](#)) and statistical ([Harrell 2001](#)) critiques have been levied against this approach, with suggestions that it may result in the underestimation of confidence intervals and *p*-values and should entail a penalty in terms of degrees of freedom lost.

Despite these critiques, significant arguments can be, and have been, made in favor of a GETS modeling process.² Particularly, it appears to perform very well in recovering the true data-generating process (DGP) in Monte Carlo experiments ([Hoover and Perez 1999](#)). For this reason, in this article, I introduce GETS modeling and the corresponding `genspec` statistical routine as an addition to the applied researcher’s toolkit in Stata. This tool is similar to what already exists in other languages such as R and OxMetrix, and it is a useful extension to Stata’s functionality. As will be shown, `genspec` performs as empirically expected and does a good job in recovering the true underlying model in benchmark Monte Carlo simulations.

1. Typically, this consists of all potentially important independent variables that the researcher can include, along with nonlinearities and lagged dependent and independent variables.

2. In the remainder of this article, I (purposely) avoid discussions of the merits and drawbacks of this routine and instead focus on how researchers can implement such a process if they deem it desirable and useful in their specific context. A long line of literature including counterarguments to the above concerns exists (see, for example, [Hansen \[1996\]](#), who provides a balanced introduction), and the interested reader is directed to these resources.

The `genspec` command, as well as the GETS statistical routine in general, is designed with regression analysis in mind. For this reason, `genspec` is based on Stata's `regress` (or `xtreg`) command. When moving from a series of potential explanatory variables to one final specific model, `genspec` runs a number of stepwise regressions, with the subsequent testing and removal of insignificant variables. This routine is defined in a flexible way to make it functional in a range of modeling situations. It can be used with cross-sectional, panel, and time-series data, and it works with Stata functions that are appropriate in models of these types. It places no limitations on arbitrary misspecification of models, allowing such features as clustered standard errors, robust standard errors, and bootstrap- and jackknife-based estimation.

To define an algorithm that is appropriate for a range of very different underlying models, a researcher must make several decisions. GETS modeling requires that the preliminary general model be subjected to a range of prespecification tests to ensure that it complies with the modeling assumptions upon which estimation is based. These assumptions, and indeed the resulting tests, vary by the type of regression model in which a researcher is interested. In the following section, I define and discuss the appropriate tests to run in a range of situations, and I discuss how to select between competing final models in different circumstances.

To illustrate the performance of `genspec`, we take a preexisting benchmark in GETS modeling (Hoover and Perez 1999) and show that similar performance can be achieved in Stata. These results suggest that GETS modeling and the user-written `genspec` command may be useful to Stata users interested in defining appropriate, flexible, and data-driven economic models.

2 Algorithm description

As alluded to before, GETS modeling requires an initial group of variables, runs a series of regressions and automated tests, and provides the researcher with a final specific model. This initial group of variables provided by the researcher is referred to as the general unrestricted model (GUM) and should contain all potentially important independent variables. Before beginning analysis, the `genspec` algorithm tests the GUM for validity via a series of statistical tests (described later); if the GUM is valid, a regression is run, with the stepwise removal of the variable with the lowest t statistic. At each step of the process (or "search path"), a prospective final (or terminal) specification is produced with the true terminal specification found when no insignificant variables remain in the current regression model. A comprehensive description of the GETS search process is provided at the end of this section.

The search algorithm undertaken by `genspec` depends upon the model type and GUM specified by the user. Whether the underlying model is based upon cross-sectional, time-series, or panel data determines the set of initial tests (henceforth, "the battery") and the set of subsequent tests run at each stage of the search path. In what follows, I discuss the general search algorithm followed for every model, delaying discussion of specific tests until the corresponding subsections for cross-sectional, time series, and panel models.

In defining the search algorithm, we follow the one described in Hoover and Perez (1999) and in appendix A of Hoover and Perez (2004). Hoover and Perez (1999) is considered an important starting point in the description of a computational GETS modeling process (see, for example, Campos, Ericsson, and Hendry [2005]) and a valid description of the nature of GETS modeling. The algorithm implemented in Stata takes the following form:

1. The user specifies her or his proposed GUM and indicates the relevant data to Stata, using `if` and `in` qualifiers if necessary.
2. Of the full sample, 90% is retained, while the remaining 10% is set aside for out-of-sample testing. The battery of tests is run on this 90% sample at the nominal size.³ If one of these tests is failed, it is eliminated from the battery in the following steps of the search path. If more than one of these tests is failed by the GUM, the user is instructed that the GUM is likely a poor representation of the true model and an alternative general model is requested.⁴
3. Each variable in the general model is ranked by the size of its t statistic, and the algorithm then follows m (by default, five) search paths. The first search path is initiated by eliminating the variable with the lowest (insignificant) t statistic from the GUM. The second follows the same process, but rather than eliminating the lowest, it eliminates the second lowest. This process is followed until reaching the m th search path that eliminates the m th-lowest variable. For each search path, the current specification then includes all remaining variables, and this specification is estimated by regression.
4. The current specification is then subjected to the full battery of tests, along with an F test, to determine whether the current specification is a valid restriction of the GUM. If any of these tests fails, the current search path is abandoned, and the algorithm jumps to the subsequent search path.
5. If the current specification passes the above tests, the variables in the current specification are once again ordered by the size of their t statistics, and the variable with the next-lowest t statistic is eliminated. This then becomes a potential current specification, which is subjected to the battery of tests. If any of these tests fails, the model reverts to the previous current specification, and the variable with the second-lowest (insignificant) t statistic is eliminated. Such a process is followed until a variable is successfully eliminated or until all insignificant variables have been attempted. If an insignificant variable is eliminated, stage 5 is restarted with the current specification. This process is followed iteratively until either all insignificant variables have been eliminated or no more variables can be successfully removed.

3. In sections 2.1–2.3, I discuss the specific nature of these tests and the determination of the in-sample and out-of-sample observations.

4. As in all terminal decisions, the user can override this decision and continue with her or his proposed GUM if so desired.

6. Once no further variables can be eliminated, a potential terminal specification is reached. This specification is estimated using the full sample of data. If all variables are significant, it is accepted as the terminal specification. If any insignificant variables remain, these are eliminated as a group, and the new terminal specification is subjected to the battery of tests. If it passes these tests, it is the terminal specification; if it does not, the previous terminal specification is accepted.
7. Each of the m terminal specifications is compared, and if these are different, the final specification is determined using encompassing or an information criterion (see the related discussion in sections 2.1–2.3).

2.1 Cross-sectional models

Cross-sectional models are subjected to an initial battery of five tests: a Doornik–Hansen test for normality of errors, the [Breusch and Pagan \(1979\)](#) test for homoskedasticity of errors,⁵ the Ramsey regression equation specification error test for the linearity of coefficients ([Ramsey 1969](#)), and an in-sample and out-of-sample stability F test. These two final tests consist of a comparison of regressions of each subsample with estimation results for the full sample: in the in-sample test, the two subsamples are composed of two halves of the full sample, while in the out-of-sample test, a comparison is made between the 90% and 10% samples. These tests are analogous to [Chow \(1960\)](#) tests.

Information criteria are used to determine the final model based on ordinary least squares with cross-sectional data. For each of the m potential terminal specifications, a regression is run, and the Bayesian information criterion (BIC) is calculated. The terminal specification that has the lowest BIC is determined to be the final specification.

2.2 Time-series models

In time-series models, an additional test is included in the battery discussed above: a test is run for autocorrelated conditional heteroskedasticity up to the second order ([Engle 1982](#)). To partition the sample into in sample and out of sample, a researcher discards the final 10% of observations to be used in out-of-sample tests. These are (as in all cases) returned to the sample in the calculation of the final model, and a BIC is once again used to choose between terminal specifications.

2.3 Panel-data models

Given the nature of panel data, the initial battery of tests here potentially includes two tests omitted in cross-sectional or time-series models. The first of these is a test for serial correlation of the idiosyncratic portion of the error term (discussed by [Wooldridge \[2010\]](#) and implemented for Stata by [Drukker \[2003\]](#)). The second is a Lagrange multiplier

5. This test is not run if the fitted model is robust to this type of misspecification.

test for random effects (given that a random-effects model is specified), which tests the validity of said model (Breusch and Pagan 1980). Along with these tests, a Doornik–Hansen-type test for normality of the idiosyncratic portion of the error term and both in-sample and out-of-sample Chow tests (as previously discussed) are estimated.

To determine the final specification from the resulting m potential terminal specifications, the algorithm uses an encompassing procedure. Each variable included in at least one terminal specification is included in the potential terminal model. This model is then tested according to step 6 of the algorithm listed in section 2.

3 The genspec command

3.1 Syntax

The syntax of the `genspec` command is as follows:

```
genspec depvar indepvars [ if ] [ in ] [ weight ] [ , vce(vcetype) xt(re|fe|be)
    ts nodiagnostic tlimit(#) numsearch(#) nopartition noserial
    verbose ]
```

Here *depvar* refers to the dependent variable in the general model, and *indepvars* refers to the full set of independent variables to be tested for inclusion in the final model.

3.2 Options

`vce(vcetype)` determines the type of standard error reported in the fitted regression model and allows standard errors that are robust to certain types of misspecification. *vcetype* may be `robust`, `cluster` *clustvar*, `bootstrap`, or `jackknife`.

`xt(re|fe|be)` specifies that the model is based on panel data. Users must specify whether they wish to fit a random-effects (`re`), fixed-effects (`fe`), or between-effects (`be`) model. `xtset` must be specified before using this option.

`ts` specifies that the model is based on time-series data. `tsset` must be specified before using this option, and if `tsset` is specified, time-series operators may be used.

`nodiagnostic` turns off the initial diagnostic tests for model misspecification. This should be used with caution.

`tlimit(#)` sets the critical t value above which variables are considered as important in the terminal specification. The default is `tlimit(1.96)`.

`numsearch(#)` defines the number of search paths to follow in the model. The default is `numsearch(5)`. If a large dataset is used, fewer search paths may be preferred to reduce computational time.

`nopartition` uses the full sample of data in all search paths and does not engage in out-of-sample testing.

`noserial` requests that no serial correlation test be performed if panel data are used. This option should be specified with the `xt` option only.

`verbose` requests full program output of each search path explored.

3.3 Stored results

`genspec` stores the following in `e()`:

Scalars	
<code>e(fit)</code>	BIC of final specification
Macros	
<code>e(genspec)</code>	list of variables from the final specification

The full `ereturn` list, which includes regression results for the terminal specification, is available by typing `ereturn list`.

4 Performance

4.1 An example with empirical data

To illustrate the performance of `genspec`, we use empirical data from a well-known applied study of GETS modeling. [Hoover and Perez \(1999\)](#), using data from [Lovell \(1983\)](#), illustrate that GETS modeling can work well in recovering the true DGP in empirical applications, even when prospective variables are multicollinear. We use the [Hoover and Perez \(1999\)](#) dataset in the example below. A brief description of the source and nature of the data is provided in data appendix [A](#).

We use their model 5 to provide an example of the functionality of `genspec`. As described in table [2](#), the dependent variable in model 5 is generated according to

$$y_{5t} = -0.046 \times \text{ggeq}_t + 0.11 \times u_t$$

In the following Stata excerpt, we see that after loading the dataset and defining the full set of candidate variables (first and second lags of all independent variables and first to fourth lags of y_{5t}), the `genspec` algorithm searches and returns a model with only one independent variable. As desired, this final model is the true DGP, with slight sampling variation in the coefficient on `ggeq` due to the relatively small sample size.

However, `genspec` raises one warning: here the GUM does not pass the full battery of defined tests. Specifically, the GUM fails the in-sample Chow test, which suggests that the coefficients estimated over the first half of the series are statistically different

from those estimated over the second half. While this may indicate a structural break signaling that the GUM may not be an appropriate model, `genspec` respects the GUM entered by the user and continues to search for (and find) the true model.

```
. use genspec_data
(Hoover and Perez (1999) data for use in GETS modelling)
. quietly ds y* u* time, not
. local xvars `r(varlist)'
. local lags 1.dcoinc 1.gd 1.ggeq 1.ggfeq 1.ggfr 1.gnpq 1.gydq 1.gpiq 1.fmrра
> 1.fmbase 1.fm1dq 1.fm2dq 1.fsdj 1.fyaaac 1.lhc 1.lhur 1.mu 1.mo
. genspec y5 `xvars' `lags' 1.y5 12.y5 13.y5 14.y5, ts
# of observations is > 10% of sample size. Will not run out-of-sample tests.
The in-sample Chow test rejects equality of coefficients
Respecify using nodiagnostic if you wish to continue without specification
tests. This option should be used with caution.

The GUM fails 1 of 4 misspecification tests. Doornik-Hansen test for normality
of errors not rejected. The presence of (1 and 2 order) ARCH components is
rejected. Breusch-Pagan test for homoscedasticity of errors not rejected.

Specific Model:

```

Source	SS	df	MS	Number of obs = 143
Model	23.6849853	1	23.6849853	F(1, 141) = 1966.22
Residual	1.69848221	141	.012045973	Prob > F = 0.0000
Total	25.3834675	142	.178756814	R-squared = 0.9331
				Adj R-squared = 0.9326
				Root MSE = .10975

y5	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
ggeq	-.0463615	.0010455	-44.34	0.000	-.0484284 -.0442945
_cons	-.0042157	.0091781	-0.46	0.647	-.0223602 .0139289

4.2 Monte Carlo simulation

The previous example suggests that a GETS algorithm performed well in this particular case. However, to be confident in the functionality of the `genspec` command, we are interested in testing whether this performs as empirically expected over a larger range of models and circumstances. For this reason, we run a set of Monte Carlo simulations based upon the empirical data described above and in data appendix A. The reason we test the performance of `genspec` on these data is twofold. First, the highly multicollinear nature of many of these variables makes recovering the true DGP a challenge for automated search algorithms. Second, and fundamentally, there is already a benchmark performance test of how a GETS algorithm should work on the data available in Hoover and Perez's (1999) results.

The Monte Carlo simulation is designed as follows. We draw a normally distributed random variable for use as the u term described in table 2. Using this draw u^D (where superscript D denotes simulated data), we generate the corresponding u^* (u^{*D}); then, combining u^D , u^{*D} , and the true macroeconomic variables, we simulate each of our nine different outcome variables y_1^D, \dots, y_9^D outlined in the data appendix. Once we have one

simulation for each dependent variable, we run `genspec` with the 40 candidate variables and determine whether the true DGP is recovered. This process makes up one simulation. We then repeat this 1,000 times, observing in each case whether `genspec` identifies the true model and, if not, how many of the true variables are correctly included and how many false variables are incorrectly included.

To determine the performance of the search algorithm, we compare the performance of `genspec` with that of the benchmark performance described in table 7 of [Hoover and Perez \(1999\)](#). We focus on two important summary statistics: gauge and potency. Gauge refers to the percent of irrelevant variables in the final model (regardless of whether they are significant or not). The gauge shows the frequency of type I errors in the search algorithm and is analogous to power in typical statistical tests. The potency of our model refers to the percent of relevant variables in our final model ([Castle, Doornik, and Hendry 2012](#)). We would hope in most searches that potency is approximately 100% because the final model should at the very least not discard true variables. We would prefer to have a higher gauge (and more irrelevant—and perhaps insignificant—variables) if this implies that the final model includes all true variables.

[Table 1](#) presents the performance of the `genspec` search algorithm and compares this with the benchmark levels expected. In each case, we see that `genspec` performs approximately identically to Hoover and Perez's ([1999](#)) empirical observations. Fundamentally, the potency of `genspec` is identical to that expected with these data, which suggests that the search algorithm performs as expected in identifying true variables. We do see, however, that `genspec` is more likely to incorrectly include false variables because it has a higher gauge than benchmark performance. This is likely due to a slight difference in the battery of tests in `genspec` compared with that of Hoover and Perez's ([1999](#)) algorithm. In `genspec`, by default, the critical value for the battery of tests is set at 5%: this increases the likelihood that a specific test is retained for the full search path. In the simulations below, [Hoover and Perez \(1999\)](#) report results for a critical value of 1% in the battery of tests, while the `genspec` algorithm reports results at 5% (and 1% for the critical t -value when eliminating irrelevant variables).

Table 1. Performance of *genspec* in Monte Carlo simulation

	1	2	3	4	5	Models	6	7	8	9	Average
Panel A: Algorithm performance											
Average rate of inclusion of											
True variables	N/A	1.00	1.89	1.00	1.00	1.01	3.00	2.95	2.33	—	
False variables	0.24	1.42	0.64	0.51	0.55	0.62	2.29	0.73	1.39	0.93	
Gauge	0.6%	3.6%	1.6%	1.3%	1.4%	1.5%	5.7%	1.8%	3.5%	2.3%	
Potency	N/A	100.0%	94.5%	100.0%	100.0%	50.1%	100.0%	98.2%	46.6%	87.7%	
Panel B: Benchmark performance											
Average rate of inclusion of											
True variables	N/A	1.00	1.89	0.99	1.00	1.01	2.82	3.00	2.86	—	
False variables	0.29	2.31	0.39	0.34	0.32	0.26	1.23	0.38	1.20	0.75	
Gauge	0.7%	5.7%	0.9%	0.8%	0.7%	0.6%	3.0%	0.9%	3.2%	1.8%	
Potency	N/A	100.0%	94.7%	99.9%	100.0%	50.3%	94.0%	99.9%	57.3%	87.0%	

Notes: Panel A shows the performance of the user algorithm written for Stata *genspec*, while panel B shows the benchmark algorithm of Hoover and Perez (1999), who simulate using the same data (see their table 7 for original results). Results from each panel are from 1,000 simulations with a 2-tail critical value of 1%. The DGP for each model is described in the data appendix of this article, and each model includes a constant that is ignored when calculating the gauge and scope. Full code and simulation results for replication are available at <https://sites.google.com/site/damianclarke/research>.

5 Conclusion

Applied researchers are often faced with determining the appropriate set of independent variables to include in an analysis when examining a given outcome variable. This process of model selection can have important implications on the results of a given research agenda, even when the research question and methodology have been set. General-to-specific modeling offers a researcher a prescriptive, defendable, and data-driven way to resolve this issue. Although this methodology has been drawn from a considerable amount of econometric literature, nothing suggests that it should not be used by all classes of researchers interested in regression analysis.

In this article, I introduce the `genspec` command to Stata. It shows that this command behaves as empirically expected and is successful in recovering the true model when given a large set of potential variables to choose from. Such a modeling technique offers important benefits to a range of users who are interested in identifying an underlying model while remaining relatively agnostic or placing few restrictions on their general theory.

The `genspec` command is flexible, allowing the user to choose from a wide array of models using either time-series, panel, or cross-sectional data. I also discuss several empirical considerations in developing such an algorithm, in particular, the nature of the tests desired when examining the proposed general model and how to deal with model selection when choosing between multiple models.

6 Acknowledgments

Financial support from the National Commission for Scientific and Technological Research of the Government of Chile is gratefully acknowledged. I thank Bent Nielsen, Marta Dormal, George Vega Yon, and Nicolas Van de Sijpe for useful comments at various stages in the writing of this command and article. I also acknowledge H. Joseph Newton and an anonymous *Stata Journal* referee for valuable comments and help. This routine nests the `xtserial` command, which was written for Stata by David Drukker. All remaining errors and omissions are my own.

7 References

- Breusch, T. S., and A. R. Pagan. 1979. A simple test for heteroscedasticity and random coefficient variation. *Econometrica* 47: 1287–1294.
- . 1980. The Lagrange multiplier test and its applications to model specification in econometrics. *Review of Economic Studies* 47: 239–253.
- Cairns, A. J. G., D. Blake, K. Dowd, G. D. Coughlan, D. Epstein, and M. Khalaf-Allah. 2011. Mortality density forecasts: An analysis of six stochastic mortality models. *Insurance: Mathematics and Economics* 48: 355–367.

- Campos, J., and N. R. Ericsson. 1999. Constructive data mining: Modeling consumers' expenditure in Venezuela. *Econometrics Journal* 2: 226–240.
- Campos, J., N. R. Ericsson, and D. F. Hendry. 2005. General-to-specific modeling: An overview and selected bibliography. International Finance Discussion Papers 838, Board of Governors of the Federal Reserve System.
- Castle, J. L., J. A. Doornik, and D. F. Hendry. 2012. Model selection when there are multiple breaks. *Journal of Econometrics* 169: 239–246.
- Chow, G. C. 1960. Tests of equality between sets of coefficients in two linear regressions. *Econometrica* 28: 591–605.
- Drukker, D. M. 2003. Testing for serial correlation in linear panel-data models. *Stata Journal* 3: 168–177.
- Engle, R. F. 1982. Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica* 50: 987–1007.
- Hansen, B. E. 1996. Methodology: Alchemy or science: Review article. *Economic Journal* 106: 1398–1413.
- Harrell, F. E., Jr. 2001. *Regression Modeling Strategies: With Applications to Linear Models, Logistic Regression, and Survival Analysis*. New York: Springer.
- Hendry, D. F., and H.-M. Krolzig. 2004. We ran one regression. *Oxford Bulletin of Economics and Statistics* 66: 799–810.
- . 2005. The properties of automatic GETS modelling. *Economic Journal* 115: C32–C61.
- Hoover, K. D., and S. J. Perez. 1999. Data mining reconsidered: Encompassing and the general-to-specific approach to specification search. *Econometrics Journal* 2: 167–191.
- . 2004. Truth and robustness in cross-country growth regressions. *Oxford Bulletin of Economics and Statistics* 66: 765–798.
- Kennedy, P. E. 2002a. Reply. *Journal of Economic Surveys* 16: 615–620.
- . 2002b. Sinning in the basement: What are the rules? The ten commandments of applied econometrics. *Journal of Economic Surveys* 16: 569–589.
- 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.
- Lovell, M. C. 1983. Data mining. *Review of Economics and Statistics* 65: 1–12.
- Ramsey, J. B. 1969. Tests for specification errors in classical linear least-squares regression analysis. *Journal of the Royal Statistical Society, Series B* 31: 350–371.

- Sucarrat, G., and A. Escrivano. 2012. Automated model selection in finance: General-to-specific modelling of the mean and volatility specifications. *Oxford Bulletin of Economics and Statistics* 74: 716–735.
- Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

About the author

Damian Clarke is a DPhil. (PhD) student in the Department of Economics at the University of Oxford.

A Data appendix

To test the performance of `genspec`, we use the benchmark performance of Hoover and Perez (1999). They use data from the Citibank economic database with 18 macroeconomic variables over the period 1959 quarter 1 to 1995 quarter 1. These variables include gross national product, M1, M2, labor force and unemployment rates, government purchases, and so on. They difference these data to ensure that each series is stationary.

In this article, we work with the same dataset after performing the same transformations. From these 18 underlying macroeconomic variables (and their first lags), Hoover and Perez (1999) generate artificial variables for consumption. Nine such models are generated with two different independent variables and their lags and the lags of the dependent variable. In table 2, we briefly describe these models (as laid out in table 3 of Hoover and Perez [1999]).

Table 2. Models to test the performance of **genspec**

Model	DGP
Model 1	$y_{1t} = 130.0 \times u_t$
Model 2	$y_{2t} = 130.0 \times u_t^*$
Model 3	$\ln(y_3)_t = 0.395 \times \ln(y_3)_{t-1} + 0.3995 \times \ln(y_3)_{t-2} + 0.00172 \times u_t$
Model 4	$y_{4t} = 1.33 \times fm1dq_t + 9.73 \times u_t$
Model 5	$y_{5t} = -0.046 \times ggeq_t + 0.11 \times u_t$
Model 6	$y_{6t} = 0.67 \times fm1dq_t - 0.023 \times ggeq_t + 4.92 \times u_t$
Model 7	$y_{7t} = 1.33 \times fm1dq_t + 9.73 \times u_t$
Model 8	$y_{8t} = -0.046 \times ggeq_t + 0.11u_t^*$
Model 9	$y_{9t} = 0.67 \times fm1dq_t - 0.023 \times ggeq_t + 4.92u_t$

Notes: The error terms follow $u_t \sim N(0, 1)$ and $u_t^* = 0.75u_{t-1}^* + u_t\sqrt{7/4}$. Models involving the first-order autoregressive u_t^* can be rearranged to include only u_t and one lag of the dependent variable and any independent variables included in the model. The independent variable $fm1dq_t$ refers to M1 money supply, and $ggeq_t$ refers to government spending.

Each of these nine models results in one artificial consumption variable denominated y_{nt} . These y_{nt} variables are then used as the dependent variables for a GETS model search, with 40 independent variables included as candidate variables. These 40 variables are each of the 18 macroeconomic variables in the Citibank economic dataset, the first lags of these variables, and the first to fourth lags of the y_{nt} variable in question. The full-transformed dataset, including a simulated set of u and y_{nt} variables, is available at <https://sites.google.com/site/damianclarke/research>.⁶

6. The untransformed original data are also available.