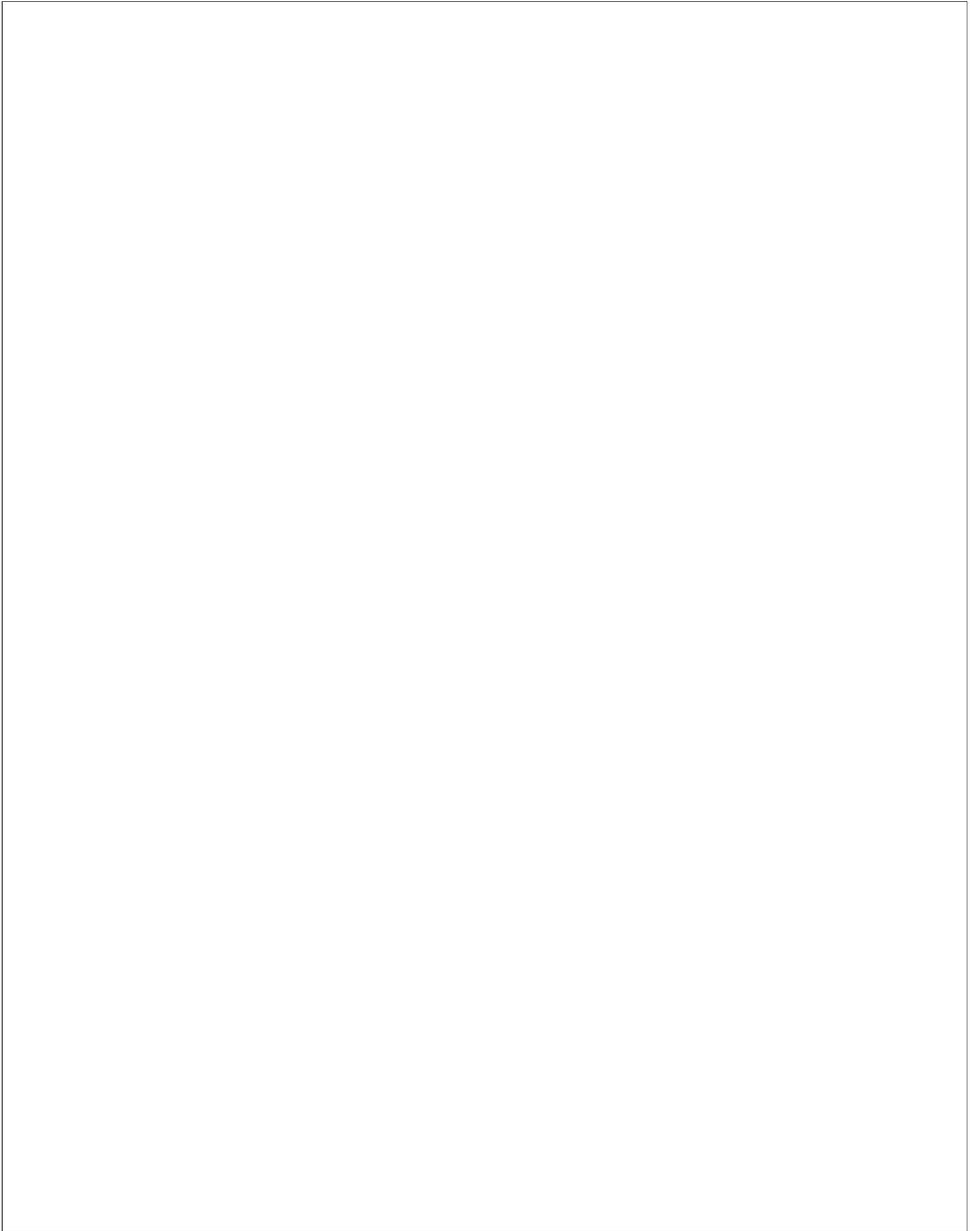


THE STATA JOURNAL

Volume 17 Number 4 2017



A Stata Press publication
StataCorp LLC
College Station, Texas



THE STATA JOURNAL

Editors

H. JOSEPH NEWTON
Department of Statistics
Texas A&M University
College Station, Texas
editors@stata-journal.com

NICHOLAS J. COX
Department of Geography
Durham University
Durham, UK
editors@stata-journal.com

Associate Editors

CHRISTOPHER F. BAUM, Boston College
NATHANIEL BECK, New York University
RINO BELLOCCO, Karolinska Institutet, Sweden, and
University of Milano-Bicocca, Italy
MAARTEN L. BUIS, University of Konstanz, Germany
A. COLIN CAMERON, University of California–Davis
MARIO A. CLEVES, University of Arkansas for
Medical Sciences
WILLIAM D. DUPONT, Vanderbilt University
PHILIP ENDER, University of California–Los Angeles
DAVID EPSTEIN, Gerson Lehrman Group
ALLAN GREGORY, Queen’s University
JAMES HARDIN, University of South Carolina
BEN JANN, University of Bern, Switzerland
STEPHEN JENKINS, London School of Economics and
Political Science

ULRICH KOHLER, University of Potsdam, Germany
FRAUKE KREUTER, Univ. of Maryland–College Park
PETER A. LACHENBRUCH, Oregon State University
JENS LAURITSEN, Odense University Hospital
STANLEY LEMESHOW, Ohio State University
J. SCOTT LONG, Indiana University
ROGER NEWSON, Imperial College, London
AUSTIN NICHOLS, Abt Associates, Washington, DC
MARCELLO PAGANO, Harvard School of Public Health
SOPHIA RABE-HESKETH, Univ. of California–Berkeley
J. PATRICK ROYSTON, MRC CTU at UCL, London, UK
MARK E. SCHAFFER, Heriot-Watt Univ., Edinburgh
JEROEN WEESIE, Utrecht University
IAN WHITE, MRC CTU at UCL, London, UK
NICHOLAS J. G. WINTER, University of Virginia
JEFFREY WOOLDRIDGE, Michigan State University

Stata Press Editorial Manager

LISA GILMORE

Stata Press Copy Editors

ADAM CRAWLEY, DAVID CULWELL, and DEIRDRE SKAGGS

The *Stata Journal* publishes reviewed papers together with shorter notes or comments, regular columns, book reviews, and other material of interest to Stata users. Examples of the types of papers include 1) expository papers that link the use of Stata commands or programs to associated principles, such as those that will serve as tutorials for users first encountering a new field of statistics or a major new technique; 2) papers that go “beyond the Stata manual” in explaining key features or uses of Stata that are of interest to intermediate or advanced users of Stata; 3) papers that discuss new commands or Stata programs of interest either to a wide spectrum of users (e.g., in data management or graphics) or to some large segment of Stata users (e.g., in survey statistics, survival analysis, panel analysis, or limited dependent variable modeling); 4) papers analyzing the statistical properties of new or existing estimators and tests in Stata; 5) papers that could be of interest or usefulness to researchers, especially in fields that are of practical importance but are not often included in texts or other journals, such as the use of Stata in managing datasets, especially large datasets, with advice from hard-won experience; and 6) papers of interest to those who teach, including Stata with topics such as extended examples of techniques and interpretation of results, simulations of statistical concepts, and overviews of subject areas.

The *Stata Journal* is indexed and abstracted by *CompuMath Citation Index*, *Current Contents/Social and Behavioral Sciences*, *RePEc: Research Papers in Economics*, *Science Citation Index Expanded* (also known as *SciSearch*), *Scopus*, and *Social Sciences Citation Index*.

For more information on the *Stata Journal*, including information for authors, see the webpage

<http://www.stata-journal.com>

Subscriptions are available from StataCorp, 4905 Lakeway Drive, College Station, Texas 77845, telephone 979-696-4600 or 800-782-8272, fax 979-696-4601, or online at

<http://www.stata.com/bookstore/sj.html>

Subscription rates listed below include both a printed and an electronic copy unless otherwise mentioned.

U.S. and Canada		Elsewhere	
Printed & electronic		Printed & electronic	
1-year subscription	\$124	1-year subscription	\$154
2-year subscription	\$224	2-year subscription	\$284
3-year subscription	\$310	3-year subscription	\$400
1-year student subscription	\$ 89	1-year student subscription	\$119
1-year institutional subscription	\$375	1-year institutional subscription	\$405
2-year institutional subscription	\$679	2-year institutional subscription	\$739
3-year institutional subscription	\$935	3-year institutional subscription	\$1,025
Electronic only		Electronic only	
1-year subscription	\$ 89	1-year subscription	\$ 89
2-year subscription	\$162	2-year subscription	\$162
3-year subscription	\$229	3-year subscription	\$229
1-year student subscription	\$ 62	1-year student subscription	\$ 62

Back issues of the *Stata Journal* may be ordered online at

<http://www.stata.com/bookstore/sjj.html>

Individual articles three or more years old may be accessed online without charge. More recent articles may be ordered online.

<http://www.stata-journal.com/archives.html>

The *Stata Journal* is published quarterly by the Stata Press, College Station, Texas, USA.

Address changes should be sent to the *Stata Journal*, StataCorp, 4905 Lakeway Drive, College Station, TX 77845, USA, or emailed to sj@stata.com.



Copyright © 2017 by StataCorp LLC

Copyright Statement: The *Stata Journal* and the contents of the supporting files (programs, datasets, and help files) are copyright © by StataCorp LLC. The contents of the supporting files (programs, datasets, and help files) may be copied or reproduced by any means whatsoever, in whole or in part, as long as any copy or reproduction includes attribution to both (1) the author and (2) the *Stata Journal*.

The articles appearing in the *Stata Journal* may be copied or reproduced as printed copies, in whole or in part, as long as any copy or reproduction includes attribution to both (1) the author and (2) the *Stata Journal*.

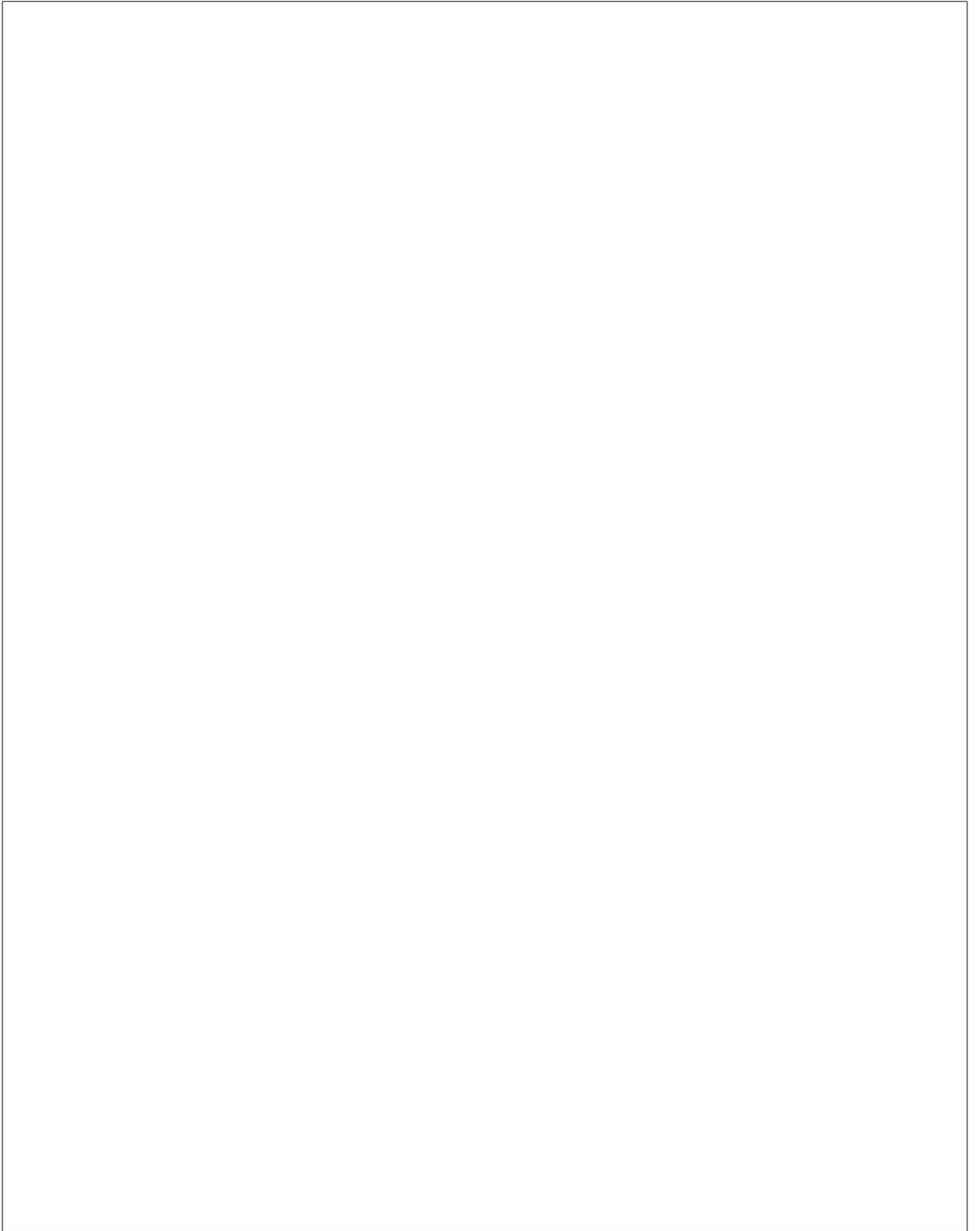
Written permission must be obtained from StataCorp if you wish to make electronic copies of the insertions. This precludes placing electronic copies of the *Stata Journal*, in whole or in part, on publicly accessible websites, fileservers, or other locations where the copy may be accessed by anyone other than the subscriber.

Users of any of the software, ideas, data, or other materials published in the *Stata Journal* or the supporting files understand that such use is made without warranty of any kind, by either the *Stata Journal*, the author, or StataCorp. In particular, there is no warranty of fitness of purpose or merchantability, nor for special, incidental, or consequential damages such as loss of profits. The purpose of the *Stata Journal* is to promote free communication among Stata users.

The *Stata Journal*, electronic version (ISSN 1536-8734) is a publication of Stata Press. Stata, **STATA**, Stata Press, Mata, **MATA**, and NetCourse are registered trademarks of StataCorp LLC.

THE STATA JOURNAL

Articles and Columns	781
The Stata Journal Editors' Prize 2017: Ben Jann	781
Reconstructing time-to-event data from published Kaplan–Meier curves Y. Wei and P. Royston	786
Identification and estimation of treatment effects in the presence of (correlated) neighborhood interactions: Model and Stata implementation via ntreatreg... G. Cerulli	803
The synth_runner package: Utilities to automate synthetic control estimation using synth S. Galiani and B. Quistorff	834
Implementing tests for forecast evaluation in the presence of instabilities B. Rossi and M. Soupre	850
Text mining with n-gram variables M. Schonlau, N. Guenther, and I. Sucholutsky	866
Econometric convergence test and club clustering using Stata..... K. Du	882
Automatic portmanteau tests with applications to market risk management..... G. Zhu, Z. Du, and J. C. Escanciano	901
Two-stage residual inclusion estimation: A practitioners guide to Stata implemen- tation J. V. Terza	916
Causal effect estimation and inference using Stata..... J. V. Terza	939
A simple command to calculate travel distance and travel time S. Weber and M. Péclat	962
Testing for Granger causality in panel data..... L. Lopez and S. Weber	972
Response surface models for the Elliott, Rothenberg, and Stock unit-root test... J. Otero and C. F. Baum	985
Assessing the calibration of dichotomous outcome models with the calibration belt G. Nattino, S. Lemeshow, G. Phillips, S. Finazzi, and G. Bertolini	1003
Estimating receiver operative characteristic curves for time-dependent outcomes: The stroccurve package M. Cattaneo, P. Malighetti, and D. Spinelli	1015
Software Updates	1024



The Stata Journal Editors' Prize 2017: Ben Jann



1 Prize announcement

The editors of the *Stata Journal* are delighted to announce the award of the Editors' Prize for 2017 to **Ben Jann**.¹

The aim of the prize is to reward contributions to the Stata community in respect of one or more outstanding papers published in the *Journal* in the previous three calendar years. For the original announcement of the prize and its precise terms of reference, see [Newton and Cox \(2012\)](#), which is accessible at the following website: <http://www.stata-journal.com/sjpdf.html?articlenum=gn0052>.

Ben Jann was born in Lucerne, Switzerland in 1972. He studied in schools in Arisdorf and Liestal and at the University of Bern and ETH Zürich, receiving a Doctor of Sciences degree from the latter in 2008 for work on the Swiss labor market. Ben was appointed Professor of Sociology, with a specialization in social stratification, at the University of Bern in 2014, having been on the faculty there since 2010. His interests span several fields in the social sciences, including research methods. Recent research includes work on the development of economic inequalities in Switzerland, survey methods for collecting data on sensitive topics, intergenerational social mobility and trends in assortative mating in Switzerland, and the relation between immigration and crime. He has

1. Ben has asked that the prize money be awarded directly to Consciente, a small charity that sponsors educational programs in Morazán, one of the poorest regions of El Salvador (<http://consciente.ch/>).

been principal investigator since 2014 of TREE, a large-scale multicohort panel study in Switzerland on transitions from education to employment (<http://www.tree.unibe.ch>).

Ben is a prolific author. Publications include a textbook on statistics ([Jann 2005a](#)), jointly edited books on sociology and survey methods, and many papers in social science journals as well as in the *Stata Journal*.

This is not Ben's first prize for publications, but his fourth, following awards for papers in *Swiss Journal for Sociology* (2008), *American Sociological Review* (2015), and *Survey Research Methods* (2017).

Ben has been highly active in the Stata community for several years. He has been an associate editor of the *Stata Journal* since 2005 and has given presentations at Stata conferences in Germany, Britain, the United States, Italy, Belgium, Norway, and Switzerland (the order is by date of the first talk in each country). The references below give a complete list of his papers in the *Stata Journal*. In addition, he has posted many programs on the Statistical Software Components (SSC) archive, being single or joint author of some 60 packages there.

The award recognizes specifically four outstanding papers by Ben ([Jann 2014a](#), [2016b,a,c](#)):

- Plotting regression coefficients and other estimates (*Stata Journal* 14: 708–737)
- Creating L^AT_EX documents from within Stata using texdoc (*Stata Journal* 16: 245–263)
- Assessing inequality using percentile shares (*Stata Journal* 16: 264–300)
- Estimating Lorenz and concentration curves (*Stata Journal* 16: 837–866)

Choosing other examples could only be capricious. However, we would be remiss not to flag his paper on the Blinder–Oaxaca decomposition for linear regression models ([Jann 2008a](#)), which is his most highly cited, with at the time of writing over 1,000 citations recorded by Google Scholar. The Blinder–Oaxaca method for decomposing differences in mean outcomes between two groups (for example, wages between blacks and whites) into contributions representing differences in average characteristics across the groups and differences across in “returns” to those characteristics is now a standard tool in much applied economics and related social sciences. Ben's program fits the regression models underpinning the method and then derives the two components of the decomposition. It also supports some variants on the original Blinder–Oaxaca decomposition rule and provides standard errors for the components.

The first paper of four identified as outstanding will be thought of by most of its readers as Ben's paper introducing his command `coefplot`. It is a major contribution to one of the most needed areas of reporting Stata results, graphical display of regression results, regression being considered very broadly. Although some fields lag in this respect (no names here from us!), graphical output is widely appreciated in presentations and in

scientific literature, not least because graphs are often much easier to read than tables. Such plots can be produced in Stata by the `marginsplot` command. However, while `marginsplot` is versatile and flexible, it has two major limitations: it can only process results left behind by `margins`, and it can handle only one set of results at a time.

The paper introduces the new command `coefplot` that overcomes these limitations. It plots results from any estimation command and combines results from several models into one graph. The default behavior of `coefplot` is to plot markers for coefficients and horizontal spikes for confidence intervals, but it can also produce other types of graphs. All that is needed is a previous modeling command leaving accessible stored results. For more information, see <http://repec.sowi.unibe.ch/stata/coefplot/>.

The second paper identified as outstanding continues the theme of tools for reporting, which recurs throughout Ben’s work on Stata commands. It is also focused on a new command—namely, `texdoc`—a tool for creating L^AT_EX documents from within Stata. Specifically, it provides a way to embed L^AT_EX code directly in a do-file and to automate the integration of results from Stata in the final document. One can use the command, for example, to assemble automatic reports, write a *Stata Journal* article, prepare slides for classes, or create solutions for homework assignments. For more information, see <http://repec.sowi.unibe.ch/stata/texdoc/>.

`texdoc` can be seen as a younger sibling or perhaps out-of-town relation to Ben’s most well-known reporting program, `estout` with wrapper `esttab`. This is well documented through not only papers in this *Journal* (Jann 2005b, 2007a, 2014b; Jann and Long 2010) but also updates on the SSC archive and <http://repec.sowi.unibe.ch/stata/estout/>.

With these and related commands, Ben has increased the potential for Stata users to report “reproducible research” and to report it as clearly as possible. The community gains from helpful displays and clear audit trails. With its do- and log-files, Stata eases this at a low level, but a finished research product is a paper, report, or book and tools at that scale are also crucial.

The third and fourth papers identified sample Ben’s more statistical work.

The third paper implements and develops percentile shares, an increasingly popular approach for analyzing distributional inequalities. For example, in work on income and wealth, economists and other social scientists often report top-percentage shares, using varying percentages as thresholds (top 10%, top 1%, top 0.1%, and so forth). However, analysis of percentile shares at other positions in the distribution may also be of interest, especially including percentiles at the lower end of a distribution (in this case, poor people too). The approach is directly related to the well-known Lorenz curve, because the ordinates of that curve are cumulative percentile shares. It is easy to think of quite different variables in different fields whose analysis might profit from such an approach.

This paper presents a new command, `pshare`, that estimates percentile shares from individual-level data and displays the results using histograms or stacked bar charts. By focusing on one particular distributional feature, Ben provides rich functionality. `pshare` is an estimation program: careful attention is given to calculating standard errors,

including support for `svyset` data. It is possible to analyze distributional differences among subpopulations or across time through estimation of contrasts between percentile shares.

The fourth paper identified is complementary. Lorenz and concentration curves are widely used tools in inequality research. The paper presents another new command, `lorenz`, that estimates Lorenz and concentration curves from individual-level data and, optionally, displays the results in a graph. The `lorenz` command supports relative, generalized, absolute, unnormalized, custom-normalized Lorenz, and concentration curves. It also provides tools for computing contrasts between different subpopulations or outcome variables. `lorenz` fully supports variance estimation for complex samples.

Here we see the same strong positive features repeated. Ben writes definitive and substantial papers that explain and extend the state of the art in particular areas. He has a special taste and talent for taking fundamental descriptive techniques and implementing twists and turns of the main idea that are sound and useful and linking them both to helpful graphics and to appropriate inferential machinery.

These papers are excellent samples of a body of work that shows outstanding grasp of scientific, statistical, and Stata considerations. Code, documentation, and case studies are all provided with exemplary care, clarity, and skill. Ben draws upon a profound and detailed understanding of underlying statistical principles and of how Stata (and, very importantly, Mata) works and can be made to work for the benefit of researchers. His own research needs have driven what he writes, but he is always mindful of that generality in software writing, which maximizes the usefulness of programs.

In sum, we salute Ben Jann for outstanding contributions to the Stata community and specifically through his recent publications in the *Stata Journal*.

As editors, we are indebted to the awardee for biographical material and to a necessarily anonymous nominator for a most helpful appreciation.

H. Joseph Newton and Nicholas J. Cox
Editors, *Stata Journal*

2 References

- Jann, B. 2004. Stata tip 8: Splitting time-span records with categorical time-varying covariates. *Stata Journal* 8: 221–222.
- . 2005a. *Einführung in die Statistik*. 2nd ed. München: Oldenbourg.
- . 2005b. Making regression tables from stored estimates. *Stata Journal* 5: 288–308.
- . 2005c. Tabulation of multiple responses. *Stata Journal* 5: 92–122.
- . 2007a. Making regression tables simplified. *Stata Journal* 7: 227–244.

- . 2007b. Stata tip 44: Get a handle on your sample. *Stata Journal* 7: 266–267.
- . 2008a. The Blinder–Oaxaca decomposition for linear regression models. *Stata Journal* 8: 453–479.
- . 2008b. Multinomial goodness-of-fit: Large-sample tests with survey design correction and exact tests for small samples. *Stata Journal* 8: 147–169.
- . 2014a. Plotting regression coefficients and other estimates. *Stata Journal* 14: 708–737.
- . 2014b. Software Updates: Making regression tables from stored estimates. *Stata Journal* 14: 451.
- . 2015a. A note on adding objects to an existing twoway graph. *Stata Journal* 15: 751–755.
- . 2015b. Software Updates: Plotting regression coefficients and other estimates. *Stata Journal* 15: 324.
- . 2015c. Stata tip 122: Variable bar widths in two-way graphs. *Stata Journal* 15: 316–318.
- . 2016a. Assessing inequality using percentile shares. *Stata Journal* 16: 264–300.
- . 2016b. Creating L^AT_EX documents from within Stata using texdoc. *Stata Journal* 16: 245–263.
- . 2016c. Estimating Lorenz and concentration curves. *Stata Journal* 16: 837–866.
- . 2016d. Software Update: Creating L^AT_EX documents from within Stata using texdoc. *Stata Journal* 16: 813–814.
- . 2016e. Software Update: Creating L^AT_EX documents from within Stata using texdoc. *Stata Journal* 16: 1072–1073.
- . 2017. Creating HTML or Markdown documents from within Stata using webdoc. *Stata Journal* 17: 3–38.
- Jann, B., and J. S. Long. 2010. Tabulating SPost results using estout and esttab. *Stata Journal* 10: 46–60.
- Newton, H. J., and N. J. Cox. 2012. Announcement of the Stata Journal Editors’ Prize 2012. *Stata Journal* 12: 1–2.

Reconstructing time-to-event data from published Kaplan–Meier curves

Yinghui Wei
Centre for Mathematical Sciences
School of Computing, Electronics, and Mathematics
Plymouth University
Plymouth, UK
yinghui.wei@plymouth.ac.uk

Patrick Royston
MRC Clinical Trials Unit
University College London
London, UK
j.royston@ucl.ac.uk

Abstract. Hazard ratios can be approximated by data extracted from published Kaplan–Meier curves. Recently, this curve approach has been extended beyond hazard-ratio approximation with the capability of constructing time-to-event data at the individual level. In this article, we introduce a command, `ipdfc`, to implement the reconstruction method to convert Kaplan–Meier curves to time-to-event data. We give examples to illustrate how to use the command.

Keywords: st0498, ipdfc, time-to-event data, Kaplan–Meier curves, hazard ratios

1 Introduction

The hazard ratio is often recommended as an appropriate effect measure in the analysis of randomized controlled trials with time-to-event outcomes (Parmar, Torri, and Stewart 1989; Deeks, Higgins, and Altman 2008) and has become the de facto standard approach to analysis. In meta-analysis of aggregated time-to-event data across trials, an essential step is to extract the (log) hazard ratio and its variance from published trial reports. Various extraction methods have been described (Parmar, Torri, and Stewart 1989; Williamson et al. 2002; Tierney et al. 2007), including direct and indirect estimates of hazard ratios based on 95% confidence intervals (CIs), p -values for the log-rank test or the Mantel–Haenszel test, and regression coefficients in the Cox proportional hazards model. An approximation to hazard ratios can also be derived by a “curve approach”, as described by Parmar, Torri, and Stewart (1989) and Tierney et al. (2007). The curve approach uses the extracted ordinate (y) and abscissa (x) values from the Kaplan–Meier curve to calculate hazard ratios for each time interval for which the number of patients at risk was reported. The overall hazard ratio during the follow-up phase is then derived by a weighted sum of the individual estimates of hazard ratios across time intervals, with the weights inversely proportional to the variance of each estimate (Parmar, Torri, and Stewart 1989).

The curve approach has been extended (Guyot et al. 2012) beyond the estimation of hazard ratios to the reconstruction of time-to-event data at the individual level. The availability of reconstructed individual-level data allows one to fit alternative models in secondary analyses if desired. Because nonproportional hazards are increasingly reported in trials, alternative measures (such as restricted mean survival time) that do not

require the proportional-hazards assumption may have a more intuitive interpretation under nonproportional hazards (Royston and Parmar 2011). Because the proportional-hazards assumption may not be satisfied for all trials in a meta-analysis, alternative effect measures to hazard ratios may be more appropriate in such settings (Wei et al. 2015). However, by definition, newly developed effect measures are not reported in earlier trial publications. The use of these measures therefore relies either on collaborative sharing of individual-level data or on methods that enable reconstruction of such data from trial reports.

The reconstruction algorithm was written as an R function (Guyot et al. 2012). In this article, we present an implementation of this algorithm with improvements by introducing a command, `ipdfc`, that has the following features:

- Uses the curve approach to reconstructing individual-level time-to-event data based on the published Kaplan–Meier curves.
- Uses the number of patients at risk, as reported in the trial publication.
- Can identify which extracted time points correspond to the lower and upper endpoint of each time interval in the risk table.
- Can use survival probabilities, survival percentages, failure probabilities, or failure percentages as data input.
- Incorporates correction of monotonicity violators in the extracted data for survival probabilities, survival percentages, failure probabilities, or failure percentages.

In the following section, we briefly overview the methods underpinning the `ipdfc` command introduced in this article. We then give detailed descriptions of syntax and options. We then demonstrate its application in two examples from trial publications and assess the approximation accuracy by comparing summary statistics between the reconstructed data and the original publications. We close with a discussion.

2 Methods

2.1 Extracting data from published Kaplan–Meier curves

The reconstruction of time-to-event observations is based on data extracted from published Kaplan–Meier curves. In such curves, the x values usually represent the follow-up time since randomization; the y values may represent the survival probabilities, survival percentages, failure probabilities, or failure percentages at the corresponding time points, as specified in the trial publication. These measures can be transformed arithmetically into survival probabilities. In addition to data from curves, the number of patients randomized into each arm of a trial should be extracted from publications.

The DigitizeIt (<http://www.digitizeit.de/>) software application is a suitable tool for extracting data from a graphical image. Data extraction using this software is far

more rapid, detailed, accurate, and reliable than manually applying pencil and ruler methods to a reduced image of the graph. If a curve is displayed as a clearly defined, unbroken line, DigitizeIt can automatically read off the x and y values at a large number of time points. This helps ensure the good quality of data required as input in the reconstruction of time-to-event observations. However, if the curve is presented as a broken (for example, dashed) line, the operator must extract data semi-manually by clicking on individual points on the curve using a mouse. Because each click returns only one data point, many clicks must be made to obtain sufficient data when there are many jumps in the curve. In contrast, within a specific time interval where there are few events or where the survival curves are flat, little information is available and correspondingly few clicks are required.

In addition, it is important to extract the number of patients at risk for each arm at regular time intervals during the follow-up. This information, usually known as the risk table, is often presented beneath the published Kaplan–Meier curves. The accuracy of the approximated time-to-event data can be improved by incorporating information provided in the risk table (Tierney et al. 2007).

2.2 Adjusting monotonicity violators

Because a survival function is by definition monotone decreasing with time, the y values extracted from a survival curve should also be monotone when ordered by the corresponding x values. However, there may be violators among the extracted data such that the monotonicity constraint is not satisfied. This is due to publication quality of the curves or errors in controlling the mouse clicks (Guyot et al. 2012). The reconstruction algorithm involves estimating survival functions. Monotonicity violators can lead to incorrect estimates for the number of events, and subsequently incorrect estimates of the survival function, which prevents the reconstruction from working. It is therefore crucial to correct the values for violators to ensure monotonicity. Because violators are often multiple, a systematic method is required.

With the `ipdfc` command, we incorporate alternative methods for the correction of violators. The first method, isotonic regression (Barlow et al. 1972), may help to detect violators and correct their values by using a pool-adjacent-violators algorithm. Adjacent violators occur where a pair of adjacent times and corresponding survival probabilities is inappropriately ordered, for example, time = (1.0, 1.1), survival = (0.91, 0.92). Briefly, the pool-adjacent-violators algorithm replaces the adjacent violators with their mean so that the data satisfy the monotonicity constraint. The technique has been recently coded in a command called `irax` (van Putten and Royston 2017), which can be called in our command. We also consider an alternative. We replace the value of a violator with the value of its adjacent violator such that the corrected data satisfy the monotonicity. We expect that using either method will lead to similar results because the absolute difference between the values of adjacent violators is often too small to have a material influence on the resulting data.

2.3 Algorithm to reconstruct survival data

We now briefly describe the algorithm underpinning the `ipdfc` command. We start with introducing notations. Let S_k denote survival probabilities at time t_k , where $k = 1, 2, \dots, K$ and K is the total number of data points extracted. The survival probabilities S_k and the corresponding time t_k may be extracted from the respective y and x coordinates of a Kaplan–Meier curve. Let $nrisk_i$ denote the number of patients at risk at time $trisk_i$, where $i = 1, \dots, T$, with T as the number of intervals where the number of patients at risk is reported. The number of extracted data points, K , is often much greater than T , the total number of intervals at the risk table. If the risk table is not reported, we have $T = 1$.

The four quantities S_k , t_k , $nrisk_i$, and $trisk_i$ are the required input in the algorithm. As mentioned above, the number of patients at risk, if available, should be included in the algorithm. Otherwise, if $T = 1$, the number of patients randomized to each arm should be included in the algorithm. The total number of events, D , can also be used in the reconstruction.

In the algorithm, we will estimate the following quantities: the number of censoring, \widehat{c}_k ; the number of events, \widehat{d}_k ; the censoring time, \widehat{ctime}_k ; and the event time, \widehat{dtime}_k . To estimate these quantities, we implement the algorithm described in [Guyot et al. \(2012\)](#) by adding three new components for improvements. First, we calculate $lower_i$ and $upper_i$ by using the input data $t_k, trisk_i$. Here, $lower_i$ and $upper_i$ are respectively the indices for the first and last time points extracted from the time interval $[trisk_i, trisk_{i+1}]$. For each of these time intervals, $lower_i$ is equal to $\min\{k : t_k \geq trisk_i\}$, and $upper_i$ is equal to $\max\{k : t_k \leq trisk_{i+1}\}$. Thus, $lower_i$ and $upper_i$ are not required as data input like the R code of [Guyot et al. \(2012\)](#). Second, we adjust the values of monotonicity violators by using isotonic regression or its alternative as just described. Third, we extend the algorithm to the situation where the number at risk is reported at the last time interval, at which we allow the calculation of the number of censoring following the same methods as those for the other intervals. The full algorithm is given in the appendix of this article.

3 The ipdfc command

3.1 Syntax

```
ipdfc, surv(varname) tstart(varname) trisk(varname) nrisk(varname)
  generate(varname1 varname2) saving(filename[, replace]) [probability
  failure isotonic totevents(#[)] ]
```

This syntax converts data extracted from a Kaplan–Meier curve to time-to-event data. The syntax does not handle more than one sample at a time. When dealing with a trial having more than one arm, the syntax converts data extracted from one curve at

a time to time-to-event data for the respective arm. This should be done for all arms individually, and further data management is needed to amalgamate the data from all arms of a trial, if the data are from a trial. We will illustrate this in the examples given later in this article.

3.2 Options

surv(*varname*) specifies the data extracted from the ordinate (*y* axis) of a published Kaplan–Meier curve. The data may be survival probabilities, survival percentages, failure probabilities, or failure percentages. By default, *varname* is assumed to contain survival percentages. **surv**() is required.

tstart(*varname*) specifies the time since randomization as extracted from the abscissa (*x* axis) of a published Kaplan–Meier curve. The time could be in any units (for example, days, months, or years), as specified in the publication. **tstart**() is required.

trisk(*varname*) specifies the times corresponding to the numbers of patients at risk in **nrisk**(). Set **trisk**() as 0 only if the total number of patients in the sample is known. **trisk**() is required.

nrisk(*varname*) supplies the number of patients at risk for each time in **trisk**(). Both **trisk**() and **nrisk**() are often found in a risk table displayed beneath published Kaplan–Meier curves. If no risk table is available, specify **nrisk**() as the number of patients in the sample, and specify **trisk**() as 0. **nrisk**() is required.

generate(*varname1* *varname2*) generates the time-to-event outputs extracted from the input information. *varname1* and *varname2* specify two new variables, the time to an event and an event indicator (1 = event, 0 = censored). For example, specifying **generate**(**time** **event**) would create **time** as the time to event and **event** as the event indicator. **generate**() is required.

saving(*filename*[, **replace**]) saves the reconstructed survival data to *filename.dta*. **replace** allows the file to be replaced if it already exists. **saving**() is required.

probability signifies that *varname* in **surv**() contains probabilities rather than the default percentages.

failure signifies that *varname* in **surv**() contains failure information rather than the default survival information.

isotonic uses isotonic regression to adjust values that violate the time-related monotonicity in **surv**(). By default, an alternative, simpler method is used to correct the values of violators by replacing the value of a violator with the value of its adjacent violator.

totevents(*#*) is the total number of events and is used to adjust the number of observations censored in the final interval of the risk table.

4 Illustrative examples

4.1 Example 1: Head and neck cancer trial

Our first example is a two-arm randomized controlled trial published in [Bonner et al. \(2006\)](#). A total of 424 participants with locoregionally advanced head and neck cancer were randomized to receive either radiotherapy plus cetuximab or radiotherapy alone. The primary outcome was the duration of locoregional control. Both Kaplan–Meier curves and the hazard ratio were reported. This example was first used in [Guyot et al. \(2012\)](#) to illustrate the application of the reconstruction method. Here we use `ipdfc` to reconstruct the survival data and to illustrate how it performs compared with [Guyot et al. \(2012\)](#) and with the results in the original publication. We run the steps for each arm separately to obtain arm-specific data based on the associated Kaplan–Meier curve from the trial report.

We read in a text file for the control arm by calling `import delimited`.

```
. import delimited using "head_and_neck_arm0.txt"
(4 vars, 102 obs)
```

The text file contains four variables: `ts` and `s` as the data extracted from the x axis and y axis of a curve, and `trisk()` and `nrisk()` from the risk table.

We regenerate data for the control group by calling `ipdfc`.

```
. ipdfc, surv(s) tstart(ts) trisk(trisk) nrisk(nrisk) isotonic
> generate(t_ipd event_ipd) saving(temp0)
```

Because the extracted y values are survival percentages in this example, we need not use either `probability` or `failure` to convert `s`. However, we use the option `isotonic` to evoke isotonic regression to correct monotonicity violators. The regenerated survival data are stored in the file `temp0.dta`.

We run the following steps for the treatment group:

```
. import delimited using "head_and_neck_arm1.txt", clear
(4 vars, 87 obs)
. ipdfc, surv(s) tstart(ts) trisk(trisk) nrisk(nrisk) isotonic
> generate(t_ipd event_ipd) saving(temp1)
```

The regenerated survival data are stored in the file `temp1.dta`.

The data simulated from both arms are then combined and specified with an arm indicator.

```
. use temp0, clear
. gen byte arm = 0
. append using temp1
. replace arm = 1 if missing(arm)
(213 real changes made)
```

792 *Reconstructing time-to-event data from published Kaplan–Meier curves*

In the amalgamated data, there are three variables—`t_ipd`, `event_ipd`, and `arm`—which are time to event, event indicator, and arm indicator, respectively. We label the arm indicator as `Radiotherapy` and `Radiotherapy plus cetuximab`, as specified in the trial publication.

```
. label define ARM 0 "Radiotherapy" 1 "Radiotherapy plus cetuximab"  
. label values arm ARM
```

We set time as the time to failure.

```
. stset t_ipd, failure(event_ipd)  
      (output omitted)
```

By calling `sts graph`, we reconstruct the survival curves (see figure 1).

```
. sts graph, by(arm) title("") xlabel(0(10)70) ylabel(0(0.2)1)  
> risktable(0(10)50, order(2 "Radiotherapy" 1 "Radiotherapy plus"))  
> xtitle("Months") l2title("Locoregional control")  
> scheme(sj) graphregion(fcolor(white))  
> plot1opts(lpatter(solid) lcolor(gs12))  
> plot2opts(lpatter(solid) lcolor(black))  
> text(-0.38 -9.4 "cetuximab")  
> legend(off)  
> text (0.52 53 "Radiotherapy plus cetuximab") text(0.20 60 "Radiotherapy")  
      failure _d: event_ipd  
      analysis time _t: t_ipd
```

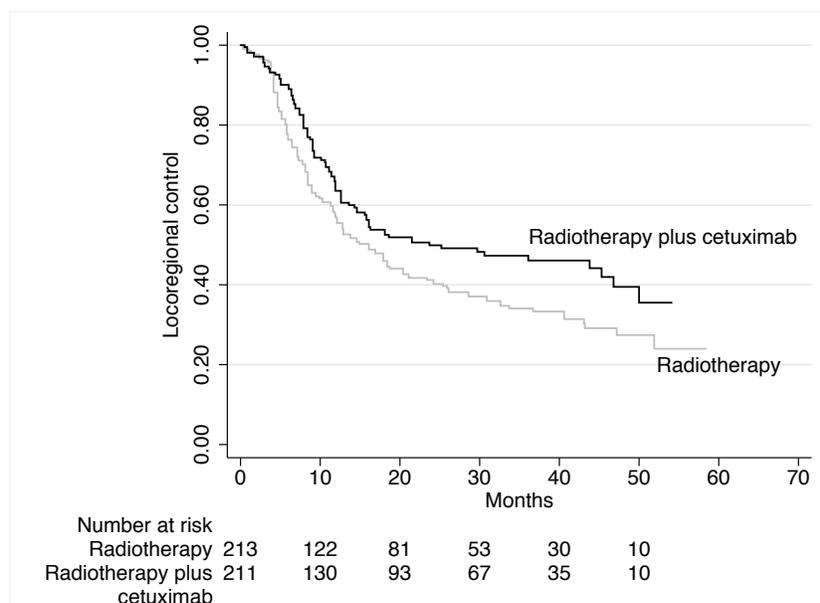


Figure 1. Reconstructed Kaplan–Meier curves for locoregional control among patients with head and neck cancer (Bonner et al. 2006). Patients are randomized to receive radiotherapy plus cetuximab or radiotherapy alone.

The survival analysis is carried out by calling `stcox arm`.

```
. stcox arm
      failure _d: event_ipd
      analysis time _t: t_ipd
Iteration 0:  log likelihood = -1323.3427
Iteration 1:  log likelihood = -1320.1905
Iteration 2:  log likelihood = -1320.1899
Refining estimates:
Iteration 0:  log likelihood = -1320.1899
Cox regression -- Breslow method for ties
No. of subjects =      424          Number of obs   =      424
No. of failures =      241
Time at risk    = 8412.821523
Log likelihood  = -1320.1899          LR chi2(1)    =      6.31
                                          Prob > chi2   =      0.0120
```

_t	Haz. Ratio	Std. Err.	z	P> z	[95% Conf. Interval]
arm	.7208993	.0947487	-2.49	0.013	.5571859 .9327152

The reconstructed Kaplan–Meier curves (see figure 1) look similar to the published curves (Bonner et al. 2006). There is only a small discrepancy in the numbers of patients

794 *Reconstructing time-to-event data from published Kaplan–Meier curves*

at risk in the radiotherapy plus cetuximab arm. For this arm, based on the reconstructed data, the numbers of patients at risk are 211, 130, 93, 67, 35, and 10, which are similar though not identical to 211, 143, 101, 66, 35, and 9 in the original publication. The discrepancy in the risk table between the approximation (figure 1) and the original publication is very small for the radiotherapy arm.

In table 1, we report percentages of patients surviving one, two, and three years; median duration of locoregional control; and hazard-ratio estimates. The estimates of percentage of surviving and median time to event are close to those in the original publication. The hazard ratio (0.72, 95% CI: [0.56, 0.93]) estimated by our command is close to the hazard ratio (0.73, 95% CI: [0.57, 0.94]) estimated by [Guyot et al. \(2012\)](#). Because we digitize the data independently of [Guyot et al. \(2012\)](#), we do not expect to obtain identical data nor identical results. Though not identical, both approximated hazard ratios are similar to the published hazard ratio (0.68, 95% CI: [0.52, 0.89]).

Table 1. Example 1. Comparison of summary measures estimated from publication and their corresponding reconstructed data

	Original publication	Guyot et al. (2012)	ipdfc
Radiotherapy arm		Percent [95% CI]	Percent [95% CI]
Percent surviving one year	55	56.1 [49.6, 63.3]	56.9 [49.9, 63.2]
Percent surviving two years	41	41.1 [34.7, 48.6]	40.9 [34.2, 47.5]
Percent surviving three years	34	34.7 [28.4, 42.5]	33.5 [27.1, 40.1]
Median duration	14.9	14.9 [11.9, 23.0]	16.1 [11.9, 20.4]
Radiotherapy plus cetuximab arm		Percent [95% CI]	Percent [95% CI]
Survival rate at one year	63	64.0 [57.8, 70.9]	65.4 [58.2, 71.6]
Survival rate at two years	50	50.4 [43.9, 57.8]	51.0 [43.3, 58.6]
Survival rate at three years	47	46.7 [40.1, 54.4]	49.6 [40.4, 55.7]
Median duration	24.4	24.3 [15.7, 45.7]	23.7 [15.6, 46.8]
Hazard ratio with 95% CI			
	0.68 [0.52, 0.89]	0.73 [0.57, 0.94]	0.72 [0.56, 0.93]

4.2 Example 2: ICON7 trial

Our second example is ICON7, a two-arm randomized controlled trial in advanced ovarian cancer ([Perren et al. 2011](#)). A total of 1,528 women were randomized to receive either standard chemotherapy plus bevacizumab or standard chemotherapy alone. From the analysis based on data with 30 months follow-up, [Perren et al. \(2011\)](#) concluded that bevacizumab improved progression-free survival in this population, with hazard ratio 0.81 (95% CI: [0.70, 0.94]; $P = 0.004$ from a log-rank test). [Perren et al. \(2011\)](#) found significant nonproportional hazards ($P < 0.001$) of the treatment effect. Kaplan–Meier curves and the associated risk table for progression-free survival were reported in their

figure 2a, on which we base our reconstruction of the survival data using `ipdfc`. Also, we use the total number of events, `tot`, because it is available.

```
. local tot0 = 464
. local tot1 = 470
. import delimited using "icon7_data_arm0.txt", clear
(4 vars, 86 obs)
. ipdfc, surv(s) tstart(ts) trisk(trisk) nrisk(nrisk) probability isotonic
> tot(`tot0`) generate(t_ipd event_ipd) saving(temp0)
. import delimited using "icon7_data_arm1.txt", clear
(4 vars, 473 obs)
. ipdfc, surv(s) tstart(ts) trisk(trisk) nrisk(nrisk) probability isotonic
> tot(`tot1`) generate(t_ipd event_ipd) saving(temp1)
```

In this example, the extracted y values are survival probabilities. According to the above codes, we use the `probability` option to specify that `surv(s)` represents survival probabilities rather than survival percentages.

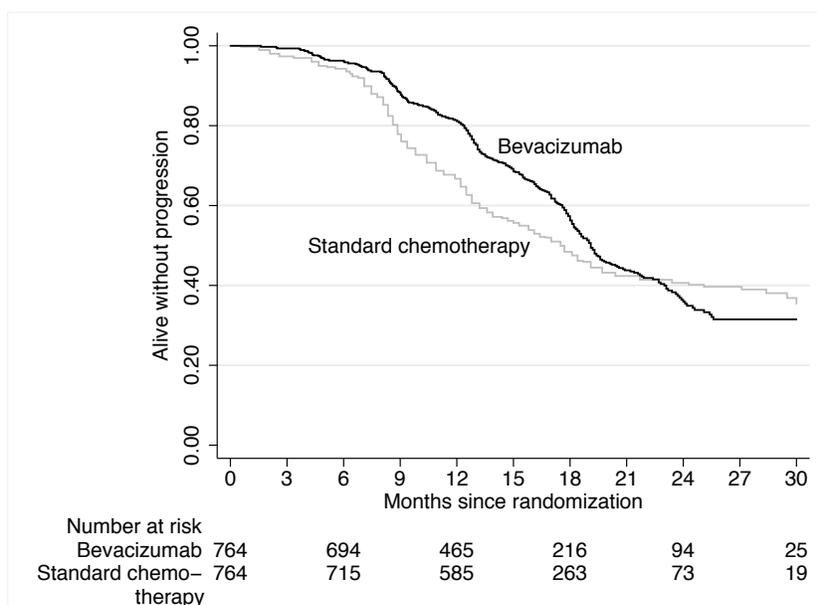


Figure 2. Reconstructed Kaplan–Meier curves for progression-free survival according to treatment group in ICON7 (Perren et al. 2011). Patients are randomized to receive standard chemotherapy plus bevacizumab or standard chemotherapy alone.

796 *Reconstructing time-to-event data from published Kaplan–Meier curves*

The reconstructed Kaplan–Meier curves in figure 2 look similar to those in the original publication (Perren et al. 2011). The number of patients at risk is also well approximated, with most numbers identical to those in the original publication. The little discrepancies lie in 6 months and 12 months. The numbers of patients at risk are 694 at 6 months and 465 at 12 months based on the approximated data, which compared similarly though not identically to the original publication numbers of 693 at 6 months and 464 at 12 months. The estimated hazard ratios, median survival time, and p -values from the log-rank test are also similar to those in the original publication. See table 2 for a comparison of summary measures.

Table 2. Example 2. Comparison of summary measures estimated from publication and their corresponding reconstructed data

	Original publication	Reconstructed data
Log-rank test	$P = 0.004$	$P = 0.009$
Nonproportional hazard test	$P < 0.001$	$P < 0.001$
Hazard ratio	0.81 (95% CI: [0.70, 0.94])	0.83 (95% CI: [0.72, 0.96])
Median survival time		
Chemotherapy arm	17.3	17.5 (95% CI: [16.1, 18.7])
Bevacizumab arm	19.0	19.1 (95% CI: [18.3, 19.9])

4.3 Example 3: EUROPA trial

Our third example, EUROPA, is a two-arm randomized placebo-controlled trial evaluating the efficacy of perindopril in reduction of cardiovascular events among patients with stable coronary artery disease (Fox 2003). In this trial, 12,218 patients were randomly assigned perindopril 8 mg once daily ($n = 6110$) or placebo ($n = 6108$). Kaplan–Meier curves and the associated risk table were presented in figure 2 of the trial report. In Fox (2003), the Cox proportional hazards model was used, but the hazard-ratio estimate was not reported. It was reported in Fox (2003) that perindopril treatment was associated with a significant reduction in the composite events (cardiovascular mortality, nonfatal myocardial infarction, and resuscitated cardiac arrest), with p -value = 0.0003 from a log-rank test and absolute risk reduction of 1.9%.

We extracted the failure percentages and the associated time points, respectively, from the y axis and the x axis of the Kaplan–Meier curves in Fox’s (2003) figure 2. In the following codes, we use the option `failure` to specify that the input data are failure percentages instead of the default survival percentages.

```

. import delimited using "europa_data_arm0.txt", clear
(4 vars, 743 obs)
. ipdffc, surv(s) failure isotonic tstart(ts) trisk(trisk) nrisk(nrisk)
> generate(t_ipd event_ipd) saving(temp0)
. import delimited using "europa_data_arm1.txt", clear
(4 vars, 650 obs)
. ipdffc, surv(s) failure isotonic tstart(ts) trisk(trisk) nrisk(nrisk)
> generate(t_ipd event_ipd) saving(temp1)

```

The Kaplan–Meier curves from the reconstructed data are presented in figure 3. The reconstructed curves appear nearly identical to the original. The reconstructed curves correctly reflect that the benefit of perindopril treatment began to appear at one year and gradually increased throughout the follow-up of the trial. The numbers of patients at risk are also very similar to the reported values, with only a small discrepancy in the placebo arm in two years of follow-up (5,781 in the original report versus 5,783 in the reconstructed data).

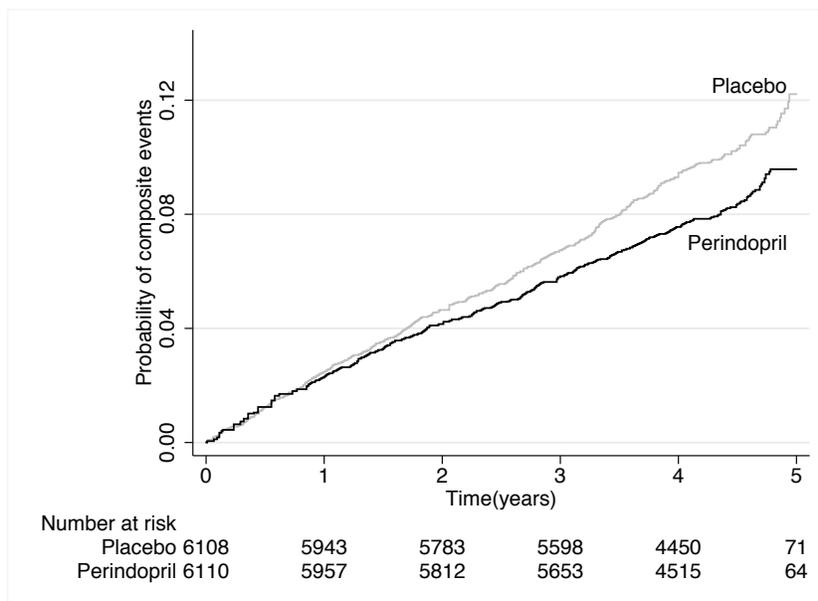


Figure 3. Reconstructed Kaplan–Meier curves for time to first occurrence of event. Patients are randomly assigned to perindopril treatment or placebo in the EUROPA trial (Fox 2003).

798 *Reconstructing time-to-event data from published Kaplan–Meier curves*

Using the reconstructed data, we obtain p -value = 0.0006 from a log-rank test (see table 3). Similar to the original report, this result also suggests that perindopril treatment was associated with a significant reduction in the composite events. We estimate the absolute risk reduction as 1.82%, similar to the 1.9% in the original publication. We are able to obtain the 95% CI [0.80, 2.84] for this based on the reconstructed data. Using the Cox proportional hazards model, we obtain the hazard-ratio estimate 0.81 (95% CI: [0.72, 0.91]). Table 3 shows that the correction of monotonicity violators by isotonic regression and by the default method lead to very similar results.

Table 3. Example 3. Comparison of summary measures estimated from publication and their corresponding reconstructed data

	Publication	<code>ipdfc</code> with <code>isotonic</code>	<code>ipdfc</code> without <code>isotonic</code>
Log-rank test	$P = 0.0003$	$P = 0.0006$	$P = 0.0006$
Absolute risk reduction (95% CI)	1.9%	1.82% [0.80, 2.84]	1.80% [0.80, 2.82]
Hazard ratio (95% CI)	not applicable	0.81 [0.72, 0.91]	0.81 [0.72, 0.92]

The availability of Kaplan–Meier curves has enabled us to reconstruct the time-to-event data and calculate the hazard ratio, which was not reported for this trial. This would be particularly helpful if this trial was included in a meta-analysis where the hazard ratio is used as an effect measure.

5 Discussion

In this article, we provide a command, `ipdfc`, to implement the algorithm of reconstructing time-to-event data based on the information extracted from published Kaplan–Meier curves. Our command has greater flexibility, incorporating several additional features. It requires fewer inputs, automatically corrects data inconsistency that violates monotonicity, and allows one to use the number of patients at risk at the final interval, if reported.

Example 1 shows that the estimates of summary statistics (table 1) based on `ipdfc` are similar to those by Guyot et al. (2012). Some estimates are better approximations than others. The approximations to median times to event are very close to those in the original publication (Perren et al. 2011). The approximated hazard ratio is also close, but not identical, to that reported in the original publication. This small discrepancy is possibly due to the numbers and positioning of events not being entirely accurately estimated by the algorithm.

Although nonproportional hazards are evident in ICON7, the reconstructed Kaplan–Meier curves and hazard-ratio estimate are in reasonable agreement with those from

the trial publication (see table 2). This suggests that nonproportional hazards may not much affect the approximation accuracy. However, further empirical evaluation of `ipdfc` in a larger number of trials, with or without obvious nonproportional hazards, is desirable; this is a topic for further research.

Where hazard ratios are not reported but Kaplan–Meier curves are available, `ipdfc` is particularly helpful because it enables the reconstruction of time-to-event data and hence allows for reanalysis of the data. For the EUROPA trial, we are able to obtain the estimate of the hazard ratio and obtain the 95% CI for the absolute risk reduction, both of which were not reported in the trial publication. It is shown that the recovered Kaplan–Meier curves and the associated risk table are both very similar to the originals. This is perhaps due to the large sample size in this trial, and the accuracy of the reconstructed data increases accordingly.

We conclude that `ipdfc` appears to perform quite well in regenerating survival data, sufficient to produce reasonable approximations to summary statistics in time-to-event analysis.

6 Acknowledgments

Patrick Royston was supported by the UK Medical Research Council (MRC) grant to the MRC Clinical Trials Unit Hub for Trials Methodology Research (grant number MSA7355QP21).

7 References

- Barlow, R. E., D. J. Bartholomew, J. M. Bremner, and H. D. Brunk. 1972. *Statistical Inference Under Order Restrictions: Theory and Application of Isotonic Regression*. New York: Wiley.
- Bonner, J. A., P. M. Harari, J. Giralt, N. Azarnia, D. M. Shin, R. B. Cohen, C. U. Jones, R. Sur, D. Raben, J. Jassem, R. Ove, M. S. Kies, J. Baselga, H. Youssoufian, N. Amellal, E. K. Rowinsky, and K. K. Ang. 2006. Radiotherapy plus cetuximab for squamous-cell carcinoma of the head and neck. *New England Journal of Medicine* 354: 567–578.
- Deeks, J. J., J. P. T. Higgins, and D. G. Altman. 2008. Analysing data and undertaking meta-analyses. In *Cochrane Handbook for Systematic Reviews of Interventions*, ed. J. P. T. Higgins and S. Green, 243–296. Chichester, UK: Wiley.
- Fox, K. M. 2003. Efficacy of perindopril in reduction of cardiovascular events among patients with stable coronary artery disease: Randomised, double-blind, placebo-controlled, multicentre trial (the EUROPA study). *Lancet* 362: 782–788.
- Guyot, P., A. E. Ades, M. J. N. M. Ouwens, and N. J. Welton. 2012. Enhanced secondary analysis of survival data: Reconstructing the data from published Kaplan–Meier survival curves. *BMC Medical Research Methodology* 12: 9.

800 *Reconstructing time-to-event data from published Kaplan–Meier curves*

- Parmar, M. K. B., V. Torri, and L. Stewart. 1989. Extracting summary statistics to perform meta-analyses of the published literature for survival endpoints. *Statistics in Medicine* 17: 2815–2834.
- Perren, T. J., A. M. Swart, J. Pfisterer, J. A. Ledermann, E. Pujade-Lauraine, G. Kristensen, M. S. Carey, P. Beale, A. Cervantes, C. Kurzeder, A. du Bois, J. Sehouli, R. Kimmig, A. Stähle, F. Collinson, S. Essapen, C. Gourley, A. Lortholary, F. Selle, M. R. Mirza, A. Leminen, M. Plante, D. Stark, W. Qian, M. K. B. Parmar, and A. M. Oza. 2011. A phase 3 trial of bevacizumab in ovarian cancer. *New England Journal of Medicine* 365: 2484–2496.
- van Putten, W., and P. Royston. 2017. irax: Stata module to perform isotonic regression analysis. Statistical Software Components S458406, Department of Economics, Boston College. <https://ideas.repec.org/c/boc/bocode/s458406.html>.
- Royston, P., and M. K. B. Parmar. 2011. The use of restricted mean survival time to estimate the treatment effect in randomized clinical trials when the proportional hazards assumption is in doubt. *Statistics in Medicine* 30: 2409–2421.
- Tierney, J. F., L. A. Stewart, D. Ghersi, S. Burdett, and M. R. Sydes. 2007. Practical methods for incorporating summary time-to-event data into meta-analysis. *Trials* 8: 16.
- Wei, Y., P. Royston, J. F. Tierney, and M. K. B. Parmar. 2015. Meta-analysis of time-to-event outcomes from randomized trials using restricted mean survival time: Application to individual participant data. *Statistics in Medicine* 34: 2881–2898.
- Williamson, P. R., C. T. Smith, J. L. Hutton, and A. G. Marson. 2002. Aggregate data meta-analysis with time-to-event outcomes. *Statistics in Medicine* 21: 3337–3351.

About the authors

Yinghui Wei is a statistician with research interests including statistical methods for medicine and health as well as statistical computing and algorithms. Her current work centers on survival analysis, meta-analysis, hierarchical models, and infectious diseases epidemiology.

Patrick Royston is a medical statistician with 40 years of experience, with a strong interest in biostatistical methods and in statistical computing and algorithms. He works largely in methodological issues in the design and analysis of clinical trials and observational studies. He is currently focusing on alternative outcome measures and tests of treatment effects in trials with a time-to-event outcome, on parametric modeling of survival data, and on novel clinical trial designs.

Appendix

Algorithm 1 Reconstructing survival data (adapted from Guyot et al. [2012])

Require: The data extracted from published survival curves.

S_k : survival percentages as extracted from y axis, $k = 1, \dots, K$, where K is the total number of extracted data points

t_k : time from randomization as extracted from x axis

$nrisk_i$: number of patients at risk at time $trisk_i$, $i = 1, \dots, T$, where T is the number of intervals where the number of patients at risk is reported

$trisk_i$: time reported at the risk table

Ensure: $S_{k+1} \leq S_k$ for all k to meet the monotonicity constraint.

Set $lower_i = \min\{k : t_k \geq trisk_i\}$ and $upper_i = \max\{k : t_k \leq trisk_{i+1}\}$.

if $i < T - 1$ and $T > 1$ **then**

Step 1. Calculate \widehat{nc}_i , the number of censored at time $[trisk_i, trisk_{i+1}]$, by

$$\widehat{nc}_i = S_{lower_{i+1}}/S_{lower_i} \times nrisk_i - nrisk_{i+1}$$

Step 2. Distribute \widehat{nc}_i evenly within $[trisk_i, trisk_{i+1}]$. The censored time is then

$$ctime_c = t_{lower_i} + c \times (t_{lower_{i+1}} - t_{lower_i}) / (\widehat{nc}_i + 1)$$

where $c = 1, \dots, \widehat{nc}_i$. We can then calculate the number of censored events, \widehat{nc}_k , in extracted intervals $[t_k, t_{k+1}]$, which is within $[trisk_i, trisk_{i+1}]$.

Step 3. Calculate the number of events at t_k as

$$\widehat{nd}_k = \widehat{n}_k \times \left(1 - S_k / \widehat{S}_{last(k)}^{KM}\right)$$

\widehat{n}_k is the estimated number at risk at time t_k . $\widehat{S}_{last(k)}^{KM}$ is the estimated survival probability at time $t_{last(k)}$ with

$$last(k) = \begin{cases} 1 & \text{if } k = 1 \\ k' & \text{otherwise} \end{cases}$$

Note that $t_{k'} \leq t_k$, k' is such that the latest event occurs at $t_{k'}$, and there are no events in $(t_{k'}, t_k)$. The estimated number of patients at risk at time t_{k+1} is then $\widehat{n}_{k+1} = \widehat{n}_k - \widehat{nd}_k - \widehat{nc}_k$, where $k \in [lower_i, upper_i]$. Thus, $\widehat{nrisk}_{i+1} = \widehat{n}_{upper_i+1}$.

802 *Reconstructing time-to-event data from published Kaplan–Meier curves*

Step 4. Set $\Delta_t = \widehat{nrisk}_{i+1} - nrisk_{i+1}$.

if $\Delta_t \neq 0$ **then**

Adjust the estimated number of censored in time interval $[trisk_i, trisk_{i+1}]$ by setting

$$\widehat{nc}_i = \widehat{nc}_i + \left(\widehat{nrisk}_{i+1} - nrisk_{i+1} \right)$$

We then repeat steps 1–4 until $\widehat{nrisk}_{i+1} = nrisk_{i+1}$.

end if

Step 5. Repeat steps 1–4 until $i + 1 = T$.

end if

if $i = T$ or $i = 1$ and $T = 1$ **then**

Step 6. Approximate \widehat{nc}_T within interval $[trisk_{T-1}, trisk_T]$ by setting

$$\widehat{nc}_T = \min \left(\frac{t_{\text{upper}_T} - t_{\text{lower}_T}}{t_{\text{upper}_{T-1}} - t_{\text{lower}_{T-1}}} \times \sum_{i=1}^{T-1} \widehat{nc}_i; nrisk_T \right)$$

We then run steps 2–3 for the last interval $[trisk_{T-1}, trisk_T]$.

end if

if the total number of events, D , is not given **then**

Stop the algorithm.

end if

if the total number of events, D , is given **then**

Step 7. Compute $\sum_{k=1}^{\text{upper}_{T-1}} \widehat{nd}_k$.

if $\sum_{k=1}^{\text{upper}_{T-1}} \widehat{nd}_k \geq D$ **then**

Stop the algorithm.

end if

if $\sum_{k=1}^{\text{upper}_{T-1}} \widehat{nd}_k < D$ **then**

Step 8. Adjust the number of censored, \widehat{nc}_T , by setting

$$\widehat{nc}_T = \widehat{nc}_T + \left(\sum_{k=1}^{\text{upper}_T} \widehat{nd}_k - D \right)$$

Repeat steps 2–3 and steps 7–8 for the last interval.

end if

end if

Identification and estimation of treatment effects in the presence of (correlated) neighborhood interactions: Model and Stata implementation via `ntreatreg`

Giovanni Cerulli
CNR-IRCrES
National Research Council of Italy
Research Institute on Sustainable Economic Growth
Rome, Italy
giovanni.cerulli@ircres.cnr.it

Abstract. In this article, I present a counterfactual model identifying average treatment effects by conditional mean independence when considering peer- or neighborhood-correlated effects, and I provide a new command, `ntreatreg`, that implements such models in practical applications. The model and its accompanying command provide an estimation of average treatment effects when the stable unit treatment-value assumption is relaxed under specific conditions. I present two instructional applications: the first is a simulation exercise that shows both model implementation and `ntreatreg` correctness; the second is an application to real data, aimed at measuring the effect of housing location on crime in the presence of social interactions. In the second application, results are compared with a no-interaction setting.

Keywords: `st0499`, `ntreatreg`, ATEs, Rubin’s causal model, SUTVA, neighborhood effects

1 Introduction

In observational program evaluation studies, aimed at estimating the effect of an intervention on the outcome of a set of targeted individuals, it is generally assumed that “the treatment received by one unit does not affect other units’ outcome” (Cox 1958). Along with other fundamental assumptions—such as the conditional independence assumption, the exclusion restriction provided by instrumental-variables estimation, or the existence of a “forcing” variable in regression discontinuity design—the no-interference assumption is additionally invoked to consistently estimate average treatment effects (ATEs). It is thus clear that, if interference (or interaction) among units is not properly accounted for, traditional program evaluation methods such as regression adjustment,

selection models, matching, or reweighting are bound to be biased estimations of the actual treatment effect (TE).¹

Rubin (1978) calls this important assumption a stable unit treatment-value assumption (SUTVA), whereas Manski (2013) calls it “individualistic treatment response” to emphasize that it restricts the form of the treatment response function that the analyst considers. SUTVA (or individualistic treatment response) implies that the treatment applied to a specific individual affects only the outcome of that individual. This means that potential “externality effects” flowing, for instance, from treated to untreated subjects are strictly ruled out.

This article is an attempt to partially relax this assumption; by excluding the alternative, it operationalizes the estimation of ATEs when peer effects are assumed to flow from treated to untreated units. This restriction, reasonable in specific contexts, allows for straightforward identification and estimation of TEs simply by invoking conditional mean independence (CMI). Although demanding, this restriction seems a valuable attempt to weaken SUTVA, even though its complete removal would require a more general approach.

Some epidemiological studies have addressed the interference topic by restricting the analysis to experimental settings with randomized treatment (see, for instance, Rosenbaum [2007]; Hudgens and Halloran [2008]; Tchetgen Tchetgen and VanderWeele [2010]; and Robins, Hernán, and Brumback [2000]). However, in this article, I move along the line traced by econometric studies, normally dealing with nonexperimental settings with sample selection (that is, no random assignment to treatment is assumed). Thus, an ex-post evaluation is envisaged (Sobel 2006). In particular, I work within the binary POM that I attempt to partially generalize to account for the presence of neighborhood effects. My theoretical reference draws upon previous works dealing with TE identification in the presence of peer effects, particularly the works by Manski (1993, 2013).

I provide a new community-contributed command, `ntreatreg`, to operationalize the estimation of the suggested model in practical applications. Stata provides a powerful package, `teffects`, for estimating TEs for observational data. `teffects` can also estimate many valuable community-contributed TE routines available for similar and more advanced purposes. However, neither official nor community-contributed commands have been provided so far to incorporate peer effects in TE estimation. The `ntreatreg` command is a first attempt of this incorporation. Therefore, it is a valuable tool for estimating ATEs for observational data when the SUTVA is relaxed according to specific conditions. Such conditions often characterize some biomedical and socioeconomic contexts of application.

1. The applied literature on the socioeconomics of peer effect is rather vast; here we focus on that related to peer (or neighborhood) effects within Rubin’s potential outcome model (POM). Recently, however, Angrist (2014) has provided a comprehensive critical review of problems arising in measuring the causal effect of a peer regressor on individual performance. His article also provides a brief survey of the literature on the subject.

This article is organized as follows: Section 2 presents some related literature and positions my approach within it. Section 3 sets out the model, its assumptions, and its propositions. Section 4 presents the model's estimation procedure. Section 5 puts forward the software implementation of the model via the community-contributed command `ntreatreg` and provides a simulative illustration by setting out the model's data-generating process (DGP); section 5.7 presents a utility command, `mkomega`, for when users want to compute a similarity matrix based on a list of covariates to be then inserted into `ntreatreg`. Section 6 illustrates an application of `ntreatreg` to real data by investigating the effect of housing location on crime; here a comparison with a no-interaction setting is also performed by using the companion routine `ivtreatreg` (Cerulli 2014). Section 7 concludes the article. Appendix A then sets out the proof of each proposition.

2 Related literature

The literature on the estimation of TEs under potential interference among units is a recent and challenging field of statistical and econometric study. So far, however, few articles have dealt formally with this relevant topic (Angrist 2014).

Rosenbaum (2007) was among the first scholars to pave the way to generalize the standard randomization statistical approach for comparing different treatments with the case of units' interference. He presented a statistical model in which a unit's response depends not only on the treatment individually received but also on the treatment received by other units, thus showing how it is possible to test the null hypothesis of no interference in a random assignment setting where randomization occurs within prespecified groups and interference between groups is ruled out.

In the same vein, Sobel (2006) provided a definition, identification, and estimation strategy for traditional ATE estimators when interference between units is allowed by using the Moving to Opportunity for Fair Housing randomized social experiment as an example. In his article, he interchangeably uses the terms "interference" and "spillover" to account for the presence of such externalities. Interestingly, he shows that a potential bias can arise when no interference is erroneously assumed, and he defines a series of direct and indirect TEs that may be identified under reasonable assumptions. Additionally, he shows some interesting links between the form of his estimators under interference and the local ATE estimator provided by Imbens and Angrist (1994), thus showing that—under interference—TEs can be identified only on specific subpopulations.

The article by Hudgens and Halloran (2008) is probably the most relevant of this literature, because these authors develop a rather general and rigorous modeling of the statistical treatment setting under randomization when interference is potentially present. Furthermore, their approach also paves the way for extensions to observational settings. Starting from the same two-stage randomization approach of Rosenbaum (2007), these authors manage to go substantially further by providing a precise characterization of the causal effects with interference in randomized trials also encompassing Sobel's approach. They define direct, indirect, total, and overall causal effects, showing the relation be-

tween these measures and providing an unbiased estimator of the upper bound of their variance.

Tchetgen Tchetgen and VanderWeele’s (2010) article follows in the footsteps traced by the approach of Hudgens and Halloran (2008), providing a formal framework for statistical inference on population-average causal effects in a finite-sample setting with interference when the outcome variable is binary. Interestingly, they also present an original inferential approach for observational studies based on a generalization of the inverse-probability weighting estimator when interference is present.

Aronow and Samii (2013) generalize the approach proposed by Hudgens and Halloran (2008), going beyond the hierarchical experiment setting and providing a general variance estimation, including covariates adjustment.

Previous literature assumes that the potential outcome y of unit i is a function of the treatment received by such a unit (w_i) and the treatment received by all the other units (w_{-i}), that is,

$$y_i(w_i; w_{-i}) \tag{1}$$

which entails that—with N units and a binary treatment, for instance—a number of 2^N potential outcomes may arise. Nevertheless, an alternative way of modeling unit i ’s potential outcome is assuming

$$y_i(w_i; y_{-i}) \tag{2}$$

where y_{-i} is the $(N - 1) \times 1$ vector of other units’ potential outcomes, excluding unit i ’s potential outcome.² The notion of interference entailed by expression (2) is different from that implied by expression (1). The latter, however, is consistent with the notion of “endogenous” neighborhood effects provided by Manski (1993, 532–533). Manski, in fact, identifies three types of effects corresponding to three arguments of the individual (potential) outcome equation incorporating social effects:³

1. *Endogenous effects.* These effects entail that the outcome of an individual depends on the outcomes of other individuals belonging to the same neighborhood.
2. *Exogenous (or contextual) effects.* These effects concern the possibility that the outcome of an individual is affected by the exogenous idiosyncratic characteristics of the individuals belonging to the same neighborhood.

2. A combined regression model, including both individual treatments and outcomes, may be expressed as

$$y_i = f(w_i; y_{-i}; w_{-i})$$

Arduini, Patacchini, and Rainone (2014) provide a first attempt to modeling such a regression on individuals eligible for treatment, showing that the coefficient of w_i (that is, their measure of ATE) combines both treatments’ and outcomes’ direct and indirect effects on y . However, such a model is not embedded within the classical Rubin POM. I instead provide a POM-consistent approach generalized to the case of possible interaction among units.

3. The literature is not homogeneous in singling out a unique name of such effects; dependent on context, authors interchangeably refer to peer, neighborhood, social, club, interference, or interaction effects.

3. *Correlated effects.* These effects are due to belonging to a specific group and thus sharing some institutional or normative condition (that one can loosely define as “environment”).

Contextual and correlated effects are exogenous because they clearly depend on predetermined characteristics of the individuals in the neighborhood (case 2) or of the neighborhood itself (case 3). Endogenous effects are of broader interest because they are affected by the behavior (measured as “outcome”) of other individuals involved in the same neighborhood. This means that endogenous effects comprise both direct and indirect effects linked to a given external intervention on individuals.

The model proposed in this article assumes that the potential untreated outcome depends on treated units’ potential outcomes. However, because the latter are assumed to depend on a set of observable exogenous confounders in the presence of uncorrelated unobservables, this model fits only “correlated effects” as defined in Manski’s taxonomy.

To concisely position this article within the literature, I will say that previous contributions assume the following:

- i) The unit potential outcome depends on its own treatment and other units’ treatment.
- ii) The assignment is randomized or conditionally unconfounded.
- iii) The treatment is multiple.
- iv) Potential outcomes have a nonparametric form.

In this article, I instead assume the following:

- i) The unit potential outcome depends on its own treatment and other units’ potential outcome.
- ii) The assignment is mean conditionally unconfounded.
- iii) The treatment is binary.
- iv) Potential outcomes have a parametric form.

The model developed here is part of a broader class of regression-adjustment TE models. These models are suitable for observational studies, though it should be recognized that regression adjustment does not provide consistent estimates if a “killing” unobservable confounder is at work. The conditional unconfoundedness assumption (ii, above) upon which the estimator relies is shared by other wellknown estimators, such as matching and (inverse-probability) reweighting. In this sense, assuming absence of correlation between the treatment variables and the one measuring neighborhood is essential to the identification of this type of model. Therefore, in this article, I suggest a simple but workable way to relax SUTVA, one that seems easy to implement in many biomedical and socioeconomic contexts of application.

3 A binary treatment model with “correlated” neighborhood effects

This section presents a model for fitting the ATEs of a policy program (or a treatment) in a nonexperimental setting in the presence of “correlated” neighborhood (or externality) interactions. We consider a binary treatment variable w —taking a value of 1 for treated and 0 for untreated units—that is assumed to affect an outcome (or target) variable y that can take a variety of forms.

Some notation can help in understanding the setting: N is the number of units involved in the experiment; N_1 is the number of treated units; N_0 is the number of untreated units; w_i is the treatment variable assuming a value of 1 if unit i is treated and 0 if it is untreated; y_{1i} is the outcome of unit i when the individual is treated; y_{0i} is the outcome of unit i when the individual is untreated; $\mathbf{x}_i = (x_{1i}, x_{2i}, x_{3i}, \dots, x_{Mi})$ is a row vector of M exogenous observable characteristics for unit $i = 1, \dots, N$.

To begin with, as usual in this literature, we define unit i 's TE as

$$\text{TE}_i = y_{1i} - y_{0i} \quad (3)$$

TE_i is equal to the difference between the value of the target variable when the individual is treated (y_1) and the value assumed by this variable when the same individual is untreated (y_0). Because TE_i refers to the same individual at the same time, the analyst can observe just one of the two quantities feeding into (3), never both. For instance, it might be the case that we can observe the investment behavior of a supported company, but we cannot know what the investment of this company would have been had it not been supported and vice versa. The analyst faces a fundamental missing observation problem (Holland 1986) that needs to be tackled econometrically to reliably recover the causal effect via some specific imputation technique (Rubin 1974, 1977).

The random pair (y_{1i}, y_{0i}) is assumed to be independent and identically distributed across all units i , and both y_{1i} and y_{0i} are generally explained by a structural part depending on observable factors and a nonstructural part depending on an unobservable (error) term. Nevertheless, recovering the entire distributions of y_{1i} and y_{0i} (and, consequently, the distribution of TE_i) may be too demanding without very strong assumptions. Therefore, the literature has focused on estimating specific moments of these distributions, particularly the “mean”, thus defining the so-called population ATE and ATE conditional on \mathbf{x}_i [that is, $\text{ATE}(\mathbf{x}_i)$] of a policy intervention as

$$\text{ATE} = E(y_{i1} - y_{i0}) \quad (4)$$

$$\text{ATE}(\mathbf{x}_i) = E(y_{i1} - y_{i0} | \mathbf{x}_i) \quad (5)$$

where $E(\cdot)$ is the mean operator. ATE is equal to the difference between the average of the target variable when the individual is treated (y_1) and the average of the target variable when the same individual is untreated (y_0). Observe that, by the law of iterated expectations, $\text{ATE} = E_{\mathbf{x}}\{\text{ATE}(\mathbf{x})\}$.

Given the definitions of the unconditional and conditional ATE in (4) and (5), respectively, one can define the same parameters in the subpopulation of treated (ATET) and untreated (ATENT) units, that is,

$$\begin{aligned} \text{ATET} &= E(y_{i1} - y_{i0} | w_i = 1) \\ \text{ATET}(\mathbf{x}_i) &= E(y_{i1} - y_{i0} | \mathbf{x}_i, w_i = 1) \end{aligned}$$

and

$$\begin{aligned} \text{ATENT} &= E(y_{i1} - y_{i0} | w_i = 0) \\ \text{ATENT}(\mathbf{x}_i) &= E(y_{i1} - y_{i0} | \mathbf{x}_i, w_i = 0) \end{aligned}$$

In this article, I aim to provide consistent parametric estimation of all previous quantities (ATEs) in the presence of neighborhood effects.

To that end, let us start with what is observable to the analyst in such a setting, that is, the actual status of the unit i , which can be obtained as

$$y_i = y_{0i} + w_i(y_{1i} - y_{0i}) \quad (6)$$

Equation (6) is Rubin's POM, which is the fundamental relation linking the unobservable to the observable outcome. Given (6), we first set out all the assumptions behind the next development of the proposed model.

- *Assumption 1. Unconfoundedness (or CMI).* Given the set of random variables $\{y_{0i}, y_{1i}, w_i, \mathbf{x}_i\}$ as defined above, the following equalities hold:

$$E(y_{gi} | w_i, \mathbf{x}_i) = E(y_{ig} | \mathbf{x}_i) \quad \text{with } g = 0, 1$$

Hence, throughout this article, we will assume unconfoundedness (that is, CMI) to hold. As we will see, CMI is a sufficient condition for identifying ATEs also when neighborhood effects are considered.

Once CMI has been assumed, we then need to properly model the potential outcomes y_{0i} and y_{1i} to get a representation of the ATEs (that is, ATE, ATET, and ATENT) while accounting for the presence of correlated externality effects. In this article, we will simplify our analysis further by assuming some restrictions in the form of the potential outcomes.

- *Assumption 2. Restrictions on the form of the potential outcomes.* Consider the general form of the potential outcome as expressed in (2) and assume this relation to depend parametrically on a vector of real numbers $\boldsymbol{\theta} = (\boldsymbol{\theta}_0; \boldsymbol{\theta}_1)$. We assume that

$$y_{1i}(w_i; \mathbf{x}_i; \boldsymbol{\theta}_1)$$

and

$$y_{0i}(w_i; \mathbf{x}_i; y_{1,-i}; \boldsymbol{\theta}_0)$$

Assumption 2 poses two important restrictions to the form given to the potential outcomes: i) it makes them dependent on some unknown parameters θ (that is, the parametric form), and ii) it entails that the externality effect occurs only in one direction, that is, from the treated individuals to the untreated, while excluding the alternative.⁴

- *Assumption 3. Linearity and weighting matrix.* We assume that the potential outcomes are linear in the parameters and that an $N \times N$ weighting matrix Ω of exogenous constant numbers is known.

Under assumptions 1, 2, and 3, the model takes on the form

$$\begin{cases} y_{1i} = \mu_1 + \mathbf{x}_i\beta_1 + e_{1i} \\ y_{0i} = \mu_0 + \mathbf{x}_i\beta_0 + \gamma s_i + e_{0i} \\ s_i = \sum_{j=1}^{N_1} \omega_{ij}y_{1j}, \quad \text{with} \quad \sum_{j=1}^{N_1} \omega_{ij} = 1 \\ y_i = y_{0i} + w(y_{1i} - y_{0i}) \\ \text{CMI holds} \end{cases} \quad (7)$$

where $i = 1, \dots, N$ and $j = 1, \dots, N_1$; μ_1 and μ_0 are scalars; β_0 and β_1 are two unknown vector parameters defining the different response of unit i to the vector of covariates \mathbf{x} ; e_0 and e_1 are two random errors with an unconditional mean of 0 and a constant variance; and s_i represents the unit i th neighborhood effect because of the treatment administered to unit j ($j = 1, \dots, N_1$). Observe that, by linearity,⁵ we have

$$s_i = \begin{cases} \sum_{j=1}^{N_1} \omega_{ij}y_{1j} & \text{if } i \in \{w = 0\} \\ 0 & \text{if } i \in \{w = 1\} \end{cases} \quad (8)$$

where the parameter ω_{ij} is the generic element of the weighting matrix Ω expressing some form of distance between unit i and unit j . Although not strictly required for consistency, we also assume that these weights sum to 1, that is, $\sum_{j=1}^{N_1} \omega_{ij} = 1$. In short, previous assumptions say that unit i 's neighborhood effect takes the form of a weighted mean of the outcomes of treated units and that this "social" effect has an impact only on unit i 's outcome when this unit is untreated. As a consequence, by substitution of (8) into (7), we get that

$$y_{0i} = \mu_0 + \mathbf{x}_i\beta_0 + \gamma \sum_{j=1}^{N_1} \omega_{ij}y_{1j} + e_{0i}$$

4. In the more general case in which peer effects take place from treated to untreated units and vice versa, identifying ATEs consistent with the POM becomes trickier because various feedback terms do arise. Using a spatial regression approach, [Arduini, Patacchini, and Rainone \(2014\)](#) estimate a TE reduced form that also includes feedback terms. However, their model is not directly derived using the POM, unlike the model in this article.

5. The linearity of the spillover effect is an assumption needed to simplify the subsequent regression analysis. However, nonlinear forms might also be used. Some sensitivity analysis could show how results change according to different mathematical forms of the spillover effect.

making it clear that untreated unit i 's outcome is a function of its own idiosyncratic characteristics (\mathbf{x}_i), the weighted outcomes of treated units multiplied by a sensitivity parameter γ , and a standard error term.

Let us now consider four propositions implied by the previous assumptions.

- *Proposition 1. Formula of ATE with neighborhood interactions.* Given assumptions 2 and 3 and the implied equations established in (7), the ATE with neighborhood interactions takes on the form

$$\begin{aligned} \text{ATE} &= E(y_{1i} - y_{0i}) = \mu + E \left\{ \mathbf{x}_i \boldsymbol{\delta} - \left(\sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j \right) \gamma \boldsymbol{\beta}_1 - e_i \right\} \\ &= \mu + \bar{\mathbf{x}} \boldsymbol{\delta} - \bar{\mathbf{v}} \boldsymbol{\lambda} \end{aligned} \quad (9)$$

where $\boldsymbol{\lambda} = \gamma \boldsymbol{\beta}_1$, $\bar{\mathbf{x}} = E(\mathbf{x}_i)$, $\bar{\mathbf{v}} = E(\underbrace{\sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j}_{\mathbf{v}_i})$ is the unconditional mean of the vector \mathbf{x}_i , and $\mu = \mu_1 - \mu_0 - \gamma \mu_1$. The proof is in appendix A.1.

Indeed, by the definition of ATE as given in (4) and by (7), we can immediately show that, for such a model,

$$\begin{aligned} \text{ATE} &= E(y_{1i} - y_{0i}) \\ &= E \left\{ (\mu_1 + \mathbf{x}_i \boldsymbol{\beta}_1 + e_{1i}) - \left(\mu_0 + \mathbf{x}_i \boldsymbol{\beta}_0 + \gamma \sum_{j=1}^{N_1} \omega_{ij} y_{1j} + e_{0i} \right) \right\} \end{aligned} \quad (10)$$

where

$$\begin{aligned} \sum_{j=1}^{N_1} \omega_{ij} y_{1j} &= \sum_{j=1}^{N_1} \omega_{ij} (\mu_1 + \mathbf{x}_j \boldsymbol{\beta}_1 + e_{1j}) \\ &= \mu_1 \sum_{j=1}^{N_1} \omega_{ij} + \sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j \boldsymbol{\beta}_1 + \sum_{j=1}^{N_1} \omega_{ij} e_{1j} \\ &= \mu_1 + \left(\sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j \right) \boldsymbol{\beta}_1 + \sum_{j=1}^{N_1} \omega_{ij} e_{1j} \end{aligned} \quad (11)$$

By developing ATE further using (11), we finally get the result in (10).

- *Proposition 2. Formula of ATE(\mathbf{x}_i) with neighborhood interactions.* Given assumptions 2 and 3 and the result in proposition 1, we have that

$$\text{ATE}(\mathbf{x}_i) = E(y_{1i} - y_{0i} | \mathbf{x}_i) = \text{ATE} + (\mathbf{x}_i - \bar{\mathbf{x}}) \boldsymbol{\delta} + (\bar{\mathbf{v}} - \mathbf{v}_i) \boldsymbol{\lambda} \quad (12)$$

where it is now easy to see that $\text{ATE} = E_{\mathbf{x}}\{\text{ATE}(\mathbf{x})\}$. The proof is in appendix A.2.

- *Proposition 3. Baseline random-coefficient regression.* By substitution of (7) into the POM of (6), we obtain the random-coefficient regression model (Wooldridge 1997)

$$y_i = \eta + w_i \times \text{ATE} + \mathbf{x}_i \boldsymbol{\beta}_0 + w_i (\mathbf{x}_i - \bar{\mathbf{x}}) \boldsymbol{\delta} + \mathbf{z}_i \boldsymbol{\lambda} + e_i \quad (13)$$

where $\mathbf{z}_i = \mathbf{v}_i + w_i(\bar{\mathbf{v}} - \mathbf{v}_i)$, $\mathbf{v}_i = \sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j$, $\bar{\mathbf{v}} = 1/N \sum_{i=1}^N (\sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j)$, $\boldsymbol{\lambda} = \gamma \boldsymbol{\beta}_1$, $\eta = \mu_0 + \gamma \mu_1$, and $\boldsymbol{\delta} = \boldsymbol{\beta}_1 - \boldsymbol{\beta}_2$. The proof is in appendix A.3.

- *Proposition 4. Ordinary least-squares (OLS) consistency.* Under assumptions 1 (CMI), 2, and 3, the error term of regression (12) has a mean of 0 conditional on (w_i, \mathbf{x}_i) , that is,

$$\begin{aligned} E(e_i | w_i, \mathbf{x}_i) &= E \left\{ \gamma \sum_{j=1}^{N_1} \omega_{ij} e_{1j} + e_{0i} + w_i (e_{1i} - e_{0i}) - w_i \gamma \sum_{j=1}^{N_1} \omega_{ij} e_{1j} | w_i, \mathbf{x}_i \right\} \\ &= 0 \end{aligned}$$

thus implying that (12) is a regression model with parameters that can be consistently estimated by OLS. The proof is in appendix A.4.

Once a consistent estimation of the parameters of (12) is obtained, we can estimate the ATE directly from the regression, and we can estimate $\text{ATE}(\mathbf{x}_i)$ by plugging the estimated parameters into (11). This is because the plug-in estimator of $\text{ATE}(\mathbf{x}_i)$ becomes a function of consistent estimates and thus becomes consistent itself:

$$\text{plim } \widehat{\text{ATE}}(\mathbf{x}_i) = \text{ATE}(\mathbf{x}_i)$$

where $\widehat{\text{ATE}}(\mathbf{x}_i)$ is the plug-in estimator of $\text{ATE}(\mathbf{x}_i)$. Observe, however, that the (exogenous) weighting matrix $\boldsymbol{\Omega} = [\omega_{ij}]$ needs to be provided in advance.

Once the formulas for $\widehat{\text{ATE}}$ and $\widehat{\text{ATE}}(\mathbf{x}_i)$ are available, it is also possible to recover the $\widehat{\text{ATE}}\overline{\text{T}}$ and the $\widehat{\text{ATE}}\overline{\text{NT}}$ as

$$\widehat{\text{ATE}}\overline{\text{T}} = \widehat{\text{ATE}} + \frac{1}{\sum_{i=1}^N w_i} \sum_{i=1}^N w_i \left\{ (\mathbf{x}_i - \bar{\mathbf{x}}) \widehat{\boldsymbol{\delta}} + (\bar{\mathbf{v}} - \mathbf{v}_i) \widehat{\boldsymbol{\lambda}} \right\}$$

and

$$\widehat{\text{ATE}}\overline{\text{NT}} = \widehat{\text{ATE}} + \frac{1}{\sum_{i=1}^N (1 - w_i)} \sum_{i=1}^N (1 - w_i) \left\{ (\mathbf{x}_i - \bar{\mathbf{x}}) \widehat{\boldsymbol{\delta}} + (\bar{\mathbf{v}} - \mathbf{v}_i) \widehat{\boldsymbol{\lambda}} \right\}$$

These quantities are functions of observable components and parameters consistently estimated by OLS (see the next section). Once these estimates are available, standard errors for $\widehat{\text{ATE}}\overline{\text{T}}$ and $\widehat{\text{ATE}}\overline{\text{NT}}$ can be correctly obtained via bootstrapping (see Wooldridge [2010, 911–919]).

4 Estimation

Starting from the previous section's results, I suggest a simple protocol for estimating ATEs. Given an independent and identically distributed sample of observed variables for each individual i ,

$$\{y_i, w_i, \mathbf{x}_i\} \quad \text{with } i = 1, \dots, N$$

1. Provide a weighting matrix $\mathbf{\Omega} = [\omega_{ij}]$ measuring some type of distance between the generic unit i (untreated) and unit j (treated);
2. Using OLS, fit a regression model of

$$y_i \quad \text{on} \quad \{1, w_i, \mathbf{x}_i, w_i(\mathbf{x}_i - \bar{\mathbf{x}}), \mathbf{z}_i\}$$

3. Obtain $\{\widehat{\beta}_0, \widehat{\delta}, \widehat{\gamma}, \widehat{\beta}_1\}$ and put them into the formulas of \widehat{ATEs} .

By comparing the formulas of the ATE with ($\gamma \neq 0$) and without ($\gamma = 0$) the neighborhood effect, we define the estimated neighborhood bias as

$$\begin{aligned} \text{Bias} &= |\text{ATE}_{\text{without}} - \text{ATE}_{\text{with}}| = |\gamma\mu_1 + \bar{\mathbf{v}}\boldsymbol{\lambda}| \\ &= \left| \gamma\mu_1 + \left\{ \frac{1}{N} \sum_{i=1}^N \left(\sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j \right) \right\} \boldsymbol{\lambda} \right| \end{aligned}$$

This is the bias arising when one neglects peer effect in assessing TE in observational studies: it depends on the weights employed, the average of the observable confounders considered in \mathbf{x} , and the magnitude of the coefficients γ and β_1 . Such bias may be positive or negative.

Furthermore, by defining

$$\gamma\beta_1 = \boldsymbol{\lambda}$$

it is also possible to determine whether this bias is statistically significant by simply testing the following null hypothesis:

$$H_0 : \lambda_1 = \lambda_2 = \dots = \lambda_M = 0$$

If this hypothesis is rejected, we cannot exclude that neighborhood effects are in place, thus significantly affecting the estimation of the causal parameters' ATEs. In a similar way, we can also obtain an estimation of the neighborhood bias for ATET and ATENT.

5 Implementation via `ntreatreg`

The previous model can easily be fit using the new command `ntreatreg`, which has the syntax given below.

5.1 Syntax

```

ntreatreg outcome treatment [varlist] [if] [in] [weight], spill(matrix)
  [hetero(varlist_h) conf(#) graphic save_graph(filename) vce(robust)
  const(noconstant) head(noheader) beta]

```

fweights, *iwweights*, and *pweights* are allowed; see [U] 11.1.6 **weight**.

5.2 Description

`ntreatreg` estimates ATEs under CMI when neighborhood interactions may be present. It incorporates such externalities within the traditional Rubin POM. As such, it provides an attempt to relax the SUTVA generally assumed in observational studies.

5.3 Options

`spill(matrix)` specifies the adjacent (weighted) matrix used to define the presence and strength of the units' relationships. It could be a distance matrix, with distance loosely defined as either vector or spatial. `spill()` is required.

`hetero(varlist_h)` specifies the variables over which one calculates the idiosyncratic $ATE(\mathbf{x})$, $ATET(\mathbf{x})$, and $ATENT(\mathbf{x})$, where $\mathbf{x} = \textit{varlist}_h$. The default is that the command fits the specified model without a heterogeneous average effect. *varlist_h* should be the same set or a subset of the variables specified in *varlist*.

`conf(#)` sets the confidence level equal to the number specified by the user. The default is `conf(95)`.

`graphic` requests a graphical representation of the density distributions of $ATE(\mathbf{x})$, $ATET(\mathbf{x})$, and $ATENT(\mathbf{x})$. It gives an outcome only if variables are specified in `hetero()`.

`save_graph(filename)` saves in *filename* the graph obtained with the option `graphic`.

`vce(robust)` allows for robust regression standard errors.

`const(noconstant)` suppresses the regression constant term.

`head(noheader)` suppresses the header.

`beta` reports standardized beta coefficients.

5.4 Remarks

`ntreatreg` creates the following variables:

`_ws_varname_h` is an additional regressor used in the model's regression when the option `hetero()` is specified.

`_z_varname_h` is a spillover additional regressor.

`_v_varname_h` is the first spillover component of `_z_varname_h`.

`_ws_v_varname_h` is the second spillover component of `_z_varname_h`.

`ATE_x` is an estimate of the idiosyncratic ATE given \mathbf{x} .

`ATET_x` is an estimate of the idiosyncratic ATET given \mathbf{x} .

`ATENT_x` is an estimate of the idiosyncratic ATENT given \mathbf{x} .

5.5 Stored results

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

Scalars	
<code>e(N_tot)</code>	total number of (used) observations
<code>e(N_treat)</code>	number of (used) treated units
<code>e(N_untreat)</code>	number of (used) untreated units
<code>e(ate)</code>	value of the ATE
<code>e(atet)</code>	value of the ATET
<code>e(atent)</code>	value of the ATENT

5.6 Simulation exercise

To provide an operational estimation of our model, we first perform an illustrative simulation exercise based on the DGP underlying the model and illustrated in (7). This step is useful, both to show that the model has a relatively simple computational counterpart and to test the syntactic and semantic correctness of `ntreatreg` as a command.

The code to properly reproduce equations in system (7) appears below. For illustrative purposes, and with no loss of generality, we consider a random treatment:

```
. ***** START SIMULATION *****
. *****
. * Step 1. Generate the matrix omega
. *****
. * Generate the matrix omega
. clear all
. set matsize 1000
. set obs 200
number of observations (_N) was 0, now 200
. set seed 10101
. generate w=rbinomial(1,0.5)
```

```

. gsort - w
. count if w==1
  100
. global N1=r(N)
. global NO=_N-$N1
. matrix def M=J(_N,_N,0)
. global N=_N
. forvalues i=1/$N {
  2. forvalues j=1/$N1 {
  3. matrix M[`i`,`j']=runiform()
  4. }
  5. }
. matrix define SUM=J(_N,1,0)
. forvalues i=1/$N {
  2. forvalues j=1/$N1 {
  3. matrix SUM[`i`,1] = SUM[`i`,1] + M[`i`,`j`]
  4. }
  5. }
. forvalues i=1/$N {
  2. forvalues j=1/$N1 {
  3. matrix M[`i`,`j']=M[`i`,`j']/SUM[`i`,1]
  4. }
  5. }
. matrix omega=M
. *****
. * Step 2. Define the models data generating process (DGP)
. *****
. * Declare a series of parameters
. scalar mu1=2
. scalar b11=5
. scalar b12=3
. scalar b13=9
. scalar e1=rnormal()
. scalar mu0=5
. scalar b01=7
. scalar b02=1
. scalar b03=6
. scalar e0=rnormal()
. generate x1=rnormal()
. generate x2=rnormal()
. generate x3=5+3*rnormal()
. scalar gamma=0.8
. * Sort the treatment so to have the ones first
. gsort - w
. * Generate y1
. generate y1 = mu1 + x1*b11 + x2*b12 + e1
. generate y1_obs=w*y1
. mkmat y1_obs, mat(y1_obs)
. * Generate s
. matrix s = omega*y1_obs

```

```

. svmat s
. * Generate y0 and finally y
. generate y0 = mu0 + x1*b01 + x2*b02 + gamma*s1 + e0
. generate y = y0 + w*(y1-y0)
. * Generate the treatment effect te
. generate te=y1-y0
. summarize te

```

Variable	Obs	Mean	Std. Dev.	Min	Max
te	200	-3.140086	2.849946	-12.89648	4.528738

```

. * Put the ATE into a scalar
. scalar ATE=r(mean)
. display ATE
-3.1400858
. *****
. * Step 3. Fit the model using ntreatreg
. *****
. * y: dependent variable
. * w: treatment
. * x: [x1; x2] are the covariates
. * Matrix of spillovers: OMEGA
. * Fit the model using ntreatreg
. set more off
. ntreatreg y w x1 x2, hetero(x1) spill(omega) graphic

```

Source	SS	df	MS	Number of obs	=	200
Model	8479.76569	6	1413.29428	F(6, 193)	=	1382.01
Residual	197.36846	193	1.02263451	Prob > F	=	0.0000
Total	8677.13415	199	43.6036892	R-squared	=	0.9773
				Adj R-squared	=	0.9765
				Root MSE	=	1.0113

```


```

y	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
w	-3.133691	.1453545	-21.56	0.000	-3.420379 -2.847004
x1	6.869514	.1046721	65.63	0.000	6.663066 7.075962
x2	2.052399	.0718827	28.55	0.000	1.910622 2.194175
_ws_x1	-1.955304	.146996	-13.30	0.000	-2.245228 -1.665379
_z_x1	7.800202	2.114743	3.69	0.000	3.629227 11.97118
_z_x2	-1.066327	1.683491	-0.63	0.527	-4.386729 2.254075
_cons	3.107643	.4468559	6.95	0.000	2.226295 3.988991

```

(200 real changes made)
(200 real changes made)
(200 real changes made)
(100 missing values generated)
(100 missing values generated)
. scalar ate_neigh = _b[w] // put ATE into a scalar
. display ate_neigh
-3.1336913

```

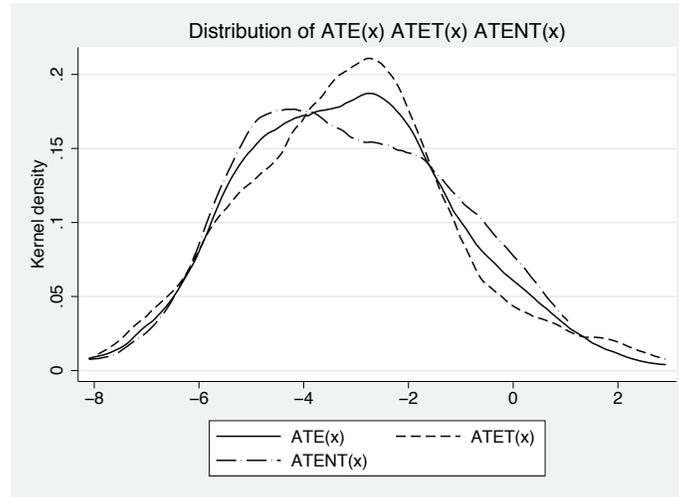


Figure 1. Resulting graph from the `ntreatreg` command with the `graphic` option

This simulation code deserves some comments:

- Step 1 provides a 200×200 matrix, Ω . This matrix is built by first generating a matrix \mathbf{M} of the same size as Ω from a $[0 - 1]$ uniform distribution and then dividing \mathbf{M} by its column sums. This last step is necessary to allow Ω to become a matrix of weights, as entailed by the third line of system (7).
- Step 2 reproduces the model DGP as defined in system (7) by giving the potential outcomes y_1 and y_0 a linear form. We first need to generate y_1 and, given this, the spillover variable s , which serves in turn as an explanatory variable for generating (with x_1 and x_2) the potential outcome of untreated units as entailed by the second line of system (7). Finally, by applying the potential outcome equation, we are able to generate the observable outcome of this process, y , and thus the TE for each unit (that is, the variable \mathbf{te}). Given this, the “true” DGP’s ATE is obtained as the mean of \mathbf{te} , which in this case is equal to -3.1401 .
- Step 3 fits the model generated by the previous DGP by using `ntreatreg`. If it is correct, we expect `ntreatreg` to provide a value of ATE close to the “true” one, that is, -3.1401 . We immediately see that `ntreatreg` estimates a statistically significant ATE equal to -3.1337 , which is strictly close to the one provided by our simulated DGP. By running the code several times, we obtain similar outcomes (not reported). We can conclude that `ntreatreg` is reliable both syntactically and semantically. Moreover, the `graphic` option of `ntreatreg` allows one to draw the distribution of $\text{ATE}(\mathbf{x})$, $\text{ATET}(\mathbf{x})$, and $\text{ATENT}(\mathbf{x})$ when SUTVA is relaxed according to the assumptions underlying the model. Finally, given the random nature of the treatment assumed in the simulated DGP, it is not surprising we discovered a similar shape for the distribution of $\text{ATE}(\mathbf{x})$, $\text{ATET}(\mathbf{x})$, and $\text{ATENT}(\mathbf{x})$.

5.7 The `mkomega` command

In the previous example, we provided the code to generate the neighborhood matrix Ω by simulation. This was useful to understand the correct form of Ω to insert into `ntreatreg`. However, a dedicated command to generate such a similarity matrix based on a set of variables (that is, covariates' vector distance) comes in handy. The command `mkomega` does this task, because it computes units' similarity matrices using the variables declared in `varlist`. Two types of similarity matrices are optionally computed by this routine: the correlation matrix and the inverse Euclidean distance matrix.⁶ The syntax of `mkomega` is set out below.

Syntax

```
mkomega treatment varlist [if] [in], sim_measure(type) out(outcome)
```

`treatment` is a binary variable taking a value of 1 for treated units and 0 for untreated ones. It is the same treatment variable that the user is going to specify in `ntreatreg`.

`varlist` is a list of numeric variables on which to build the distance measure. These variables should be of numeric significance, not categorical. Some of these variables might be specified as confounders in `ntreatreg`.

Description

`mkomega` computes a unit's similarity matrix using the variables declared in `varlist` to be later used in the command `ntreatreg`. Two types of similarity matrices are optionally allowed by this command: the correlation matrix and the inverse Euclidean distance matrix.

Options

`sim_measure(type)` specifies the similarity matrix to use. `sim_measure()` is required. `type` may be `corr`, for the correlation matrix, or `L2`, for the inverse Euclidean distance matrix.

`out(outcome)` specifies the outcome variable one is going to use in `ntreatreg`. `out()` is required.

6. Geographical distance can sometimes more suitably catch TE transmission. For this reason, the `mkomega` command provides the option `sim_measure(L2)` to calculate an (inverse) Euclidean distance matrix based on `varlist`. Indeed, if `varlist` is made of two variables identifying geographical coordinates (for instance, `X1` = latitude and `X2` = longitude), then using option `sim_measure(L2)` allows for an estimation of ATE adjusted for spillovers based on (pure) geographical distances.

Stored results

`mkomega` stores the following in `r()`:

Scalars	
<code>r(N1)</code>	number of treated units
<code>r(N0)</code>	number of untreated units
Matrices	
<code>r(M)</code>	similarity matrix

6 Application to real data: The effect of housing location on crime

In this application, we consider the dataset `spatial_columbus.dta` provided by [Anselin \(1988\)](#) containing information on property crimes in 49 neighborhoods in Columbus, Ohio, in 1980.

The aim of this illustrative application is to determine the effect of housing location on crimes, that is, the causal effect of the variable `cp`—taking a value of 1 if the neighborhood is located in the “core” of the city and 0 if it is located in the “periphery”—on the number of residential burglaries and vehicle thefts per thousand households (that is, the variable `crime`).

Several conditioning (or confounding) observable factors are included in the dataset. Here we consider only two of them: household income in \$1,000s (`inc`) and housing value in \$1,000s (`hval`).

We are interested in detecting the effect of housing location on the number of crimes in such a setting, by taking into account possible interactions among neighborhoods. Our research presumption is that the number of burglaries in a specific peripheral neighborhood is not only affected by the neighborhood’s idiosyncratic characteristics (the variables `inc` and `hval`) but also by the number of burglaries that occur in the core neighborhoods. The conjecture behind this statement is that a saturation effect may take place, with thieves more prone to move toward peripheral areas of the city when core areas become saturated. This may be due to, for instance, new installations of security systems in the core whenever a certain amount of burglaries occur.

To build the matrix Ω , we use the command `mkomega` with option type L2, which calculates the Euclidean distance matrix associated with the covariates `x` and `y`, representing respectively the longitude and latitude of the neighborhood. This matrix presents values that are greater than 0. The assumption here is that geographical distance properly catches interactions among different neighborhoods.

After running `mkomega` to generate Ω as in system (7), we consider the relation between `crime` and `cp` using `inc` and `hval` as regressors subject to observable heterogeneity. We then estimate the model presented here by comparing the estimates obtained using `ntreatreg`, which takes into account peer effects, with those obtained

using `ivtreatreg` (Cerulli 2014), which estimates the same model but without accounting for peer effects.

The coefficient of the treatment variable, `cp`, is equal to 14.6 in the regression incorporating peer effects and equal to 13.6 in the one not incorporating them. Both are significant. The adjusted R -squared is rather high and similar in the two regressions. The percentage bias is around 7%. By performing a test to see whether the coefficients of the peer effects are jointly 0 (that is, $H_0 : \gamma\beta_0 = 0$), we reject this null hypothesis, getting an F test equal to 5.78, which is highly significant (because the p -value is equal to 0.0061). This means that we cannot reject that peer effects are present in this example. We can also graphically compare the distribution of $ATE(\mathbf{x})$, $ATET(\mathbf{x})$, and $ATENT(\mathbf{x})$ with and without neighborhood interaction.

The whole Stata code showing all the steps needed to perform such estimates is reported below. Such code can be used as a general template for using `ntreatreg` to correctly apply the model presented in this article.

```
. ***** START IMPLEMENTATION *****
. *****
. * Step 1. Model inputs
. *****
. set scheme sj
. use "spatial_columbus", clear
. global y "crime"
. global w "cp"
. global xvars "inc hoval"
. global hvars "inc hoval"
. global dvars "x y"
. *****
. * Step 2. Deal with missing values
. *****
. * Eliminate common missing values
. quietly regress $y $w $xvars $hvars $dvars
. * Consider a dataset made of only nonmissing values
. keep if e(sample)
(0 observations deleted)
. * Sort the treatment (treated first)
. gsort - $w
. global N = r(N)
. count if $w==1
    24
. global N1 = r(N)
. global N0 = $N-$N1
. *****
. * Step 3. Run mkomega to generate the matrix "omega"
. *****
. mkomega $w $dvars, sim_measure(L2) out($y)
(0 observations deleted)
. matrix omega=r(M)
```

```

. *****
. * Step 4. Fit the model using ntreatreg (to get ATE with neighborhood
> interactions) (resulting graph shown in figure 2)
. *****
. ntreatreg $y $w $xvars, hetero($hvars) spill(omega) graphic

```

Source	SS	df	MS	Number of obs	=	49
Model	10269.266	7	1467.038	F(7, 41)	=	18.98
Residual	3168.95358	41	77.2915508	Prob > F	=	0.0000
				R-squared	=	0.7642
				Adj R-squared	=	0.7239
Total	13438.2195	48	279.962907	Root MSE	=	8.7916

```


```

crime	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
cp	14.5955	3.75345	3.89	0.000	7.015257 22.17575
inc	-.936559	.3619498	-2.59	0.013	-1.667531 -.2055866
hoval	-.1753827	.0961938	-1.82	0.076	-.3696501 .0188847
_ws_inc	-1.157042	.9291237	-1.25	0.220	-3.033445 .7193614
_ws_hoval	.1890178	.2091914	0.90	0.372	-.2334528 .6114885
_z_inc	-10.99322	7.124302	-1.54	0.131	-25.38104 3.394596
_z_hoval	-7.99784	2.5437	-3.14	0.003	-13.13495 -2.860734
_cons	400.355	111.1496	3.60	0.001	175.8839 624.8262

```

(49 real changes made)
(49 real changes made)
(49 real changes made)
(49 real changes made)
(25 missing values generated)
(24 missing values generated)
. * Store the estimates, put the ATE into a scalar, and rename variables
. estimates store REG_peer
. scalar ate_neigh = _b[$w]
. rename ATE_x _ATE_x_spill
. rename ATET_x _ATET_x_spill
. rename ATENT_x _ATENT_x_spill
. display ate_neigh
14.595504
. *****
. * Step 5. Test if the coefficients of the peer effect are jointly zero
. *****
. * If we accept the null Ho: gamma*beta0=0, the peer effect is negligible
. * If we do not accept the null, the peer effect is at work
. test _z_inc = _z_hoval = 0
(1) _z_inc - _z_hoval = 0
(2) _z_inc = 0
      F( 2, 41) = 5.78
      Prob > F = 0.0061
. * We reject that peer effects are negligible

```

```

. *****
. * Step 6. Fit the model using ivtreatreg (to get ATE without
> neighborhood interactions)
. *****
. ivtreatreg $y $w $xvars, hetero($hvars) model(cf-ols)

```

Source	SS	df	MS	Number of obs	=	49
Model	9375.05895	5	1875.01179	F(5, 43)	=	19.84
Residual	4063.1606	43	94.4921069	Prob > F	=	0.0000
				R-squared	=	0.6976
				Adj R-squared	=	0.6625
Total	13438.2195	48	279.962907	Root MSE	=	9.7207

```


```

crime	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
cp	13.59008	4.119155	3.30	0.002	5.283016 21.89715
inc	-.8335211	.3384488	-2.46	0.018	-1.516068 -.1509741
hoval	-.1885477	.1036879	-1.82	0.076	-.3976543 .0205588
_ws_inc	-1.26008	1.004873	-1.25	0.217	-3.286599 .7664396
_ws_hoval	.2021829	.2300834	0.88	0.384	-.2618246 .6661904
_cons	46.52524	6.948544	6.70	0.000	32.51217 60.53832

```

(49 real changes made)
(49 real changes made)
(25 missing values generated)
(24 missing values generated)

. * Store the estimates and put the ATE into a scalar
. estimates store REG_no_peer
. scalar ate_no_neigh = _b[$w]
. *****
. * Step 7. Calculate the magnitude of the neighborhood-interactions bias
. *****
. * Bias in level
. scalar bias= ate_no_neigh - ate_neigh
. * Bias in percentage
. scalar bias_perc=(bias/ate_no_neigh)*100
. display bias_perc
-7.3981897
. ***** END IMPLEMENTATION *****

```

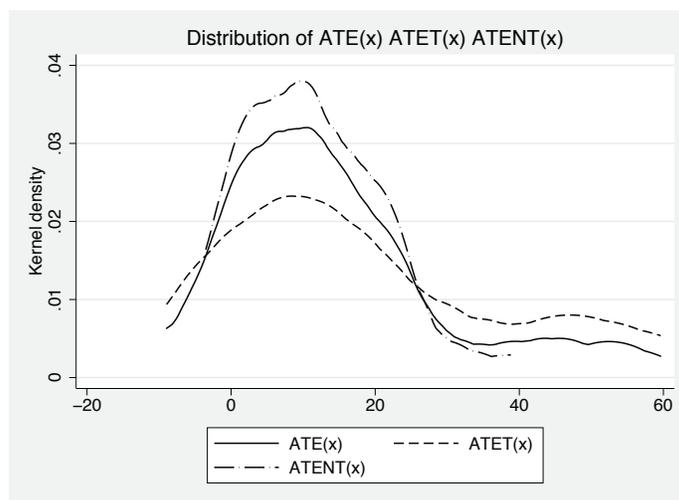


Figure 2. Graphical representation of the model with neighborhood effects

In conclusion, if the analyst does not properly consider neighborhood effects, then the actual effect of housing location on crime will be underestimated, although such underestimation seems not too large in this example. Moreover, the test shows that peer effects are relevant in such a context, because the regression coefficients of the peer component are jointly significant.

7 Conclusion

In this article, I presented a counterfactual model (embedded into the traditional Rubin POM) identifying ATEs by CMI when correlated peer (or neighborhood) effects are considered. I provided the command `ntreatreg` to implement such models in practical applications. Moreover, as a by-product of and designed to be used in combination with `ntreatreg`, I provided the `mkomega` command for generating a similarity matrix based on a set of covariates.

The model and its accompanying command estimate ATEs when the SUTVA is relaxed under specific conditions. I set out two instructional applications: i) a simulation exercise, useful to show both model implementation and `ntreatreg` correctness, and ii) an application to real data, aimed at measuring the effect of housing location on crime. In this second application, results are also compared with a no-interaction setting.

This model has various limitations. In what follows, I suggest some potential developments that other Stata developers can account for. Indeed, the model might be improved by

- allowing for treated units to be affected by other treated units' outcomes and untreated units to be affected by other untreated ones' outcomes;
- extending the model to “multiple” or “continuous” treatment, when treatment may be multivalued or fractional, for instance, by still holding CMI;
- allowing for unit potential outcome to depend on other units' treatment;
- identifying ATEs with neighborhood interactions when the treatment is endogenous (that is, relaxing CMI) by implementing a generalized method of moments instrumental-variables estimation procedure;
- going beyond the potential outcomes' parametric form, thus relying on a nonparametric or semiparametric specification.⁷

Finally, an issue that deserves further inquiry is the assumption of exogeneity of the weighting matrix Ω . Indeed, a challenging question is: what happens if individuals strategically modify their behavior to better take advantage of others' treatment outcome? It is clear that if there exists some correlation between unobservable confounders affecting outcome and neighborhood choice, the weights may become endogenous, thus yielding further identification problems for previous causal effects. Future studies should tackle situations in which this possibility may occur.

8 References

- Angrist, J. D. 2014. The perils of peer effects. *Labour Economics* 30: 98–108.
- Anselin, L. 1988. *Spatial Econometrics: Methods and Models*. Dordrecht: Kluwer.
- Arduini, T., E. Patacchini, and E. Rainone. 2014. Identification and estimation of outcome response with heterogeneous treatment externalities. CPR Working Paper 167. <https://www.maxwell.syr.edu/uploadedFiles/cpr/publications/working-papers2/wp167.pdf>.
- Aronow, P. M., and C. Samii. 2013. Estimating average causal effects under general interference, with application to a social network experiment. ArXiv Working Paper No. arXiv:1305.6156. <https://arxiv.org/abs/1305.6156>.
- Cerulli, G. 2014. `ivtreatreg`: A command for fitting binary treatment models with heterogeneous response to treatment and unobservable selection. *Stata Journal* 14: 453–480.

7. For instance, one possibility would be that of relaxing the linear dependence of y on x , by using a partially linear form for the potential outcomes.

- Cox, D. R. 1958. *Planning of Experiments*. New York: Wiley.
- Holland, P. W. 1986. Statistics and causal inference. *Journal of the American Statistical Association* 81: 945–960.
- Hudgens, M. G., and M. E. Halloran. 2008. Toward causal inference with interference. *Journal of the American Statistical Association* 103: 832–842.
- Imbens, G. W., and J. D. Angrist. 1994. Identification and estimation of local average treatment effects. *Econometrica* 62: 467–475.
- Manski, C. F. 1993. Identification of endogenous social effects: The reflection problem. *Review of Economic Studies* 60: 531–542.
- . 2013. Identification of treatment response with social interactions. *Econometrics Journal* 16: S1–S23.
- Robins, J. M., M. A. Hernán, and B. Brumback. 2000. Marginal structural models and causal inference in epidemiology. *Epidemiology* 11: 550–560.
- Rosenbaum, P. R. 2007. Interference between units in randomized experiments. *Journal of the American Statistical Association* 102: 191–200.
- Rubin, D. B. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology* 66: 688–701.
- . 1977. Assignment to treatment group on the basis of a covariate. *Journal of Educational Statistics* 2: 1–26.
- . 1978. Bayesian inference for causal effects: The role of randomization. *Annals of Statistics* 6: 34–58.
- Sobel, M. E. 2006. What do randomized studies of housing mobility demonstrate?: Causal inference in the face of interference. *Journal of the American Statistical Association* 101: 1398–1407.
- Tchetgen Tchetgen, E. J., and T. J. VanderWeele. 2010. On causal inference in the presence of interference. *Statistical Methods in Medical Research* 21: 55–75.
- Wooldridge, J. M. 1997. On two stage least squares estimation of the average treatment effect in a random coefficient model. *Economics Letters* 56: 129–133.
- . 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

About the author

Giovanni Cerulli is a researcher at CNR-IRCrES, National Research Council of Italy, Institute for Research on Sustainable Economic Growth. He received a degree in statistics and a PhD in economic sciences from Sapienza University of Rome and is editor-in-chief of the *International Journal of Computational Economics and Econometrics*. His main research interest is applied

microeconometrics with a focus on counterfactual treatment-effects models for program evaluation. He is the author of the book *Econometric Evaluation of Socio-Economic Programs: Theory and Applications* (Springer, 2015). He has published articles in high-quality, refereed economics journals.

A Appendix A

This appendix shows how to obtain the formulas of ATE and ATE(\mathbf{x}) set out in (3) and (3), and then shows how regression (12) can be obtained. Finally, the appendix details proof that assumption 1 is sufficient for consistently estimating the parameters of regression (12) by OLS.

A.1 Formula of ATE with neighborhood interactions

Given assumptions 2 and 3, and the implied equations in (7), we get that

$$\begin{aligned}
 y_{1i} &= \mu_1 + \mathbf{x}_i \boldsymbol{\beta}_1 + e_{1i} \\
 y_{0i} &= \mu_0 + \mathbf{x}_i \boldsymbol{\beta}_0 + \gamma s_i + e_{0i} \\
 s_i &= \sum_{j=1}^{N_1} \omega_{ij} y_{1j} \\
 \text{ATE} &= E(y_{1i} - y_{0i}) \\
 &= E \left((\mu_1 + \mathbf{x}_i \boldsymbol{\beta}_1 + e_{1i}) \right. \\
 &\quad \left. - \left[\mu_0 + \mathbf{x}_i \boldsymbol{\beta}_0 + \gamma \left\{ \mu_1 + \left(\sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j \right) \boldsymbol{\beta}_1 + \sum_{j=1}^{N_1} \omega_{ij} e_{1j} \right\} + e_{0i} \right] \right) \\
 &= E \left[\mu_1 + \mathbf{x}_i \boldsymbol{\beta}_1 + e_{1i} \right. \\
 &\quad \left. - \left\{ \mu_0 + \mathbf{x}_i \boldsymbol{\beta}_0 + \gamma \mu_1 + \gamma \left(\sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j \right) \boldsymbol{\beta}_1 + \gamma \sum_{j=1}^{N_1} \omega_{ij} e_{1j} + e_{0i} \right\} \right] \\
 &= E \left\{ \mu_1 + \mathbf{x}_i \boldsymbol{\beta}_1 + e_{1i} - \mu_0 - \mathbf{x}_i \boldsymbol{\beta}_0 - \gamma \mu_1 - \gamma \left(\sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j \right) \boldsymbol{\beta}_1 \right. \\
 &\quad \left. - \gamma \sum_{j=1}^{N_1} \omega_{ij} e_{1j} - e_{0i} \right\}
 \end{aligned}$$

$$\begin{aligned}
&= E \left\{ \mu_1 - \gamma \mu_1 - \mu_0 + \mathbf{x}_i \boldsymbol{\beta}_1 - \mathbf{x}_i \boldsymbol{\beta}_0 - \gamma \left(\sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j \right) \boldsymbol{\beta}_1 \right. \\
&\quad \left. - \gamma \sum_{j=1}^{N_1} \omega_{ij} e_{1j} + e_{1i} - e_{0i} \right\} \\
&= E \left\{ \mu_1 (1 - \gamma) - \mu_0 + \mathbf{x}_i (\boldsymbol{\beta}_1 - \boldsymbol{\beta}_0) - \left(\sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j \right) \gamma \boldsymbol{\beta}_1 \right. \\
&\quad \left. - \gamma \sum_{j=1}^{N_1} \omega_{ij} e_{1j} + e_{1i} - e_{0i} \right\} \\
&= E \left\{ \mu_1 (1 - \gamma) - \mu_0 + \mathbf{x}_i \boldsymbol{\delta} - \left(\sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j \right) \gamma \boldsymbol{\beta}_1 \right. \\
&\quad \left. - \gamma \sum_{j=1}^{N_1} \omega_{ij} e_{1j} + e_{1i} - e_{0i} \right\} \\
&= \mu + E \left\{ \mathbf{x}_i \boldsymbol{\delta} - \left(\sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j \right) \gamma \boldsymbol{\beta}_1 - e_i \right\} \\
&= \mu + E(\mathbf{x}_i) \boldsymbol{\delta} - \gamma E \left(\underbrace{\sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j}_{\mathbf{v}_i} \right) \boldsymbol{\beta}_1
\end{aligned}$$

This implies that $\text{ATE} = E(y_{1i} - y_{0i}) = \mu + E(\mathbf{x}_i) \boldsymbol{\delta} - \gamma E(\mathbf{v}_i) \boldsymbol{\beta}_1$, which has a sample equivalent of

$$\begin{aligned}
\widehat{\text{ATE}} &= \widehat{\mu} \frac{1}{N} \left(\sum_{i=1}^N \mathbf{x}_i \right) \widehat{\boldsymbol{\delta}} - \widehat{\gamma} \frac{1}{N} \left(\sum_{i=1}^N \mathbf{v}_i \right) \widehat{\boldsymbol{\beta}}_1 \\
&= \mu + \frac{1}{N} \left(\sum_{i=1}^N \mathbf{x}_i \right) \widehat{\boldsymbol{\delta}} - \widehat{\gamma} \frac{1}{N} \left\{ \sum_{i=1}^N \left(\sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j \right) \right\} \widehat{\boldsymbol{\beta}}_1
\end{aligned}$$

where $\mu = \mu_1(1 - \gamma) - \mu_0$ and $\boldsymbol{\delta} = \boldsymbol{\beta}_1 - \boldsymbol{\beta}_0$.

As an example, consider the case in which $N = 4$ and $N_1 = N_0 = 2$. Suppose that the matrix Ω is organized as follows:

$$\begin{array}{cc} & \begin{array}{cc} \text{T} & \text{C} \end{array} \\ \begin{array}{c} \text{T} \\ \text{C} \end{array} & \begin{bmatrix} \omega_{11} & \omega_{12} & \omega_{13} & \omega_{14} \\ \omega_{21} & \omega_{22} & \omega_{23} & \omega_{24} \\ \omega_{31} & \omega_{32} & \omega_{33} & \omega_{34} \\ \omega_{41} & \omega_{42} & \omega_{43} & \omega_{44} \end{bmatrix} \end{array}$$

Suppose we have just one confounder, x . In this case, we have

$$\begin{aligned} \widehat{\text{ATE}} &= \widehat{\mu} + \frac{1}{4} \left(\sum_{i=1}^4 x_i \right) \widehat{\delta} - \widehat{\gamma} \times \widehat{\beta}_1 \frac{1}{4} \left\{ \sum_{i=1}^4 \left(\sum_{j=1}^2 \omega_{ij} x_j \right) \right\} \\ &= \widehat{\mu} + \frac{1}{4} \left(\sum_{i=1}^4 x_i \right) \widehat{\delta} - \widehat{\gamma} \times \widehat{\beta}_1 \frac{1}{4} \left\{ \sum_{i=1}^4 (\omega_{i1} x_1 + \omega_{i2} x_2) \right\} \\ &= \widehat{\mu} + \overline{x} \widehat{\delta} - \widehat{\gamma} \times \widehat{\beta}_1 \frac{1}{4} \left\{ \underbrace{\sum_{i=1}^4 (\omega_{i1} x_1 + \omega_{i2} x_2)}_{v_1} \right\} = \widehat{\mu} + \overline{x} \widehat{\delta} - \widehat{\gamma} \times \widehat{\beta}_1 \overline{v} \end{aligned}$$

Observe that

$$\begin{aligned} v_1 &= \omega_{11} x_1 + \omega_{12} x_2 \\ v_2 &= \omega_{21} x_1 + \omega_{22} x_2 \\ v_3 &= \omega_{31} x_1 + \omega_{32} x_2 \\ v_4 &= \omega_{41} x_1 + \omega_{42} x_2 \end{aligned}$$

implying

$$\widehat{\text{ATE}} = \widehat{\mu} + \overline{x} \widehat{\delta} - \widehat{\gamma} \times \widehat{\beta}_1 \frac{1}{4} \left\{ \sum_{i=1}^4 (\omega_{i1} x_1 + \omega_{i2} x_2) \right\} = \widehat{\mu} + \overline{x} \widehat{\delta} - \widehat{\gamma} \times \widehat{\beta}_1 (\overline{w}_{.1} x_1 + \overline{w}_{.2} x_2)$$

where

$$\overline{w}_{.1} = \frac{1}{4} \sum_{i=1}^4 \omega_{i1} \quad \text{and} \quad \overline{w}_{.2} = \frac{1}{4} \sum_{i=1}^4 \omega_{i2}$$

This means that, by assuming that the externality effect comes only from treated to untreated units, thus excluding other types of feedbacks, it is equivalent to consider only the first two columns of Ω in the calculation of the externality component, those refereeing to the treated units. That is,

$$\begin{array}{cc}
 & \begin{array}{cc} \text{T} & \text{C} \end{array} \\
 \begin{array}{c} \text{T} \\ \text{C} \end{array} & \begin{bmatrix} \omega_{11} & \omega_{12} & \omega_{13} & \omega_{14} \\ \omega_{21} & \omega_{22} & \omega_{23} & \omega_{24} \\ \omega_{31} & \omega_{32} & \omega_{33} & \omega_{34} \\ \omega_{41} & \omega_{42} & \omega_{43} & \omega_{44} \end{bmatrix}
 \end{array}$$

where neither of the two columns refers to the control group.

A.2 Formula of ATE(\mathbf{x}_i) with neighborhood interactions

Given assumptions 2 and 3, and the result in A1, we get

$$\begin{aligned}
 \text{ATE}(\mathbf{x}_i) &= E(y_{1i} - y_{0i} | \mathbf{x}_i) = \mu + E \left\{ \mathbf{x}_i \boldsymbol{\delta} - \left(\sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j \right) \gamma \boldsymbol{\beta}_1 - e_i | \mathbf{x}_i \right\} \\
 &= \mu + \mathbf{x}_i \boldsymbol{\delta} - \left(\sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j \right) \gamma \boldsymbol{\beta}_1 + (\bar{\mathbf{x}} \boldsymbol{\delta} - \bar{\mathbf{x}} \boldsymbol{\delta}) \\
 &\quad + \left\{ E \left(\sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j \right) \gamma \boldsymbol{\beta}_1 - E \left(\sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j \right) \gamma \boldsymbol{\beta}_1 \right\} \\
 &= \left\{ \mu + \bar{\mathbf{x}} \boldsymbol{\delta} - E \left(\sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j \right) \gamma \boldsymbol{\beta}_1 \right\} + (\mathbf{x}_i - \bar{\mathbf{x}}) \boldsymbol{\delta} \\
 &\quad + \left\{ E \left(\sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j \right) - \left(\sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j \right) \right\} \gamma \boldsymbol{\beta}_1 \\
 &= \text{ATE} + (\mathbf{x}_i - \bar{\mathbf{x}}) \boldsymbol{\delta} + (\bar{\mathbf{v}} - \mathbf{v}_i) \boldsymbol{\lambda}
 \end{aligned}$$

where $\boldsymbol{\lambda} = \gamma \boldsymbol{\beta}_1$.

A.3 Obtaining regression (12)

By substitution of the potential outcome, as in (7), into the POM, we get that

$$\begin{aligned}
y_i &= \left(\mu_0 + \mathbf{x}_i \boldsymbol{\beta}_0 + \gamma \sum_{j=1}^{N_1} \omega_{ij} y_{1j} + e_{0i} \right) \\
&\quad + w_i \left\{ (\mu_1 + \mathbf{x}_i \boldsymbol{\beta}_1 + e_{1i}) - \left(\mu_0 + \mathbf{x}_i \boldsymbol{\beta}_0 + \gamma \sum_{j=1}^{N_1} \omega_{ij} y_{1j} + e_{0i} \right) \right\} \\
&= \left(\mu_0 + \mathbf{x}_i \boldsymbol{\beta}_0 + \gamma \sum_{j=1}^{N_1} \omega_{ij} y_{1j} + e_{0i} \right) + w_i (\mu_1 - \mu_0) + w_i \mathbf{x}_i (\boldsymbol{\beta}_1 - \boldsymbol{\beta}_0) \\
&\quad + w_i (e_{1i} - e_{0i}) - w_i \gamma \sum_{j=1}^{N_1} \omega_{ij} y_{1j} \\
&= \mu_0 + \mathbf{x}_i \boldsymbol{\beta}_0 + \gamma \mu_1 + \left(\gamma \sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j \right) \boldsymbol{\beta}_1 + \gamma \sum_{j=1}^{N_1} \omega_{ij} e_{1j} + e_{0i} + w_i (\mu_1 - \mu_0) \\
&\quad + w_i \mathbf{x}_i (\boldsymbol{\beta}_1 - \boldsymbol{\beta}_0) + w_i (e_{1i} - e_{0i}) - w_i \gamma \mu_1 - w_i \gamma \sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j \boldsymbol{\beta}_1 - w_i \gamma \sum_{j=1}^{N_1} \omega_{ij} e_{1j} \\
&= \mu_0 + \mathbf{x}_i \boldsymbol{\beta}_0 + \gamma \mu_1 + \left(\gamma \sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j \right) \boldsymbol{\beta}_1 \\
&\quad + \underbrace{\left\{ \gamma \sum_{j=1}^{N_1} \omega_{ij} e_{1j} + e_{0i} + w_i (e_{1i} - e_{0i}) - w_i \gamma \sum_{j=1}^{N_1} \omega_{ij} e_{1j} \right\}}_{e_i} + w_i (\mu_1 - \mu_0) \\
&\quad + w_i \mathbf{x}_i (\boldsymbol{\beta}_1 - \boldsymbol{\beta}_0) - w_i \gamma \mu_1 - w_i \gamma \left(\sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j \right) \boldsymbol{\beta}_1 \\
&= \mu_0 + \mathbf{x}_i \boldsymbol{\beta}_0 + \gamma \mu_1 + \left(\gamma \sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j \right) \boldsymbol{\beta}_1 + w_i (\mu_1 - \mu_0) + w_i \mathbf{x}_i (\boldsymbol{\beta}_1 - \boldsymbol{\beta}_0) - w_i \gamma \mu_1 \\
&\quad - w_i \gamma \left(\sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j \right) \boldsymbol{\beta}_1 + e_i
\end{aligned}$$

$$\begin{aligned}
&= (\mu_0 + \gamma\mu_1) + \left(\gamma \sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j \right) \boldsymbol{\beta}_1 + w_i (\mu_1 - \mu_0 - \gamma\mu_1) + \mathbf{x}_i \boldsymbol{\beta}_0 + w_i \mathbf{x}_i (\boldsymbol{\beta}_1 - \boldsymbol{\beta}_0) \\
&\quad - w_i \gamma \left(\sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j \right) \boldsymbol{\beta}_1 + e_i \\
&= (\mu_0 + \gamma\mu_1) + \left(\gamma \sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j \right) \boldsymbol{\beta}_1 + w_i (\mu_1 - \mu_0 - \gamma\mu_1) + \mathbf{x}_i \boldsymbol{\beta}_0 + w_i \mathbf{x}_i \boldsymbol{\delta} \\
&\quad - w_i \left(\sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j \right) \gamma \boldsymbol{\beta}_1 + e_i \\
&= (\mu_0 + \gamma\mu_1) + \left(\sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j \right) \gamma \boldsymbol{\beta}_1 + w_i (\mu_1 - \mu_0 - \gamma\mu_1) + \mathbf{x}_i \boldsymbol{\beta}_0 + w_i \mathbf{x}_i \boldsymbol{\delta} \\
&\quad - w_i \left(\sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j \right) \gamma \boldsymbol{\beta}_1 + e_i + (w_i \bar{\mathbf{x}} \boldsymbol{\delta} - w_i \bar{\mathbf{x}} \boldsymbol{\delta}) \\
&\quad + \left\{ w_i E \left(\underbrace{\sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j}_{\mathbf{v}_i} \right) \gamma \boldsymbol{\beta}_1 - w_i E \left(\underbrace{\sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j}_{\mathbf{v}_i} \right) \gamma \boldsymbol{\beta}_1 \right\} \\
&= (\mu_0 + \gamma\mu_1) + w_i (\mu + \bar{\mathbf{x}} \boldsymbol{\delta} - \bar{\mathbf{v}} \boldsymbol{\lambda}) + \mathbf{x}_i \boldsymbol{\beta}_0 + w_i (\mathbf{x}_i - \bar{\mathbf{x}}) \boldsymbol{\delta} + \mathbf{v}_i \boldsymbol{\lambda} + w_i \bar{\mathbf{v}} \boldsymbol{\lambda} - w_i \mathbf{v}_i \boldsymbol{\lambda} + e_i \\
&= \eta + w_i \times \text{ATE} + \mathbf{x}_i \boldsymbol{\beta}_0 + w_i (\mathbf{x}_i - \bar{\mathbf{x}}) \boldsymbol{\delta} + \{ \mathbf{v}_i + w_i (\bar{\mathbf{v}} - \mathbf{v}_i) \} \boldsymbol{\lambda} + e_i
\end{aligned}$$

Therefore, we can conclude that

$$y_i = \eta + w_i \times \text{ATE} + \mathbf{x}_i \boldsymbol{\beta}_0 + w_i (\mathbf{x}_i - \bar{\mathbf{x}}) \boldsymbol{\delta} + \mathbf{z}_i \boldsymbol{\lambda} + e_i$$

where $\mathbf{z}_i = \mathbf{v}_i + w_i (\bar{\mathbf{v}} - \mathbf{v}_i)$, $\mathbf{v}_i = \sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j$, $\bar{\mathbf{v}} = 1/N \sum_{i=1}^N (\sum_{j=1}^{N_1} \omega_{ij} \mathbf{x}_j)$, $\boldsymbol{\lambda} = \gamma \boldsymbol{\beta}_1$, $\eta = \mu_0 + \gamma\mu_1$, and $\boldsymbol{\delta} = \boldsymbol{\beta}_1 - \boldsymbol{\beta}_0$.

A.4 OLS consistency

Under assumption 1 (CMI), the parameters of regression (12) can be consistently estimated by OLS. Indeed, we immediately see that the mean of e_i conditional on $(w_i; \mathbf{x}_i)$ is equal to 0:

$$\begin{aligned}
 & E \left\{ \gamma \sum_{j=1}^{N_1} \omega_{ij} e_{1j} + e_{0i} + w_i(e_{1i} - e_{0i}) - w_i \gamma \sum_{j=1}^{N_1} \omega_{ij} e_{1j} \middle| w_i, \mathbf{x}_i \right\} \\
 &= E \left\{ \gamma \sum_{j=1}^{N_1} \omega_{ij} e_{1j} \middle| w_i, \mathbf{x}_i \right\} + E \{ e_{0i} | w_i, \mathbf{x}_i \} + E \{ w_i(e_{1i} - e_{0i}) | w_i, \mathbf{x}_i \} \\
 &\quad - E \left(w_i \gamma \sum_{j=1}^{N_1} \omega_{ij} e_{1j} \middle| w_i, \mathbf{x}_i \right) \\
 &= \gamma \sum_{j=1}^{N_1} \omega_{ij} E(e_{1j} | \mathbf{x}_i) + E(e_{0i} | \mathbf{x}_i) + w_i E \{ (e_{1i} - e_{0i}) | \mathbf{x}_i \} - w_i \gamma \sum_{j=1}^{N_1} \omega_{ij} E(e_{1j} | \mathbf{x}_i) \\
 &= 0
 \end{aligned}$$

where $\eta = \mu_0 + \gamma\mu_1$.

The `synth_runner` package: Utilities to automate synthetic control estimation using `synth`

Sebastian Galiani
University of Maryland
College Park, MD
galiani@econ.umd.edu

Brian Quistorff
Microsoft AI and Research
Redmond, WA
Brian.Quistorff@microsoft.com

Abstract. The synthetic control methodology (Abadie and Gardeazabal, 2003, *American Economic Review* 93: 113–132; Abadie, Diamond, and Hainmueller, 2010, *Journal of the American Statistical Association* 105: 493–505) allows for a data-driven approach to small-sample comparative studies. `synth_runner` automates the process of running multiple synthetic control estimations using `synth`. It conducts placebo estimates in space (estimations for the same treatment period but on all the control units). Inference (p -values) is provided by comparing the estimated main effect with the distribution of placebo effects. It also allows several units to receive treatment, possibly at different time periods. It allows automatic generation of the outcome predictors and diagnostics by splitting the pretreatment into training and validation portions. Additionally, it provides diagnostics to assess fit and generates visualizations of results.

Keywords: st0500, `synth_runner`, synthetic control methodology, randomization inference

1 Introduction

The synthetic control methodology (SCM) (Abadie and Gardeazabal 2003; Abadie, Diamond, and Hainmueller 2010) is a data-driven approach to small-sample comparative case studies for estimating treatment effects. Similar to a difference-in-differences design, SCM exploits the differences in treated and untreated units across the event of interest. However, in contrast to a difference-in-differences design, SCM does not give all untreated units the same weight in the comparison. Instead, it generates a weighted average of the untreated units that closely matches the treated unit over the pretreatment period. Outcomes for this synthetic control are then projected into the posttreatment period using the weights identified from the pretreatment comparison. This projection is used as the counterfactual for the treated unit. Inference is conducted using placebo tests.

Along with their article, Abadie, Diamond, and Hainmueller (2010) released the `synth` Stata command for single estimations. The `synth_runner` package builds on `synth` to help conduct multiple estimations, inference, and diagnostics as well as to help generate visualizations of results. `synth_runner` is designed to accompany `synth` but not supersede it. For more details about single estimations (variable weights, ob-

ervation weights, covariate balance, and synthetic control outcomes when there are multiple time periods), use `synth` directly.

2 SCM

Abadie, Diamond, and Hainmueller (2010) posit the following data-generating process. Let D_{jt} be an indicator for treatment for unit j at time t . Next, let the observed outcome variable Y_{jt} be the sum of a time-varying treatment effect, $\alpha_{jt}D_{jt}$, and the no-treatment counterfactual Y_{jt}^N , which is specified using a factor model

$$\begin{aligned} Y_{jt} &= \alpha_{jt}D_{jt} + Y_{jt}^N \\ &= \alpha_{jt}D_{jt} + (\delta_t + \boldsymbol{\theta}_t\mathbf{Z}_j + \boldsymbol{\lambda}_t\boldsymbol{\mu}_j + \varepsilon_{jt}) \end{aligned} \quad (1)$$

where δ_t is an unknown time factor, \mathbf{Z}_j is an $(r \times 1)$ vector of observed covariates unaffected by treatment, $\boldsymbol{\theta}_t$ is a $(1 \times r)$ vector of unknown parameters, $\boldsymbol{\lambda}_t$ is a $(1 \times F)$ vector of unknown factors, $\boldsymbol{\mu}_j$ is an $(F \times 1)$ vector of unknown factor loadings, and the error ε_{jt} is independent across units and time with zero mean. Letting the first unit be the treated unit, we estimate the treatment effect by approximating the unknown Y_{1t}^N with a weighted average of untreated units

$$\hat{\alpha}_{1t} = Y_{1t} - \sum_{j \geq 2} w_j Y_{jt}$$

Equation (1) simplifies to the traditional fixed-effects equation if $\boldsymbol{\lambda}_t\boldsymbol{\mu}_j = \phi_j$. The fixed-effects model allows for unobserved heterogeneity that is only time invariant. The factor model employed by SCM generalizes this to allow for the existence of nonparallel trends between treated and untreated units after controlling for observables.

2.1 Estimation

To begin with, let there be a single unit that receives treatment. Let T_0 be the number of pretreatment periods of the T total periods. Index units $\{1, \dots, J+1\}$ such that the first unit is the treated unit and the others are “donors”. Let \mathbf{Y}_j be $(T \times 1)$ the vector of outcomes for unit j and \mathbf{Y}_0 be the $(T \times J)$ matrix of outcomes for all donors. Let \mathbf{W} be a $(J \times 1)$ observation-weight matrix $(w_2, w_3, \dots, w_{J+1})'$, where $\sum_{j=2}^{J+1} w_j = 1$ and $w_j \geq 0 \forall j \in \{2, \dots, J+1\}$. A weighted average of donors over the outcome is constructed as $\mathbf{Y}_0\mathbf{W}$. Partition the outcome into pretreatment and posttreatment vectors $\mathbf{Y}_j = (\overleftarrow{\mathbf{Y}}_j \setminus \overrightarrow{\mathbf{Y}}_j)$. Let \mathbf{X} represent a set of k pretreatment characteristics (“predictors”). This includes \mathbf{Z} (the observed covariates above) and M linear combinations of $\overleftarrow{\mathbf{Y}}$ so that $k = r + M$. Analogously, let \mathbf{X}_0 be the $(k \times J)$ matrix of donor predictors. Let \mathbf{V} be a $(k \times k)$ variable-weight matrix indicating the relative significance of the predictor variables.

Given \mathbf{Y} and \mathbf{X} , estimation of SCM consists of finding the optimal weighting matrices \mathbf{W} and \mathbf{V} . For a given \mathbf{V} , \mathbf{W} is picked to minimize the root mean squared

prediction error (RMSPE) of the predictor variables, $\|\mathbf{X}_1 - \mathbf{X}_0 \mathbf{W}\|_{\mathbf{V}}$. In this way, the treated unit and its synthetic control look similar along dimensions that matter for predicting pretreatment outcomes. The inferential procedure is valid for any \mathbf{V} , but [Abadie, Diamond, and Hainmueller \(2010\)](#) suggest that \mathbf{V} be picked to minimize the prediction error of the pretreatment outcome between the treated unit and the synthetic control. Define distance measures $\|\mathbf{A}\|_{\mathbf{B}} = \sqrt{\mathbf{A}'\mathbf{B}\mathbf{A}}$ and $\|\mathbf{A}\| = \sqrt{\mathbf{A}'\text{cols}(\mathbf{A})^{-1}\mathbf{A}}$. $\|\overleftarrow{\mathbf{Y}}_1 - \overleftarrow{\mathbf{Y}}_0 \mathbf{W}\|$ is then the pretreatment RMSPE with a given weighted average of the control units. Define this pretreatment RMSPE as \overleftarrow{s}_1 , and define the posttreatment RMSPE as \overrightarrow{s}_1 . \mathbf{V} is then picked to minimize \overleftarrow{s}_1 (note that \mathbf{W} is a function of \mathbf{V}).

If weights can be found such that the synthetic control matches the treated unit in the pretreatment period

$$\|\overleftarrow{\mathbf{Y}}_1 - \overleftarrow{\mathbf{Y}}_0 \mathbf{W}\| = 0 = \|\mathbf{Z}_1 - \mathbf{Z}_0 \mathbf{W}\|$$

and $\sum_{t=1}^{T_0} \boldsymbol{\lambda}'_t \boldsymbol{\lambda}_t$ is nonsingular, then $\hat{\boldsymbol{\alpha}}_1$ will have a bias that goes to zero as the number of preintervention periods grows large relative to the scale of the ε_{jt} .

2.2 Inference

After estimating the effect, determine statistical significance by running placebo tests. Estimate the same model on each untreated unit, assuming it was treated at the same time, to get a distribution of “in-place” placebo effects. Disallow the actual treated unit from being considered for the synthetic controls of these other units. If the distribution of placebo effects yields many effects as large as the main estimate, then it is likely that the estimated effect was observed by chance. This nonparametric, exact test has the advantage of not imposing any distribution on the errors.

Suppose that the estimated effect for a particular posttreatment period is $\hat{\alpha}_{1t}$ and that the distribution of corresponding in-place placebos is $\hat{\alpha}_{1t}^{PL} = \{\hat{\alpha}_{jt} : j \neq 1\}$. The two-sided p -value is then

$$\begin{aligned} p\text{-value} &= \Pr(|\hat{\alpha}_{1t}^{PL}| \geq |\hat{\alpha}_{1t}|) \\ &= \frac{\sum_{j \neq 1} 1(|\hat{\alpha}_{jt}| \geq |\hat{\alpha}_{1t}|)}{J} \end{aligned}$$

and the one-sided p -values (for positive effects) are

$$p\text{-value} = \Pr(\hat{\alpha}_{1t}^{PL} \geq \hat{\alpha}_{1t})$$

When treatment is randomized, this becomes classical randomization inference.¹ If treatment is not randomly assigned, the p -value still has the interpretation of being the

1. One may want to include $\hat{\alpha}_{1t}$ in the comparison distribution as is common in randomization inference. This adds a 1 to the numerator and denominator of the p -value fraction. [Abadie, Diamond, and Hainmueller \(2015\)](#) and [Cavallo et al. \(2013\)](#), however, do not take this approach. With multiple treatments, there would be several approaches to adding the effects on the treated to the comparison distribution, so they are not dealt with here.

proportion of control units that have an estimated effect at least as large as that of the treated unit. Confidence intervals can be constructed by inverting the p -values for $\hat{\alpha}_{1t}$. However, one should take care with these. As noted by Abadie, Diamond, and Hainmueller (2015), they do not have the standard interpretation when treatment is not considered randomly assigned.

To gauge the joint effect across all posttreatment periods, Abadie, Diamond, and Hainmueller (2010) suggest using posttreatment RMSPE \vec{s}_1 . In this case, \vec{s}_1 would be compared with the corresponding \vec{s}_1^{PL} .

The placebo effects may be quite large if those units were not matched well in the pretreatment period. This would cause p -values to be too conservative. To control for this, one may want to adjust $\hat{\alpha}_{jt}$ and \vec{s}_j for the quality of the pretreatment matches. Adjustment can be achieved by two mechanisms:

- Restricting the comparison set of control effects to include only those that match well. This is done by setting a multiple m and removing all placebos j with $\overleftarrow{s}_j > m \overleftarrow{s}_1$.
- Dividing all effects by the corresponding pretreatment match quality \overleftarrow{s} to get standardized (studentized) measures: $\hat{\alpha}_{jt}/\overleftarrow{s}_j$ and $\vec{s}_j/\overleftarrow{s}_j$.

Inference can then be conducted over four quantities ($\hat{\alpha}_{jt}$, \vec{s}_j , $\hat{\alpha}_{jt}/\overleftarrow{s}_j$, $\vec{s}_j/\overleftarrow{s}_j$), and the comparison set can also be limited by the choice of m .

2.3 Multiple events

The extension by Cavallo et al. (2013) allows for more than one unit to experience treatment and at possibly different times. Index treated units $g \in \{1, \dots, G\}$. Let J be those units that never undergo treatment. For a particular treatment g , one can estimate an effect, say, the first posttreatment period effect $\hat{\alpha}_g$ (one could use any of the four types discussed above). We omit the t subscript because treatment dates may differ across events. Over all the treatments, the average effect is $\bar{\alpha} = G^{-1} \sum_{g=1}^G \hat{\alpha}_g$.

For each treatment g , one generates a corresponding set of placebo effects $\hat{\alpha}_g^{PL}$, where each untreated unit is thought of as entering treatment at the same time as unit g . If two treated units have the same treatment period, then their placebo sets will be the same.

Averaging over the treatments to obtain $\bar{\alpha}$ smooths out noise in the estimate. The same should be done in constructing $\bar{\alpha}^{PL}$, the set of placebos with which the average treatment estimate is compared for inference. It should be constructed from all possible averages where a single placebo is taken from each $\hat{\alpha}_g^{PL}$. There are $N_{PL}^m = \prod_{g=1}^G J_g$ such possible averages.² Let i index be a selection where a single placebo effect is chosen from

2. The pool may be restricted by match quality. If J_g^m is the number of controls that match as well as treated unit g for the same time period, then $N_{PL}^m = \prod_{g=1}^G J_g^m$.

each treatment placebo set. Let $\bar{\alpha}^{PL(i)}$ represent the average of that placebo selection. Inference is now

$$\begin{aligned} p\text{-value} &= \Pr(|\bar{\alpha}^{PL}| \geq |\bar{\alpha}|) \\ &= \frac{\sum_{i=1}^{N_{PL}} 1(|\bar{\alpha}^{PL(i)}| \geq |\bar{\alpha}|)}{N_{PL}} \end{aligned}$$

2.4 Diagnostics

Cavallo et al. (2013) perform two basic checks to see whether the synthetic control serves as a valid counterfactual. The first is to check whether a weighted average of donors can approximate the treated unit in the pretreatment. This should be satisfied if the treated unit lies within the convex hull of the control units. One can visually compare the difference in pretreatment outcomes between a unit and its synthetic control. Additionally, one could look at the distribution of pretreatment RMSPEs and see what proportion of control units have values at least as high as that of the treated unit. Cavallo et al. (2013) discard several events from their study because they cannot be matched appropriately.

Secondly, one can exclude some pretreatment outcomes from the list of predictors and see whether the synthetic control matches well the treated unit in these periods.³ Because this is still pretreatment, the synthetic control should match well. The initial section of the pretreatment period is often designated the “training” period, with the latter part being the “validation” period. Cavallo et al. (2013) set aside the first half of the pretreatment period as the training period.

3 The *synth_runner* package

The *synth_runner* package contains several tools to help conduct the SCM estimation. It requires the *synth* package, which can be obtained from the Statistical Software Components archive. The main program is *synth_runner*, which is outlined here. There are also simple graphing utilities (*effect_graphs*, *pval_graphs*, and *single_treatment_graphs*) that show basic graphs. These are explained in the following code examples and can be modified easily.

3. Note also that unless some pretreatment outcome variables are dropped from the set of predictors, all other covariate predictors are rendered redundant. The optimization of V will put no weight on those additional predictors in terms of predicting pretreatment outcomes.

3.1 Syntax

```
synth_runner deprvar predictorvars, {trunit(#) trperiod(#)|d(varname)}
  [trends pre_limit_mult(real) training_propr(real) gen_vars
  noenforce_const_pre_length ci max_lead(#) n_pl_avgs(string)
  pred_prog(string) deterministicoutput parallel pvals1s
  drop_units_prog(string) xperiod_prog(string) mspeperiod_prog(string)
  synthsettings ]
```

Postestimation graphing commands are shown in the examples below. The syntax is similar to the `synth` command. New options include `d()`, `trends`, `pre_limit_mult()`, `training_propr()`, `ci`, `pvals1s`, `max_lead()`, `n_pl_avgs()`, `parallel`, `deterministicoutput`, `pred_prog()`, `drop_units_prog()`, `xperiod_prog()`, and `mspeperiod_prog()`. Options not explicitly matched will be passed to `synth` as *synthsettings*.

Required settings:

- *deprvar* specifies the outcome variable.
- *predictorvars* specifies the list of predictor variables. See `help synth` help for more details.

3.2 Options

There are two methods for specifying the unit and time period of treatment: either `trunit()` and `trperiod()` or `d()`. Exactly one of these is required.

`trunit(#)` and `trperiod(#)`, used by `synth`, can be used when there is a single unit entering treatment. Because synthetic control methods split time into pretreatment and treated periods, `trperiod()` is the first of the treated periods and, slightly confusingly, also called posttreatment.

`d(varname)` specifies a binary variable, which is 1 for treated units in treated periods and 0 everywhere else. This allows for multiple units to undergo treatment, possibly at different times.

`trends` will force `synth` to match on the trends in the outcome variable. It does this by scaling each unit's outcome variable so that it is 1 in the last pretreatment period.

`pre_limit_mult(real)` will not include placebo effects in the pool for inference if the match quality of that control, namely, the pretreatment RMSPE, is greater than `pre_limit_mult()` times the match quality of the treated unit. *real* must be greater than or equal to 1.

`training_propr`(*real*) instructs `synth_runner` to automatically generate the outcome predictors. The default is `training_propr(0)`, which is to not generate any (the user then includes the desired ones in *predictorvars*). If the value is set to a number greater than 0, then that initial proportion of the pretreatment period is used as a training period, with the rest being the validation period. Outcome predictors for every time in the training period will be added to the `synth` commands. Diagnostics of the fit for the validation period will be outputted. If the value is between 0 and 1, there will be at least one training period and at least one validation period. If it is set to 1, then all the pretreatment period outcome variables will be used as predictors. This will make other covariate predictors redundant. *real* must be greater than or equal to 0 and less than or equal to 1.

`gen_vars` generates variables in the dataset from estimation. This is allowed only if there is a single period in which units enter treatment. These variables are required for the following: `single_treatment_graphs` and `effect_graphs`. If `gen_vars` is specified, it will generate the following variables:

`lead` contains the respective time period relative to treatment. `lead = 1` specifies the first period of treatment. This is to match Cavallo et al. (2013) and is effectively the offset from T_0 .

`depvar_synth` contains the unit's synthetic control outcome for that time period.

`effect` contains the difference between the unit's outcome and its synthetic control for that time period.

`pre_rmspe` contains the pretreatment match quality in terms of RMSPE. It is constant for a unit.

`post_rmspe` contains a measure of the posttreatment effect (jointly over all post-treatment time periods) in terms of RMSPE. It is constant for a unit.

`depvar_scaled` (if the match was done on trends) is the unit's outcome variable normalized so that its last pretreatment period outcome is 1.

`depvar_scaled_synth` (if the match was done on trends) is the unit's synthetic control (scaled) outcome variable.

`effect_scaled` (if the match was done on trends) is the difference between the unit's scaled outcome and its synthetic control's (scaled) outcome for that time period.

`noenforce_const_pre_length` specifies that maximal histories are desired at each estimation stage. When there are multiple periods, estimations at later treatment dates will have more pretreatment history available. By default, these histories are trimmed on the early side so that all estimations have the same amount of history.

`ci` outputs confidence intervals from randomization inference for raw effect estimates. These should be used only if the treatment is randomly assigned (conditional on covariates and interactive fixed effects). If treatment is not randomly assigned, then these confidence intervals do not have the standard interpretation (see above).

`max_lead(int)` will limit the number of posttreatment periods analyzed. The default is the maximum number of leads that is available for all treatment periods.

`n_pl_avgs(string)` controls the number of placebo averages to compute for inference. The possible total grows exponentially with the number of treated events. The default behavior is to cap the number of averages computed at 1,000,000 and, if the total is more than that, to sample (with replacement) the full distribution. The option `n_pl_avgs(all)` can be used to override this behavior and compute all the possible averages. The option `n_pl_avgs(#)` can be used to specify a specific number less than the total number of averages possible.

`pred_prog(string)` allows for time-contingent predictor sets. The user writes a program that takes as input a time period and outputs via `r(predictors)` a `synth`-style predictor string. If one is not using `training_propr()`, then `pred_prog()` could be used to dynamically include outcome predictors. See example 3 for usage details.

`deterministicoutput`, when used with `parallel`, will eliminate displayed output that would vary depending on the machine (for example, timers and number of parallel clusters) so that log files can be easily compared across runs.

`parallel` will enable parallel processing if the `parallel` command is installed and configured. Version 1.18.2 is needed at a minimum.⁴

`pvals1s` outputs one-sided p -values in addition to the two-sided p -values.

`drop_units_prog(string)` specifies the name of a program that, when passed the unit to be considered treated, will drop other units that should not be considered when forming the synthetic control. This is usually because they are neighboring or interfering units. See example 3 for usage details.

`xperiod_prog(string)` allows for setting `synth`'s `xperiod()` option, which varies with the treatment period. The user-written program is passed the treatment period and should return, via `r(xperiod)`, a `numlist` suitable for `synth`'s `xperiod()` (the period over which generic predictor variables are averaged). See `synth` for more details on the `xperiod()` option. See example 3 for usage details.

`mspeperiod_prog(string)` allows for setting `synth`'s `mspeperiod()` option, which varies with the treatment period. The user-written program is passed the treatment period and should return, via `r(mspeperiod)`, a `numlist` suitable for `synth`'s `mspeperiod()` (the period over which the prediction outcome is evaluated). See `synth` for more details on the `mspeperiod()` option. See example 3 for usage details.

`synthsettings` specifies pass-through options sent to `synth`. See `help synth` for more information. The following are disallowed: `cunit()`, `figure`, `resultspanel()`.

4. At the time of writing, Statistical Software Components does not contain a new enough version. Newer versions are available via the development website <https://github.com/gvegayon/parallel/>.

3.3 Stored results

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

Scalars

<code>e(n_pl)</code>	number of placebo averages used for comparison
<code>e(pval_joint_post)</code>	proportion of placebos that have a posttreatment RMSPE at least as large as the average for the treated units
<code>e(pval_joint_post_t)</code>	proportion of placebos that have a ratio of posttreatment RMSPE over pretreatment RMSPE at least as large as the average ratio for the treated units
<code>e(avg_pre_rmspe_p)</code>	proportion of placebos that have a pretreatment RMSPE at least as large as the average of the treated units; the farther this measure is from 0 toward 1, the better the relative fit of the treated units
<code>e(avg_val_rmspe_p)</code>	when one specifies <code>training_propr()</code> , this is the proportion of placebos that have an RMSPE for the validation period at least as large as the average of the treated units; the farther this measure is from 0 toward 1, the better the relative fit of the treated units

Matrices

<code>e(treat_control)</code>	average treatment outcome (centered around treatment) and the average of the outcome of those units' synthetic controls for the pretreatment and posttreatment periods
<code>e(b)</code>	a vector with the per-period effects (unit's actual outcome minus the outcome of its synthetic control) for posttreatment periods
<code>e(pvals)</code>	a vector of the proportions of placebo effects that are at least as large as the main effect for each posttreatment period
<code>e(pvals_std)</code>	a vector of the proportions of placebo standardized effects that are at least as large as the main standardized effect for each posttreatment period
<code>e(failed_opt_targets)</code>	errors when constructing the synthetic controls for nontreated units are handled gracefully; if any are detected, they will be listed in this matrix (errors when constructing the synthetic control for treated units will abort the method)

3.4 Example usage

The following examples use a dataset from the `synth` package. Ensure that `synth` was installed with ancillary files (for example, `ssc install synth, all`). This panel dataset contains information for 39 U.S. States for the years 1970–2000 (see Abadie, Diamond, and Hainmueller [2010] for details).

```
. sysuse smoking
(Tobacco Sales in 39 US States)

. tsset state year
      panel variable:  state (strongly balanced)
      time variable:  year, 1970 to 2000
                delta: 1 unit
```

► **Example 1**

Reconstruct example 1 from the `synth` help file (note this is not the exact estimation strategy used in [Abadie, Diamond, and Hainmueller \[2010\]](#)):

```
. synth_runner cigsale beer(1984(1)1988) lnincome(1972(1)1988)
>   retprice age15to24 cigsale(1988) cigsale(1980) cigsale(1975),
>   trunit(3) trperiod(1989) gen_vars
Estimating the treatment effects
Estimating the possible placebo effects (one set for each of the 1 treatment
> periods)
|                                     | Total: 38
.....| 11.00s elapsed.
Conducting inference: 5 steps, and 38 placebo averages
Step 1... Finished
Step 2... Finished
Step 3... Finished
Step 4... Finished
Step 5... Finished
Post-treatment results: Effects, p-values, standardized p-values
```

	estimates	pvals	pvals_std
c1	-7.887098	.1315789	0
c2	-9.693599	.1842105	0
c3	-13.8027	.2105263	0
c4	-13.344	.1315789	0
c5	-17.0624	.1052632	0
c6	-20.8943	.0789474	0
c7	-19.8568	.1315789	.0263158
c8	-21.0405	.1578947	0
c9	-21.4914	.1052632	.0263158
c10	-19.1642	.1842105	.0263158
c11	-24.554	.1052632	0
c12	-24.2687	.1052632	.0263158

The program notes progress toward estimating prediction errors and conducting inference. Results for posttreatment periods are shown by default. In this example, they are negative and significant by the standardized effect measure indicating that California's Proposition 99 studied by [Abadie, Diamond, and Hainmueller \(2010\)](#) likely had a negative effect on cigarette sales.

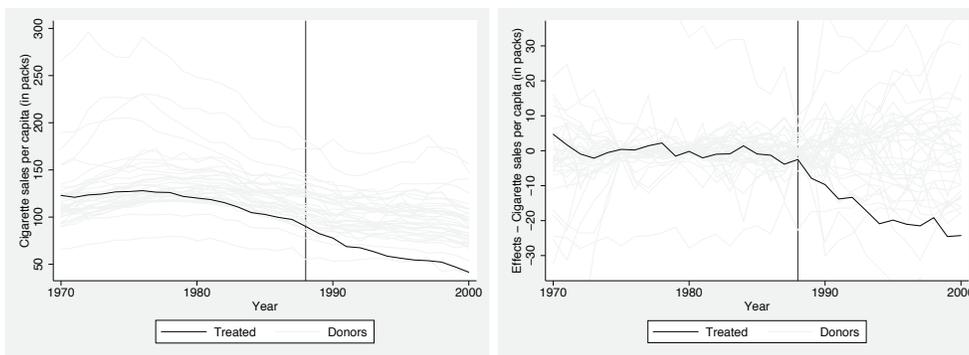
The following are returned by `synth_runner`:

```
. ereturn list
scalars:
      e(n_pl) = 38
      e(n_pl_used) = 38
      e(pval_joint_post) = .131578947368421
      e(pval_joint_post_std) = 0
      e(avg_pre_rmspe_p) = .9210526315789473
macros:
      e(trperiod) : "1989"
      e(trunit) : "3"
      e(treat_type) : "single unit"
      e(depvar) : "cigsale"
      e(cmd) : "synth_runner"
      e(properties) : "b"
matrices:
      e(b) : 1 x 12
      e(pvals_std) : 1 x 12
      e(pvals) : 1 x 12
      e(treat_control) : 31 x 2
. // If truly random, can modify the p-value
. display (e(pval_joint_post_std)*e(n_pl)+1)/(e(n_pl)+1)
.02564103
```

The `e(pval_joint_post)` lists the proportion of effects from control units that have posttreatment RMSPE at least as great as the treated unit. While it is not quite significant at the 10% level, it does not account for pretreatment match quality. The `e(pval_joint_post_std)` value lists the same proportion but scales all values by the relevant pretreatment RMSPE. This measure does show significance. The final measure, `e(avg_pre_rmspe_p)`, is a diagnostic measure noting that the treated unit was matched better than the majority of the control units. If the treatment is considered truly at random, then the true p -value is a modification that adds 1 to the numerator and denominator (in cases with a single treatment). This is shown for the case of the ratio of posttreatment to pretreatment RMSPE.

Next, we create common synthetic control graphs. Note that both `effect_graphs` and `single_treatment_graphs` require variables generated from the `gen_vars` option above. The `single_treatment_graphs` command creates the graphs in figure 1 (which are easy to do when there is a single treatment). The first graphs the outcome path of all units, while the second graphs the prediction differences for all units. The `effect_graphs` command creates the graphs in figure 2. One plots the outcome for the unit and its synthetic control, while the other plots the difference between the two (which for posttreatment is the “effect”). The two previous graphing commands allow the option `trlinediff(real)`, which allows the user to offset a vertical treatment from the first treatment period. Likely options include values in the range from (first treatment period–last posttreatment period) to 0, and the default value is -1 (to match [Abadie, Diamond, and Hainmueller \[2010\]](#)). The `pval_graphs` command creates the graphs in figure 3. These plot the p -values per period for posttreatment periods for both raw and standardized effects.

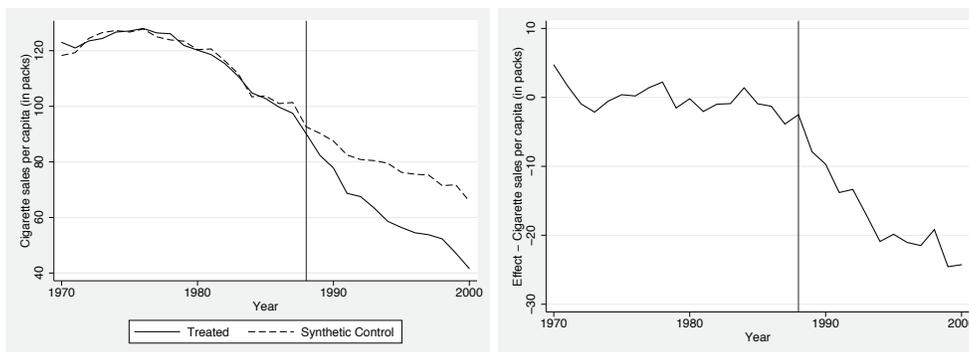
```
. single_treatment_graphs, trlinediff(-1) effects_ylabels(-30(10)30)
> effects_ymax(35) effects_ymin(-35)
. effect_graphs, trlinediff(-1)
. pval_graphs
```



(a) Outcome trends

(b) Effects

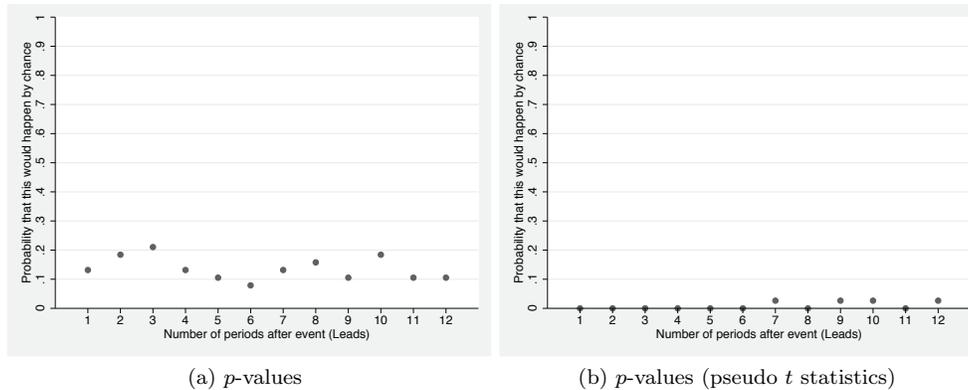
Figure 1. Graphs from `single_treatment_graphs`



(a) Treatment versus control

(b) Effect

Figure 2. Graphs from `effect_graphs`

Figure 3. Graphs from `pval_graphs`

◀

▶ **Example 2**

In this example, we analyze the same treatment but use some of the more advanced options:

```
. sysuse smoking, clear
(Tobacco Sales in 39 US States)

. tsset state year
(output omitted)

. generate byte D = (state==3 & year>=1989)

. synth_runner cigsale beer(1984(1)1988) lnincome(1972(1)1988)
>   retprice age15to24, trunit(3) trperiod(1989) trends
>   training_propr(`=13/19`) pre_limit_mult(10) gen_vars
(output omitted)

. // Proportion of control units that have a higher RMSPE than the
. // treated unit in the validation period:"
. display round(`e(avg_val_rmspe_p)`, 0.001)
.842

. single_treatment_graphs, scaled
. effect_graphs, scaled
. pval_graphs
```

Again, there is a single treatment period, so synthetic control data can be merged into our main dataset. In this setting, we i) specify the treated units or periods with a binary variable; ii) generate the outcome predictors automatically using the initial 13 periods of the pretreatment era (the rest is the “validation” period); iii) match on trends; and iv) limit the control units during inference to those with pretreatment match quality no more than 10 times worse than the match quality of the corresponding treatment units. Now that we have a training or validation period split, there is a new diagnostic. It shows that 84% of the control units have a worse match (a higher RMSPE) during the

validation period. The graphing commands are equivalent. The ones showing the range of effects and raw data are shown in figure 4. One can see that all lines converge on the last pretreatment period because that is the unit by which all are standardized (and all the synthetic controls then match their real units and have zero prediction error).

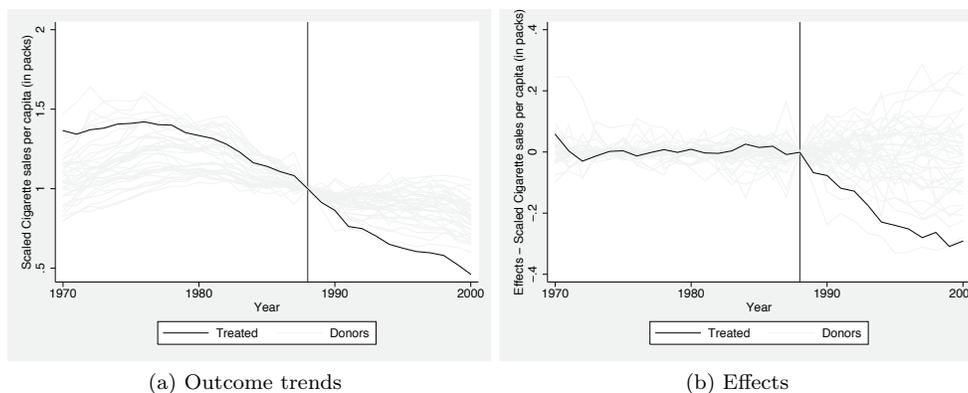


Figure 4. Graphs from `single_treatment_graphs`

◀

► Example 3

The final example involves multiple treatments at different time periods and shows usage of user-generated programs to customize the individual `synth` calls based on treatment year and the unit considered as treated. These programs allow, for instance, that the predictors include the last four periods of beer sales and income as predictors and shows how units can be dropped from the comparison set because of concerns about treatment spillovers.

```
. sysuse smoking, clear
(Tobacco Sales in 39 US States)

. tsset state year
(output omitted)

. program my_pred, rclass
1.     args tyear
2.     local beerv "beer(`='tyear'-4'(1)`,`='tyear'-1')"
```

```
3.     return local predictors "`'beerv' lnincome(`='tyear'-4'(1)`,`='tyear'-1)'"
4. end

. program my_drop_units
1.     args tunit
2.     if `tunit'==39 qui drop if inlist(state,21,38)
3.     if `tunit'==3 qui drop if state==21
4. end
```

```

. program my_xperiod, rclass
1.     args tyear
2.     return local xperiod "`='tyear'-12'(1)='tyear'-1'"
3. end

. program my_mspeperiod, rclass
1.     args tyear
2.     return local mspeperiod "`='tyear'-12'(1)='tyear'-1'"
3. end

. generate byte D = (state==3 & year>=1989) | (state==7 & year>=1988)
. synth_runner cigsale retprice age15to24, d(D) pred_prog(my_pred) trends
>   training_propr(`=13/18`) drop_units_prog(my_drop_units)
>   xperiod_prog(my_xperiod) mspeperiod_prog(my_mspeperiod)
(output omitted)
. effect_graphs
. pval_graphs

```

We extend example 2 by considering a control state now to be treated (Georgia in addition to California). Note that no treatment actually happened in Georgia in 1987. This is just to illustrate additional usage options. With several treatment periods, we cannot automatically include all the synthetic control outputs back into the main dataset. Some graphs (of `single_treatment_graphs`) can also no longer be made. The `effect_graphs` are shown in figure 5. In addition to showing how predictors and unit dropping can be dynamically generated for the underlying `synth` calls, we also show how this can be done for the `xperiod()` and `mspeperiod()` options to `synth`.

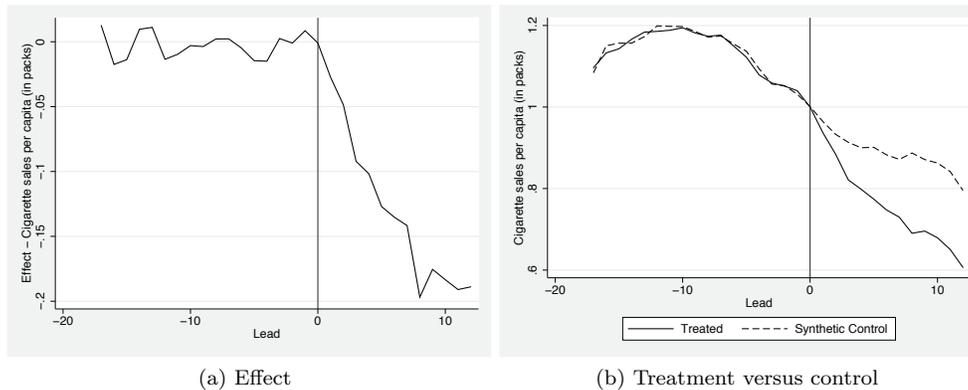


Figure 5. Graphs from `effect_graphs`

◀

4 Discussion

The SCM (Abadie and Gardeazabal 2003; Abadie, Diamond, and Hainmueller 2010) allows researchers to quantitatively estimate effects in many small-sample settings in a

manner grounded by theory. In this article, we provided an overview of the theory of SCM and the `synth_runner` package, which builds on the `synth` package of Abadie, Diamond, and Hainmueller (2010). `synth_runner` provides tools to help with the common tasks of fitting a synthetic control model. It automates the process of conducting in-place placebos and calculating inference on the various possible measures. Following Cavallo et al. (2013), it i) extends the initial estimation strategy to allow for multiple units that receive treatment (at potentially different times); ii) allows for matching on trends in the outcome variable rather than on the level; and iii) automates the process of splitting pretreatment periods into “training” and “validation” sections. It also provides graphs of diagnostics, inference, and estimate effects.

5 Acknowledgment

We thank the Inter-American Development Bank for financial support.

6 References

- Abadie, A., A. Diamond, and J. Hainmueller. 2010. Synthetic control methods for comparative case studies: Estimating the effect of California’s tobacco control program. *Journal of the American Statistical Association* 105: 493–505.
- . 2015. Comparative politics and the synthetic control method. *American Journal of Political Science* 59: 495–510.
- Abadie, A., and J. Gardeazabal. 2003. The economic costs of conflict: A case study of the Basque country. *American Economic Review* 93: 113–132.
- Cavallo, E., S. Galiani, I. Noy, and J. Pantano. 2013. Catastrophic natural disasters and economic growth. *Review of Economics and Statistics* 95: 1549–1561.

About the authors

Sebastian Galiani is a professor of economics at the University of Maryland, College Park.

Brian Quistorff is a research economist at Microsoft AI and Research, Redmond.

Implementing tests for forecast evaluation in the presence of instabilities

Barbara Rossi
ICREA Professor at University of Pompeu Fabra
Barcelona Graduate School of Economics, and
CREI
Barcelona, Spain
barbara.rossi@upf.edu

Matthieu Soupre
University of Pompeu Fabra
Barcelona, Spain
matthieu.soupre@upf.edu

Abstract. In this article, we review methodologies to fix the size distortions of tests for forecast evaluation in the presence of instabilities. The methodologies implement tests for relative and absolute forecast evaluation that are robust to instabilities. We also introduce the `giacross` and `rosssekh` commands, which implement these procedures in Stata.

Keywords: st0501, giacross, rosssekh, forecasting, instabilities, structural change

1 Introduction

Researchers often test models' forecasting ability and are often particularly interested in determining which of two competing forecasting models predicts the best. Such tests are known as “tests of relative forecast comparisons”. Examples of such tests include [Diebold and Mariano \(1995\)](#), [West \(1996\)](#), and [Clark and McCracken \(2001\)](#). Another typical but different type of forecasting ability test involves evaluating whether forecasts fulfill some minimal requirements, such as being unbiased or producing unpredictable forecast errors using any information available when a forecast is made; such tests are typically referred to as “tests of absolute forecasting performance”. Examples of such tests include [Mincer and Zarnowitz \(1969\)](#) and [West and McCracken \(1998\)](#). While both tests of relative and absolute forecast performance are tests of forecasting ability, they differ substantially in their theoretical properties and purpose; in fact, the former are used to compare forecasting models, while the latter are used to evaluate one specific forecasting model.

When applying tests of forecasting ability to macroeconomic time-series data, researchers face an important practical problem: economic time-series data are prone to instabilities. A recent example is the Great Recession of 2007–2009, when several macroeconomic relationships changed drastically. For example, interest rates lost their ability to predict output growth during that time, while credit spreads became useful predictors ([Ng and Wright 2013](#)). Similarly, [Rossi \(2013b\)](#) finds severe instabilities in exchange rate forecasting models. More generally, [Stock and Watson \(1996\)](#) investigated instabilities in different forecasting models in a large dataset of key macroeconomic variables (76 representative U.S. monthly postwar macroeconomic series) using formal testing procedures. The tests for structural breaks that [Stock and Watson](#)

(1996) used include the Quandt (1960) and Andrews (1993) quaslikelihood-ratio test, the mean and exponential Wald test statistics by Andrews and Ploberger (1994), the Ploberger and Krämer (1992) cumulative sum (CUSUM) of squares statistic, and Nyblom's (1989) test. Their analyses uncovered substantial and widespread instabilities in many economic time series. Thus, when researchers test models' forecasting ability, it is potentially important to allow their forecasting ability to change over time. In fact, traditional tests of forecast evaluation are not reliable in the presence of instabilities, which may lead to incorrect inference. The problem arises because traditional tests assume stationarity, which is violated in the presence of instabilities.

In this article, we present the `giacross` and `rosssekh` commands, which illustrate how to test forecast unbiasedness and rationality as well as how to compare competing models' forecasting performance in a way that is robust to the presence of instabilities. The tests are based on methodologies developed by Giacomini and Rossi (2010) and Rossi and Sekhposyan (2016) and discussed thoroughly by Rossi (2013a). The commands we present implement both Rossi and Sekhposyan's (2016) Fluctuation Rationality Test and Giacomini and Rossi's (2010) Fluctuation Test. The tests are separately presented because they address different concerns. For instance, Rossi and Sekhposyan's (2016) Fluctuation Rationality Test allows researchers to evaluate whether the forecasts fulfill some minimal requirements (such as being unbiased and being highly correlated with the ex-post realized value) in environments characterized by instabilities; hence, such tests are "tests of absolute forecasting performance robust to instabilities". Giacomini and Rossi's (2010) Fluctuation Test instead allows researchers to detect which model forecasts the best in unstable environments. Hence, it is a "test of relative forecasting performance robust to instabilities". In the presence of instabilities, the latter tests are more powerful than traditional tests and illustrate when predictive ability appears or breaks down in the data. For each test, we first introduce the test, present the commands that implement it, and then discuss a simple empirical exercise to illustrate the test output and show how to interpret the results.

In section 2, we establish the notation and definitions. Section 3 discusses Rossi and Sekhposyan's (2016) Fluctuation Rationality Test. Section 4 discusses Giacomini and Rossi's (2010) Fluctuation Test. In both sections 3 and 4, we explain the syntax of the commands and demonstrate their usage.

2 Notation and definitions

We first introduce the notation and discuss the assumptions about the data, the models, and the estimation procedures. We are interested in evaluating h -step-ahead forecasts for the variable y_t , which we assume to be a scalar for simplicity. The evaluation can be relative (that is, comparing the relative forecasting performance of competing models) or absolute (that is, evaluating the forecasting performance of a model in isolation).

We assume that the researcher has a sequence of P h -step-ahead out-of-sample forecasts for two models, denoted, respectively, by $y_{t,h}^{(1)}$ and $y_{t,h}^{(2)}$, made at time t , where

$t = 1, \dots, P$.¹ Finally, let the forecast error associated with the h -step-ahead forecast made at time t by the first model be denoted by $v_{t,h}$.²

3 Tests of relative forecast comparisons robust to instabilities

3.1 Giacomini and Rossi's (2010) Fluctuation Test

The Fluctuation Test compares the relative forecasting performances of competing models over time, where the performance is judged based on a loss function chosen by the forecaster. Let $L(\cdot)$ denote the (general) loss function chosen by the researcher and let $L^{(j)}(\cdot)$ denote the loss corresponding to model j , $j = 1, 2$. The researcher can use a sequence of P out-of-sample forecast loss differences, $\{\Delta L_{t,h}\}_{t=1}^P$, where $\Delta L_{t,h} \equiv L_{t,h}^{(1)} - L_{t,h}^{(2)}$, which depend on the realizations of the variable y_{t+h} . For example, for the traditional quadratic loss associated with mean squared forecast error measures, $L_{t,h}^{(1)} = v_{t+h}^2$, and $\Delta L_{t,h}$ is the difference between the squared forecast errors of the two competing models.³ Because the square loss function is the most widely used loss function in practice, we implement it in the procedure described below.

Giacomini and Rossi (2010) define the local relative loss for the two models as the sequence of out-of-sample loss differences computed over rolling windows of size m :

$$m^{-1} \sum_{j=t-m+1}^t \Delta L_{j,h} \quad t = m, m+1, \dots, P \quad (1)$$

They are interested in testing the null hypothesis of equal predictive ability at each point in time,

$$H_0: E(\Delta L_{t,h}) = 0, \forall t$$

and the alternative can be either $E(\Delta L_{t,h}) \neq 0$ (two-sided alternative) or $E(\Delta L_{t,h}) > 0$ (one-sided alternative).

When one considers the two-sided alternative, their Fluctuation Test Statistic is the largest value over the sequence of the (rescaled) relative forecast error losses defined in (1),

$$\max_t |\mathcal{F}_{t,m}^{\text{OOS}}| \quad (2)$$

where

$$\mathcal{F}_{t,m}^{\text{OOS}} = \hat{\sigma}^{-1} m^{-1/2} \sum_{j=t-m+1}^t \Delta L_{j,h} \quad t = m, m+1, \dots, P \quad (3)$$

-
1. The models' parameters are estimated using either a fixed or a rolling scheme, where the size of the sample used to estimate the parameters is fixed. This rules out recursive estimation schemes.
 2. For example, in a simple linear regression model with h -period lagged ($k \times 1$) vector of regressors \mathbf{x}_t , where $E_t y_{t+h} = \mathbf{x}_t' \boldsymbol{\gamma}$, the forecast at time t is $y_{t,h} = \mathbf{x}_t' \hat{\boldsymbol{\gamma}}_{t,R}$ and the forecast error is $v_{t,h} = y_{t+h} - \mathbf{x}_t' \hat{\boldsymbol{\gamma}}_{t,R}$, where $\hat{\boldsymbol{\gamma}}_{t,R}$ is the estimated vector of coefficients.
 3. In fact, $P^{-1} \sum_{t=1}^P \Delta L_{t,h}$ is exactly the mean squared forecast error.

and $\hat{\sigma}^2$ is a heteroskedasticity- and autocorrelation-consistent (HAC) estimator of the long-run variance of the loss differences (Newey and West 1987). The null hypothesis is rejected against the two-sided alternative hypothesis $E(\Delta L_{t,h}) \neq 0$ when $\max_t |\mathcal{F}_{t,m}^{\text{OOS}}| > k_{\alpha,\mu}$, where the critical value $k_{\alpha,\mu}$ depends on the choice of μ , which is the size of the rolling window relative to the number of out-of-sample loss differences P , or formally, $m = \lceil \mu P \rceil$.

Similarly, when one considers the one-sided alternative, the Fluctuation Test Statistic is

$$\max_t \mathcal{F}_{t,m}^{\text{OOS}} \quad (4)$$

and the null hypothesis is rejected in favor of the alternative that model 2 forecasts better at some point in time when $\max_t \mathcal{F}_{t,m}^{\text{OOS}}$ is larger than the one-sided critical value.⁴

Note also that $\mathcal{F}_{t,m}^{\text{OOS}}$ is simply a traditional test of equal predictive ability computed over a sequence of rolling out-of-sample windows of size m .

3.2 The `giacross` command

Syntax

The `giacross` command is the equivalent to the MATLAB command written by Giacomini and Rossi (2010). The `dmariano` command (Baum 2003) is required. To install the `dmariano` command, type `ssc install dmariano` in the Command window. The general syntax of the `giacross` command is

```
giacross realized_value forecast1 forecast2, window(size) alpha(level)
      [nw(bandwidth) side(#)]
```

`realized_value` contains the realizations of the target variable (the realized values against which each forecast is compared), that is, y_{t+h} as per the notation in section 3.1, $t = 1, 2, \dots, P$, where P is the number of forecasts available.

`forecast1` and `forecast2` each contain the forecasts from the competing tested models, that is, $y_{t,h}^{(1)}$ and $y_{t,h}^{(2)}$. Note that the inputs of the function are simply the forecasts; there is no need to input the models' parameter estimates in the procedure. In fact, as explained in Giacomini and Rossi (2010), the test can also be implemented if the researcher does not know the models that generated the forecasts (for example, in the case of survey forecasts).

Options

`window(size)` controls the size of the rolling window used for the test, that is, m . `window()` is required.

4. The critical value for the one-sided test differs from that of the two-sided one.

854 *Implementing tests for forecast evaluation in the presence of instabilities*

`alpha(level)` equals the significance level of the test, either 0.05 for a 5% level or 0.10 for 10%. `alpha()` is required.

`nw(bandwidth)` allows the user to choose the truncation lag used in the estimation of the variance $\hat{\sigma}^2$. If no bandwidth is specified, the truncation lag is automatically determined using the [Schwert \(1987\)](#) criterion.

`side(#)` takes the value 1 or 2 and specifies if the null is compared with a one- or two-sided alternative, respectively. If the alternative is one sided, the alternative hypothesis is that the first model forecasts worse than the second model. If the alternative is two sided, models' forecasts significantly differ from each other under the alternative.

Stored results

`giacross` stores the following in `r()`:

Scalars

<code>r(tstat_sup)</code>	maximum absolute value of the (rolling) test statistic over the sample, that is, the value of (2)
<code>r(cv)</code>	critical value of the test
<code>r(level)</code>	significance level of the test specified by the user

Macros

<code>r(cmd)</code>	<code>giacross</code>
<code>r(cmdline)</code>	command as typed
<code>r(testtype)</code>	whether the test is one or two sided

Matrices

<code>r(RollStat)</code>	whole temporal sequence of $\mathcal{F}_{t,m}^{OOS}$, which is also saved as a variable called <code>FlucTest</code> ; note that <code>FlucTest</code> is not the Fluctuation Test Statistic, which is either (2) or (4), depending on whether the test is two sided or one sided
--------------------------	--

Finally, the `giacross` command automatically produces a graph plotting the test statistic against time with the critical values implied by the specified significance level. We show such a graph in the example in the next section.

3.3 Example of practical implementation in Stata

In what follows, we illustrate how to use the `giacross` command to implement the two-sided test. The comma-separated file we use (`giacross_test_data.csv`, provided with the article files) includes quarterly realizations of inflation for the United States starting in 1968Q4 until 2008Q4 as well as the Greenbook (labeled `forc`) and the Survey of Professional Forecast nowcasts (labeled `spf`) for the same variable.

```

. insheet using giacross_test_data.csv, clear
(5 vars, 161 obs)
. generate year = int(pdate)
. generate quarter = (pdate - int(pdate))*4 + 1
. generate tq = yq(year, quarter)
. format tq %tq
. tsset tq
      time variable:  tq, 1968q4 to 2008q4
              delta:  1 quarter
. * lag length set to 3, default 2-sided test
. giacross realiz forc spf, window(60) alpha(0.05) nw(3)

```

Running the Giacomini - Rossi (2010) test for forecast comparison...

REMINDER

First forecast: forc
Second forecast: spf
Actual series: realiz

Newey - West HAC estimator bandwidth: 3

NOTE: the program generates the following variables for plotting:
2 sided alternative: cvlo and cvhi, which contain the lower and upper critical
> values
1 sided alternative: cvone, which contains the one-sided critical value
FlucTest, which contains the sequence of rolling Giacomini - Rossi test
> statistics

(output omitted)

```

. display "The value of the test statistic is " r(tstat_sup)
The value of the test statistic is 3.1646998
. display "The critical value is " r(cv) " at significance level " r(level)
The critical value is 2.89 at significance level .05
. graph save GR_demo_1, asis replace
(output omitted)

```

Here is how to interpret the results. The value of the test statistic ($\max_t |\mathcal{F}_{t,m}^{\text{OOS}}|$) is 3.1647, which is larger than the critical value at the 5% significance level, or 2.89. Therefore, we reject the null hypothesis that the models' forecasting performance is the same in favor of the alternative—that the first model forecasts significantly better.

The code also returns a graph showing the whole sequence of the Giacomini and Rossi rolling statistic $\mathcal{F}_{t,m}^{\text{OOS}}$, reported in figure 1. The sequence of $\mathcal{F}_{t,m}^{\text{OOS}}$ over time (depicted by a continuous line) is clearly outside the critical value lines (± 2.89 , depicted by the dashed lines). The strongest evidence against the null appears to be around the beginning of the 2000s; this is when the empirical evidence in favor of the first model is the strongest. The figure is saved as GR_demo_1.

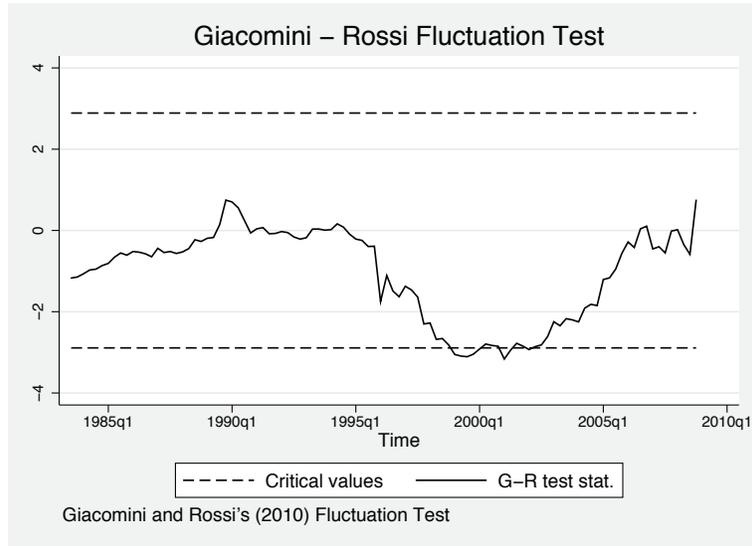


Figure 1. Giacomini and Rossi's test (two sided). The figure depicts $\mathcal{F}_{t,m}^{\text{OOS}}$ from (3) as a function of time (t) for the first example in section 3. The time on the x axis corresponds to the endpoint of the rolling window.

We also include an example of the one-sided version of the test using the following sample code:

```
. * automatic lag length selection based on Schwert criterion, one-sided test
. giacross realiz forc spf, window(60) alpha(0.05) side(1)

Running the Giacomini - Rossi (2010) test for forecast comparison...

REMINDER
First forecast: forc
Second forecast: spf
Actual series: realiz

Newey - West HAC estimator bandwidth chosen automatically with the Schwert
> criterion.

NOTE: the program generates the following variables for plotting:
2 sided alternative: cvlo and cvhi, which contain the lower and upper critical
> values
1 sided alternative: cvone, which contains the one-sided critical value
FlucTest, which contains the sequence of rolling Giacomini - Rossi test
> statistics
(output omitted)
. display "The value of the test statistic is " r(tstat_sup)
The value of the test statistic is .8266927
```

```
. display "The critical value is " r(cv) " at significance level " r(level)
The critical value is 2.624 at significance level .05
. graph save GR_demo_2, asis replace
(output omitted)
```

Here is how to interpret the results. The value of the test statistic is 0.8267, and the critical value is 2.624 at significance level 0.05. The test does not reject the null hypothesis that the two models' forecast performance is the same against the alternative—that the second model forecasts better than the first model.

The output also includes a plot of the models' relative forecasting performance over time, depicted in figure 2.

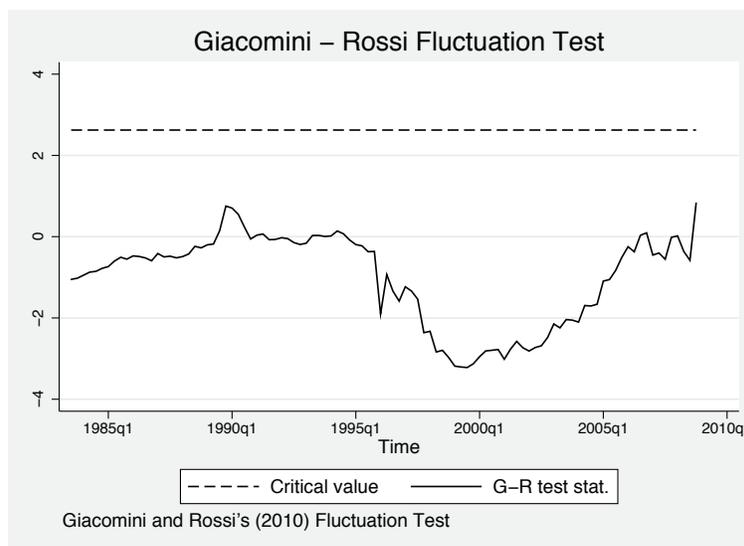


Figure 2. Giacomini and Rossi's test (one sided). The figure depicts $\mathcal{F}_{t,m}^{\text{OOS}}$ from (3) as a function of time (t) for the second example in section 3. The time on the x axis corresponds to the endpoint of the rolling window.

3.4 A comparison with traditional tests

A common test used to compare models' forecasting performance is the Diebold and Mariano (1995) and West (1996) test. The Diebold, Mariano, and West (DMW) test statistic is

$$\text{DMW}_P = \hat{\sigma}^{-1} P^{-1/2} \sum_{t=1}^P \Delta L_{j,h}$$

858 *Implementing tests for forecast evaluation in the presence of instabilities*

where $\hat{\sigma}^2$ is a HAC estimator of the long-run variance of the loss differences. The test is designed to test the (unconditional) null hypothesis $H_0: E(\Delta L_{t,h}) = 0$ and, under the null, has an asymptotic standard normal distribution.

The DMW_P test can be obtained in Stata using the following code:⁵

```
. * Diebold Mariano comparison of forecast accuracy (to compare with GR test)
. dmariano realiz forc spf, max(3)

Diebold-Mariano forecast comparison test for actual : realiz
Competing forecasts: forc versus spf
Criterion: MSE over 161 observations
Maxlag = 3   Kernel : uniform

Series          MSE
-----
forc            1.145
spf            1.338
Difference      -.1935

By this criterion, forc is the better forecast
H0: Forecast accuracy is equal.
S(1) =    -1.233   p-value = 0.2177
```

Here is how to interpret the results. The command yields a p -value of 0.2177, so the test does not reject the null of equal-forecast accuracy of the two forecasts at the 0.05 significance level. Note that the empirical conclusions are very different from those a researcher would obtain with the Fluctuation Test. In fact, the DMW_P test ignores the time variation in the relative forecasting performance over time, visible in figure 1: instead, it averages across all the out-of-sample observations, thus losing power to detect differences in the models' forecasting performance.

4 Tests of absolute forecasting performance robust to instabilities

4.1 Rossi and Sekhposyan's (2016) Fluctuation Rationality Test

Tests for forecast rationality evaluate whether forecasts satisfy some “minimal” requirements, such as being an unbiased predictor or being uncorrelated with any additional information available at the time of the forecast. Thus, traditional tests of forecast rationality (such as [Mincer and Zarnowitz \[1969\]](#) and [West and McCracken \[1998\]](#)) verify that forecast errors have zero mean or that they are uncorrelated with any other variable known at the time of the forecast. However, they assume stationarity and are thus invalid in the presence of instabilities.

To make the tests robust to instabilities, [Rossi and Sekhposyan \(2016\)](#) propose estimating the following forecast rationality regressions in rolling windows (of size m),

$$v_{j,h} = \mathbf{g}'_j \boldsymbol{\theta} + \eta_{j,h} \quad j = t - m + 1, \dots, t \quad (5)$$

5. The DMW_P test statistic is also the same as [Giacomini and White's \(2006\)](#).

where the forecast error denoted by $v_{j,h}$ refers to an h -step-ahead out-of-sample forecast made at time j using data available up to that point in time, \mathbf{g}_j is an $(\ell \times 1)$ vector function of period j data (which can also possibly be a function of the models' parameter estimates), $\boldsymbol{\theta}$ is an $(\ell \times 1)$ parameter vector, and $\eta_{j,h}$ is the residual in the regression. The regression in (5) is thus estimated in rolling windows of size m . At time t , the researcher uses data from $t - m + 1$ to t to obtain the parameter estimate, $\hat{\boldsymbol{\theta}}_t$; by repeating the procedure at times $t = m, m + 1, \dots, P$, the researcher obtains a sequence of parameter estimates over time.

Rossi and Sekhposyan's (2016) main interest is testing forecast rationality in the presence of instabilities. In fact, in the presence of instabilities, tests that focus on the average out-of-sample performance of a model may be misleading because they may average out instabilities. Thus, the hypothesis to be tested is

$$H_0: \boldsymbol{\theta}_t = \boldsymbol{\theta}_0 \text{ versus } H_A: \boldsymbol{\theta}_t \neq \boldsymbol{\theta}_0, \forall t$$

where $\boldsymbol{\theta}_0 = \mathbf{0}$ and $\boldsymbol{\theta}_t$ is the time-varying parameter value.

In (5), we focus on tests of forecast unbiasedness ($\mathbf{g}_t = \mathbf{1}$), forecast efficiency (\mathbf{g}_t is the forecast itself), and forecast rationality (\mathbf{g}_t includes both the forecast and 1).⁶ We refer to tests under the maintained assumption that $\boldsymbol{\theta}_0 = \mathbf{0}$ as "tests for forecast rationality". The zero restriction under the null hypothesis ensures that the forecast errors are truly unpredictable given the information set available at the time the forecast is made.

Rossi and Sekhposyan (2016) propose the following "Fluctuation Rationality" Test,

$$\max_t \mathcal{W}_{t,m} \tag{6}$$

where

$$\mathcal{W}_{t,m} = m \hat{\boldsymbol{\theta}}_t' \hat{\mathbf{V}}_{\theta}^{-1} \hat{\boldsymbol{\theta}}_t \quad \text{for } t = m, m + 1, \dots, P$$

is the Wald test in regressions computed at time t over rolling windows of size m and is based on the parameter estimate $\hat{\boldsymbol{\theta}}_t$, which is sequentially estimated in regression (5), and $\hat{\mathbf{V}}_{\theta}$ is a HAC estimator of the asymptotic variance of the parameter estimate $\hat{\boldsymbol{\theta}}_t$ in the same rolling window.

Here we focus on the version of the Rossi and Sekhposyan (2016) test where parameter estimation error is irrelevant, the forecasts are model free, or the models' parameters are rollingly reestimated in a finite window of data, although their test is valid in more general situations as well (see Rossi and Sekhposyan [2016]).

The null hypothesis is rejected if $\max_t \mathcal{W}_{t,m} > \kappa_{\alpha,\ell}$, where $\kappa_{\alpha,\ell}$ is the critical value at the $100\alpha\%$ significance level with the number of restrictions equal to ℓ .

6. In general, \mathbf{g}_t may also contain any other variable known at time t that was not included in the forecasting model; the framework in (5) also potentially includes tests of forecast encompassing (\mathbf{g}_t is the forecast of the encompassed model) and serial uncorrelation tests (\mathbf{g}_t is the lagged forecast error). See Rossi and Sekhposyan (2016) for details on the implementation in the general case.

4.2 The `rosssekh` command

Syntax

The `rosssekh` command is the equivalent to the MATLAB command written by Rossi and Sekhposyan (2016). The general syntax of the `rosssekh` command is

```
rosssekh realized_value forecast, window(size) alpha(level) [nw(bandwidth)]
```

`realized_value` contains the realizations of the target variable (the realized values against which each forecast is compared), that is, y_{t+h} in the notation of section 3.1, $t = 1, 2, \dots, P$, where P is the number of forecasts available. `forecast` is g_t in our notation.

Options

`window(size)` corresponds to the size of the window in the implementation of the test, that is, m . `window()` is required.

`alpha(level)` equals the significance level of the test, either 0.05 for a 5% level or 0.10 for 10%. `alpha()` is required.

`nw(bandwidth)` allows the user to choose the truncation lag used in the HAC variance estimation. If no bandwidth is specified, the truncation lag is automatically determined using the Schwert (1987) criterion.

Stored results

`rosssekh` stores the following in `r()`:

Scalars

<code>r(tstat_sup)</code>	contains the maximum value attained by the (rolling) test statistic $\mathcal{W}_{t,m}$ over the sample, that is, (6)
<code>r(cv)</code>	matrix of critical values of the test at the level specified by the user
<code>r(level)</code>	level of the test specified by the user

Macros

<code>r(cmd)</code>	<code>rosssekh</code>
<code>r(cmdline)</code>	command as typed

Matrices

<code>r(RollStat)</code>	whole time series of the rolling statistic $\mathcal{W}_{t,m}$, which is also saved in the variable <code>RossSekhTest</code> ; note that the Fluctuation Rationality Test Statistic (6) is the largest value over the sequence
<code>r(CV)</code>	contains the critical values of the test

4.3 Example of practical implementation in Stata

In what follows, we illustrate how to use `rosssekh`. `rosssekh_test_data.csv` is the same as in section 3.3. We focus on evaluating forecast rationality of Greenbook forecasts (labeled `forc`).

```

. rosssekh realiz forc, window(60) alpha(0.05) nw(3)

Running the Rossi - Sekhposyan (2016) forecast rationality test...

Critical value for the test: 10.9084
NOTE: the program generates two variables for plotting:
      - cvrs, which contains the critical values
      - RossSekhTest, which contains the sequence of rolling Rossi -
> Sekhposyan test statistics
  (output omitted)
. display "The value of the test statistic is " r(tstat_sup)
The value of the test statistic is 38.899807
. display "The critical value is " r(cv) " at significance level " r(level)
The critical value is 10.9084 at significance level .05
. graph save RS_demo_1, asis replace
  (output omitted)

```

Here is how to interpret the results. The test statistic ($\max_t \mathcal{W}_{t,m}$) reaches a maximum of 38.90, and the critical value is 10.9084. Thus, the test rejects the null hypothesis of forecast rationality.

The code also produces a graph showing Rossi and Sekhposyan's (2016) sequence of test statistics $\mathcal{W}_{t,m}$ over time [defined in (6)], reported in figure 3. The sequence of $\mathcal{W}_{t,m}$ (depicted by a continuous line) is clearly outside the critical value line (depicted by the dashed line). The strongest evidence against forecast rationality appears to be around the beginning of 1995. The figure is saved as RS_demo_1.

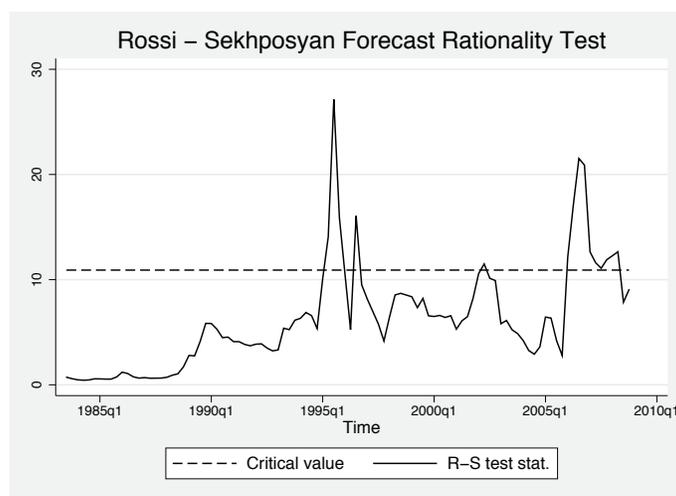


Figure 3. Rossi and Sekhposyan's (2016) test. The figure plots Rossi and Sekhposyan's (2016) sequence of test statistics ($\mathcal{W}_{t,m}$) over time. The time on the x axis corresponds to the endpoint of the rolling window.

862 *Implementing tests for forecast evaluation in the presence of instabilities*

A similar result can be obtained by using an automatic lag length selection with the following sample code:

```
. * automatic lag length selection, integer part of window^0.25
. rosssekh realiz forc, window(60) alpha(0.05) nw(0)

Running the Rossi - Sekhposyan (2016) forecast rationality test...

Critical value for the test: 10.9084
NOTE: the program generates two variables for plotting:
      - cvrs, which contains the critical values
      - RossSekhTest, which contains the sequence of rolling Rossi -
> Sekhposyan test statistics
  (output omitted)
. display "The value of the test statistic is " r(tstat_sup)
The value of the test statistic is 27.139633
. display "The critical value is " r(cv) " at significance level " r(level)
The critical value is 10.9084 at significance level .05
. graph save RS_demo_2, asis replace
  (output omitted)
```

Here is how to interpret the results. The test statistic reaches a maximum of 27.14 for a critical value of 10.9084. The test does reject the null hypothesis of forecast rationality. In this case, the plot is qualitatively similar to that in figure 3. Therefore, we do not report it to save space.

4.4 A comparison with traditional tests

A common test to evaluate the forecasting performance of a model is the Mincer and Zarnowitz (1969) test. The Mincer and Zarnowitz (MZ) (1969) test statistic, MZ_P , is a simple F test in the regression $v_{j,h} = \mathbf{g}'_j \boldsymbol{\theta} + \eta_t$, $j = 1, \dots, P$,

$$MZ_P = P \widehat{\boldsymbol{\theta}}_P' \widehat{\mathbf{V}}_\theta^{-1} \widehat{\boldsymbol{\theta}}_P$$

where $\widehat{\mathbf{V}}_\theta$ is a HAC estimator of the asymptotic variance of the parameter estimates.

The test is designed to test the (unconditional) null hypothesis that $H_0: \boldsymbol{\theta} = \mathbf{0}$ and, under the null, has an asymptotic chi-squared distribution with ℓ degrees of freedom. Again, note that, unlike $\max_t \mathcal{W}_{t,m}$, it is not robust to instabilities.

The MZ_P test can be obtained in Stata from a simple F test as follows:⁷

```
. * Mincer Zarnowitz regression for systematic forecast bias
> (to compare with RS test)
. generate fcsterror=realiz-forc
. newey fcsterror forc, lag(3)
Regression with Newey-West standard errors      Number of obs   =      161
maximum lag: 3                                F( 1,          159) =      0.60
                                                Prob > F          =      0.4386
```

fcsterror	Newey-West		t	P> t	[95% Conf. Interval]	
	Coef.	Std. Err.				
forc	-.0421532	.0542834	-0.78	0.439	-.1493628	.0650564
_cons	.0745427	.1953463	0.38	0.703	-.3112655	.460351

```
. newey fcsterror spf, lag(3)
Regression with Newey-West standard errors      Number of obs   =      161
maximum lag: 3                                F( 1,          159) =      0.03
                                                Prob > F          =      0.8565
```

fcsterror	Newey-West		t	P> t	[95% Conf. Interval]	
	Coef.	Std. Err.				
spf	-.0100134	.0552681	-0.18	0.856	-.1191676	.0991409
_cons	-.0540947	.203149	-0.27	0.790	-.4553132	.3471238

Here is how to interpret the results. The MZ_P test statistic is 0.60 and its p -value is 0.4386, so the test does not reject the null at the 0.05 significance level. Again, in this case, the empirical conclusions differ from those that a researcher would obtain by using the Fluctuation Rationality Test. In fact, the MZ_P test ignores the time variation in the relative forecasting performance over time, visible in figure 2; by averaging across all the out-of-sample observations, it misses the lack of forecast rationality that appears sporadically in time.

5 References

- Andrews, D. W. K. 1993. Tests for parameter instability and structural change with unknown change point. *Econometrica* 61: 821–856.
- Andrews, D. W. K., and W. Ploberger. 1994. Optimal tests when a nuisance parameter is present only under the alternative. *Econometrica* 62: 1383–1414.
- Baum, C. 2003. dmario: Stata module to calculate Diebold–Mariano comparison of forecast accuracy. Statistical Software Components S433001, Department of Economics, Boston College. <http://econpapers.repec.org/software/bocbocode/s433001.htm>.

7. We used a lag length equal to 3 to compare the results with those in the previous example.

864 **Implementing tests for forecast evaluation in the presence of instabilities**

- Clark, T. E., and M. W. McCracken. 2001. Tests of equal forecast accuracy and encompassing for nested models. *Journal of Econometrics* 105: 85–110.
- Diebold, F. X., and R. S. Mariano. 1995. Comparing predictive accuracy. *Journal of Business and Economic Statistics* 13: 253–263.
- Giacomini, R., and B. Rossi. 2010. Forecast comparisons in unstable environments. *Journal of Applied Econometrics* 25: 595–620.
- Giacomini, R., and H. White. 2006. Tests of conditional predictive ability. *Econometrica* 74: 1545–1578.
- Mincer, J., and V. Zarnowitz. 1969. The evaluation of economic forecasts. In *Economic Forecasts and Expectations: Analysis of Forecasting Behavior and Performance*, ed. J. Mincer, 3–46. New York: National Bureau of Economic Research.
- Newey, W. K., and K. D. West. 1987. A simple, positive semi-definite, heteroskedasticity and autocorrelation consistent covariance matrix. *Econometrica* 55: 703–708.
- Ng, S., and J. H. Wright. 2013. Facts and challenges from the Great Recession for forecasting and macroeconomic modeling. *Journal of Economic Literature* 51: 1120–1154.
- Nyblom, J. 1989. Testing for the constancy of parameters over time. *Journal of the American Statistical Association* 84: 223–230.
- Ploberger, W., and W. Krämer. 1992. The CUSUM test with OLS residuals. *Econometrica* 60: 271–285.
- Quandt, R. E. 1960. Tests of the hypothesis that a linear regression system obeys two separate regimes. *Journal of the American Statistical Association* 55: 324–330.
- Rossi, B. 2013a. Advances in forecasting under instability. In *Handbook of Economic Forecasting*, vol. 2B, ed. G. Elliott and A. Timmermann, 1203–1324. Amsterdam: Elsevier.
- . 2013b. Exchange rate predictability. *Journal of Economic Literature* 51: 1063–1119.
- Rossi, B., and T. Sekhposyan. 2016. Forecast rationality tests in the presence of instabilities, with applications to federal reserve and survey forecasts. *Journal of Applied Econometrics* 31: 507–532.
- Schwert, G. W. 1987. Effects of model specification on tests for unit roots in macroeconomic data. *Journal of Monetary Economics* 20: 73–103.
- Stock, J. H., and M. W. Watson. 1996. Evidence on structural instability in macroeconomic time series relations. *Journal of Business and Economic Statistics* 14: 11–30.
- West, K. D. 1996. Asymptotic inference about predictive ability. *Econometrica* 64: 1067–1084.

West, K. D., and M. W. McCracken. 1998. Regression-based tests of predictive ability. *International Economic Review* 39: 817–840.

About the authors

Barbara Rossi is an ICREA professor of economics at the University of Pompeu Fabra. She is a CEPR Fellow, a member of the CEPR Business Cycle Dating Committee, and a director of the International Association of Applied Econometrics. Funding from the ERC through Grant 615608 is gratefully acknowledged.

Matthieu Soupre is a PhD student of economics at the University of Pompeu Fabra.

Text mining with n -gram variables

Matthias Schonlau
University of Waterloo
Waterloo, Canada
schonlau@uwaterloo.ca

Nick Guenther
University of Waterloo
Waterloo, Canada
nguenthe@uwaterloo.ca

Ilia Sucholutsky
University of Waterloo
Waterloo, Canada
isucholu@uwaterloo.ca

Abstract. Text mining is the process of turning free text into numerical variables and then analyzing them with statistical techniques. We introduce the command `ngram`, which implements the most common approach to text mining, the “bag of words”. An n -gram is a contiguous sequence of words in a text. Broadly speaking, `ngram` creates hundreds or thousands of variables, each recording how often the corresponding n -gram occurs in a given text. This is more useful than it sounds. We illustrate `ngram` with the categorization of text answers from two open-ended questions.

Keywords: st0502, ngram, bag of words, sets of words, unigram, gram, statistical learning, machine learning

1 Introduction

Text mining is the process of turning free text into numerical variables and then analyzing them with statistical techniques. Because the number of variables can be large and can exceed the number of observations, statistical or machine-learning techniques play a large role in the analysis. (“Statistical learning” and “machine learning” are synonymous, reflecting important contributions by both statisticians and computer scientists to the field. We use the term “statistical learning” in this article.)

Text mining is useful in many contexts: For example, it can be used to categorize or cluster answers to open-ended questions in surveys. Text mining can also be used to build spam filters for emails; for example, if you have ever been solicited for money by an alleged Nigerian prince, you will agree that the presence of the two-word sequence “Nigerian prince” is a strong indicator of spam. Text mining can also be used for authorship attribution (Madigan et al. 2005), including plagiarism; for example, it would be suspicious if two texts contained the same long sequence of words.

The most popular approach to text mining is based on n -gram variables. This approach is often called the “bag of words” because it simply counts word occurrences and mostly¹ ignores word order. (A lesser-used term is “set of words”, referring to recording only the presence or absence of a word rather than word frequency.)

The outline of this article is as follows: Section 2 explains how text is “numericized”, that is, how text is turned into numerical variables. Sections 3 and 4 give examples ana-

1. Unigram (1-gram) variables ignore word order completely. Higher-order n -gram variables partially recover word order.

lyzing open-ended questions. They each feature a different statistical learning algorithm. Finally, section 5 concludes with a discussion.

2 Text mining with n-gram variables

This section explains how to turn text into numerical variables for analyses. A variable containing frequency counts of a single word in each text is called a unigram. For example, consider the four texts shown in table 1. Each column represents an x variable. Each cell gives the frequency of a word in the given text. The last column, `n_token`, records the number of words in the text.

Table 1. Example: Turning text into unigrams (single-word variables)

text	t_4	t_advice	t_daily	t_fun	t_funny	t_hike	t_hiked	t_hikes	t_hiking	t_john	t_machu	t_ooooh	t_picchu	t_take	t_times	n_token
Take my hiking advice: Hiking is fun.	0	1	0	1	0	0	0	0	2	0	0	0	0	1	0	7
John, ooooh John, take a hike!	0	0	0	0	0	1	0	0	0	2	0	1	0	1	0	6
John is funny: he hikes daily.	0	0	1	0	1	0	0	1	0	1	0	0	0	0	0	6
He hiked to Machu Picchu 4 times.	1	0	0	0	0	0	1	0	0	0	1	0	1	0	1	7

The variables in table 1 were generated with the command²

```
ngram text, degree(1) threshold(1)
```

The option `degree(1)` refers to unigrams. That is, variables contain counts of individual words (rather than from multiword sequences). When words appear too infrequently, corresponding unigrams are typically discarded. The option `threshold(1)` means that all resulting variables are retained, even if an individual word appears only in a single text.

Very common words are called “stopwords”, and their presence or absence in the text is unlikely to help one differentiate among the texts. By default, `ngram` removes stopwords. Table 1 does not contain variables for the stopwords “he”, “is”, “to”, and “my”.

2. To install the `ngram` command introduced in this article, type

```
net install ngram, from(http://www.schonlau.net/stata)
```

`ngram` is available for Windows (64-bit), Mac (64-bit), and Linux (Ubuntu, 64-bit).

The words “hiked”, “hikes”, and “hike” all have the same root word. Because these words typically refer to the same meaning, it is often useful to create only a single variable for them. This is accomplished by stemming each (different) word to its (identical) stem. There are many different stemming algorithms. We have implemented the [Porter \(1980\)](#) stemmer, which is arguably the most popular stemmer. Adding option `stemmer`, we see that the command becomes

```
ngram text, degree(1) threshold(1) stemmer
```

The result is shown in table 2.

Table 2. Unigrams with stemming

text	t_4	t_advic	t_dai1l	t_fun	t_funni	t_hike	t_john	t_machu	t_oooh	t_picchu	t_take	t_time	n_token
Take my hiking advice: Hiking is fun.	0	1	0	1	0	2	0	0	0	0	1	0	7
John, oooh John, take a hike!	0	0	0	0	0	1	2	0	1	0	1	0	6
John is funny: he hikes daily.	0	0	1	0	1	1	1	0	0	0	0	0	6
He hiked to Machu Picchu 4 times.	1	0	0	0	0	1	0	1	0	1	0	1	7

The Porter algorithm strips the ending of words or changes them slightly. For example, “funny” is now listed as “funni” (so that “funnier” and “funny” reduce to the same stem). All three variables related to “hike” now have the same stem, “hike”, and only one variable is created instead of three.

We started out creating unigrams, that is, single-word variables. In addition, we can create variables for each two-word sequence. Such variables are called bigrams. Table 3 contains such bigrams, created with this command:

```
ngram text, degree(2) threshold(1) stemmer
```

Table 3. Bigrams with stemming

	t_4	t_advic	t_daili	t_fun	t_funni	t_hike	t_john	t_machu	t_oooh	t_picchu	t_take	t_time	t_STX_hike	t_STX_john	t_STX_take	t_4_time	t_advic_hike	t_daili_ETX	t_fun_ETX	t_funni_hike	t_hike_ETX	t_hike_advic	t_hike_daili	t_hike_fun	t_hike_machu	t_john_funni	t_john_oooh	t_john_take	t_machu_picchu	t_oooh_john	t_picchu_4	t_take_hike	t_time_ETX	n_token			
	0	1	0	1	0	2	0	0	0	0	1	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	1	0	1	0	7
	0	0	0	0	0	1	2	0	1	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	1	0	1	0	1	0	6	
	0	0	1	0	1	1	1	0	0	0	0	0	1	0	0	0	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	6	
	1	0	0	0	0	1	0	1	0	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	1	0	7	

As you can see, the number of variables increases quite rapidly. There are also some new variables starting with STX and ending in ETX. For example, STX_hike means the word stem “hike” appears as the first nonstopword in the text. Analogously, time_ETX means that the word stem “time” appears as the last nonstopword in a text.

A threshold value specifies how many observations an n -gram needs to occur in before a variable is created. We increase the threshold value to 2:

```
ngram text, degree(2) threshold(2) stemmer
```

The result is shown in table 4. In a more realistic example, one might set this threshold to a larger value. Variables that are only nonzero for some observations are not useful for making predictions. By default, the threshold is set to 5.

Table 4. Bigrams with threshold 2

text	t_hike	t_john	t_take	t_STX_john	t_take_hike	n_token
Take my hiking advice: Hiking is fun.	2	0	1	0	1	7
John, oooh John, take a hike!	1	2	1	1	1	6
John is funny: he hikes daily.	1	1	0	1	0	6
He hiked to Machu Picchu 4 times.	1	0	0	0	0	7

When the texts are short, it is often sufficient to know whether an n -gram is present or absent; the count is not actually needed. This can be accomplished with the option `binarize` as follows:

```
ngram text, degree(2) threshold(2) stemmer binarize
```

The result is shown in table 5.

Table 5. Bigrams with threshold 2 and binarizing

text	t_hike	t_john	t_take	t_STX_john	t_take_hike	n_token
Take my hiking advice: Hiking is fun.	1	0	1	0	1	7
John, oooh John, take a hike!	1	1	1	1	1	6
John is funny: he hikes daily.	1	1	0	1	0	6
He hiked to Machu Picchu 4 times.	1	0	0	0	0	7

We have described the core of a simple but powerful mechanism to turn text into variables. We next address how this works in foreign languages.

2.1 Internationalization

Language settings affect `ngram` for the removal of stopwords and for stemming. By default, `ngram` uses the language specified in the current locale. This can be reset to any other language. For example, to set the current locale to English, specify

```
set locale_functions en
```

n -grams can be created for any language.

Stopwords

Stopwords can be removed for any language. We have included stopword files for all languages for which a stemmer is implemented. For other languages, lists of stopwords are readily available on the Internet and can easily be added by specifying the file `stopwords_la.txt` on the `adopath` (for example, in the current folder), where *la* is the two-letter language code of the current locale. If you want to see what words are considered stopwords, the default file for English is `C:\ado\plus\s\stopwords_en.txt`.

Stemming

An ideal stemmer stems all words that are derived from the same root to the same stem. “Understemming” refers to creating word groups that still have multiple roots; that is, the groups are not separated enough. “Overstemming” refers to creating more than one group for words with the same root; that is, groups are separated more than needed. The Porter stemmer is considered a light stemmer (Kraaij and Pohlmann 1994). Typically, light stemming leads to better results than heavy stemming (Paice 1994).

In European languages, stemming can lead to large gains (Manning, Raghavan, and Schütze 2008, 32). However, the extent of the benefit varies by language, and different evaluations have reported different gains. Hollink et al. (2004) investigated Porter stemmers in eight European languages. While stemming always improves average precision, they found that there were much larger gains in some languages than in others. Noticeably, there were large gains for Finnish (30%), Spanish (10.5%), and German (7.3%), middling gains in English (4.9%) and Italian (4.9%), and smaller gains in Dutch (1.2%), French (1.2%), and Swedish (1.7%). Savoy (2006, 1034) reports a Porter stemmer improved the mean average precision on average by 30.5% for French, 7.7% for Portuguese, and 12.4% for German. Gaustad and Bouma (2002) found that a Dutch stemming did not consistently improve prediction accuracy. Hull (1996) concludes that, for English, “some form of stemming is almost always beneficial”.

Stemming is available for the following locales: `da` (Danish), `de` (German), `en` (English), `es` (Spanish), `fr` (French), `it` (Italian), `nl` (Dutch), `no` (Norwegian), `pt` (Portuguese), `ro` (Romanian), `ru` (Russian), and `sv` (Swedish). The language-specific Porter stemmers we use are described at <http://snowball.tartarus.org/>.

2.2 Additional program options

The option `n_token` adds a variable with the number of words of the original text (that is, with stopwords). This variable is often useful and is supplied by default.

All text is first converted into lowercase by default. This behavior can be disabled by specifying the option `lower`.

The option `punctuation` adds variables for punctuation. For example, a separate unigram is created for the symbol “?” if it appears. What is considered punctuation is language dependent.

2.3 Statistical learning

After one turns the text into numerical variables, the number of variables are typically large and sometimes larger than the number of observations. If so, linear and logistic regression will fail, because they require estimating more parameters than observations. There is also a high degree of collinearity in the x matrix, because most values are 0; most words are not found in most texts. Because of these reasons, we turn to statistical learn-

ing. In Stata, there are at least two alternatives: boosting (Schonlau 2005), which implements the algorithm for gradient boosting (Hastie, Tibshirani, and Friedman 2009), and support vector machines (Guenther and Schonlau 2016), invented in the '90s by Vapnik (Vapnik 2000).

3 Beliefs about immigrants

As part of their research on cross-national equivalence of measures of xenophobia, Braun, Behr, and Kaczmirek (2013) categorized answers to open-ended questions on beliefs about immigrants. The questions were part of the 2003 International Social Survey Program questionnaire on “National Identity”. The authors investigated identical questions asked in multiple countries (in different languages); for simplicity, we focus here on the German survey with 1,006 observations. The questionnaire contained four statements about immigrants, such as

“Immigrants take jobs from people who were born in Germany”.

Respondents were asked to rate this statement on a 5-point Likert scale ranging from “strongly disagree” to “strongly agree”. After each question, respondents were then asked an open-ended question:

“Which type of immigrants were you thinking of when you answered the question? The previous statement was: [text of the respective item repeated].”

This question is then categorized by (human) raters into the following categories (Braun, Behr, and Kaczmirek 2013):

- General reference to immigrants
- Reference to specific countries of origin, ethnicities, or both (Islamic countries, eastern Europe, Asia, Latin America, sub-Saharan countries, Europe, and Gypsies)
- Positive reference of immigrant groups (“people who contribute to our society”)
- Negative reference of immigrant groups (“any immigrants that[...] cannot speak our language”)
- Neutral reference of immigrant groups “immigrants who come to the United States primarily to work”)
- Reference to legal or illegal immigrant distinction (“illegal immigrants not paying taxes”)
- Other answers (“no German wants these jobs”)
- Nonresponse, incomprehensible answer, or unclear answer (“it’s a choice”)

3.1 Creating n -gram variables

The text variable is called `probe_all`. We specify an n -gram of degree 2 (unigrams and bigrams), stemming, and set the threshold at 5:

```
set locale_functions de
ngram probe_all, degree(2) threshold(5) stemmer binarize
```

The first line, `set locale_functions de`, signals that German stemming and the German stopword list are to be used.

`ngram` created 242 n -grams and a variable that counts the number of words, `n_token`. How much does the number of variables change as a function of degree? By changing the `degree()` option and rerunning the program, we can find out that there are 167 unigrams (`degree(1)`). As we already know, the number of variables rises to 242 when including bigrams (`degree(2)`). The number further rises to 256 when including trigrams (`degree(3)`).

Out of curiosity, we briefly assess the effect of stemming and stopwords on the number of variables for these data. Stemming reduced the number of variables only by 19. When we specify bigrams (`degree(2)`) without stemming, the number of variables reduces from 242 to 223. Removing stopwords has a substantial effect; when we specify bigrams with stemming but do not remove stopwords (`stopwords(.)`), the number of variables increases from 242 to 423.

3.2 Statistical learning

Next, we apply statistical-learning techniques to the n -gram variables. Here we use the implementation (Schonlau 2005) of (gradient) boosting (Hastie, Tibshirani, and Friedman 2009). The term “boosting” is used differently in statistics and computer science. In statistics, this term refers to the multiplicative algebraic reconstruction technique algorithm popularized by Hastie, Tibshirani, and Friedman (2009). Computer scientists think of “boosting” as a generic term for a large number of related algorithms.

Statistical-learning models are black-box models and generally difficult to interpret. In boosting, the concept of “influential variables” makes interpreting the black-box model easier. We choose 500 observations as training data identified by the variable `train` and compare several runs with and without stemming in German, removal of German stopwords, and binarizing of count variables.

```
boost y t_* n_token if train, dist(multinomial) influence pred(pred) ///
seed(12) interaction(3) shrink(.1)
```

`n_token` is the number of words in the text answer. By default, all n -grams have the prefix `t_`, making it easy to specify all n -grams. The options `interaction(3)` and `shrink(.1)` refer to boosting-specific tuning parameters that are not further explained here. The option `seed()` passes a seed for the random generation of numbers, making results reproducible.

The accuracy of predicted categories on the test data is shown in table 6. The first row represents typical choices: stemming, stopwords, and binarizing. Surprisingly, the runs that do not remove stopwords have the highest accuracy. We will comment on this later.

Table 6. Performance metrics for boosting the immigrant data for different options

German stemming	Remove German stopwords	binarize	Accuracy
yes	remove	yes	61.9 %
yes	remove	no	62.5 %
no	remove	yes	61.9 %
no	keep	yes	68.2%
yes	keep	yes	71.2%

We proceed with the run with the highest accuracy in the last row of table 6. For this choice, table 7 shows the breakdown of accuracy by response category. For the largest two categories, “general” and “type of foreigner”, and for category “nonproductive”, accuracy is very high. Accuracy is low for smaller categories (“positive”, “negative”, “legal/illegal”, “neutral”) and middling for “other”.

Table 7. Immigrant data: Accuracy by response category on the test data

y	Accuracy	N
positive	25%	36
negative	30%	20
neutral	40%	10
general	81%	144
type of foreigner	86%	146
legal/illegal	0%	1
other	63%	110
nonproductive	82%	38

Each of the eight response categories in multinomial boosting is associated with a set of influential variables. Influences are measured in percentages, and the influence of all variables sums to 100%. The influential variables for outcome “general” are shown in figure 1. The most influential variable is “all” (with the same meaning in English). The second most influential variable, “allgemein”, means “general” in English. The bigram “kein_bestimmt” translates to “no particular” as in “no particular type of foreigner”. Several other influential variables refer to general groups of foreigners, such as stemmed words of “nationality” and “foreigners”.

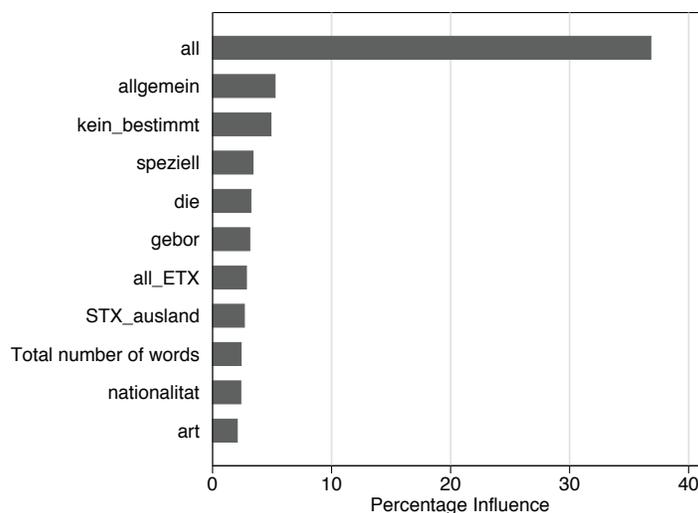


Figure 1. Influential variables for outcome “general”. Variables with influence of less than 2% are not shown.

Influential variables for outcome “nonproductive” are shown in figure 2. Because nonproductive answers are typically very short (a word or two), the most influential variable in determining whether the entry is nonproductive is whether the line is empty: `STX_ETX` is the bigram where the beginning of the line is followed by the end of the line. The second most important variable is the number of words: nonproductive, nonempty lines such as “-”, “.”, and “???” have a zero number of words. `kein_ETX` and `nicht_ETX` refer to the words “kein” (no, none) or “nicht” (not) appearing as the last word in the text. Looking at the stopword file, we see that several influential words for this outcome were all stopwords: “kein” (no, none), “nicht” (not), “ich” (I), and “kann” (can). This is unusual and explains why it was a bad idea to exclude stopwords in this case. In most studies, it is a good idea to remove stopwords.

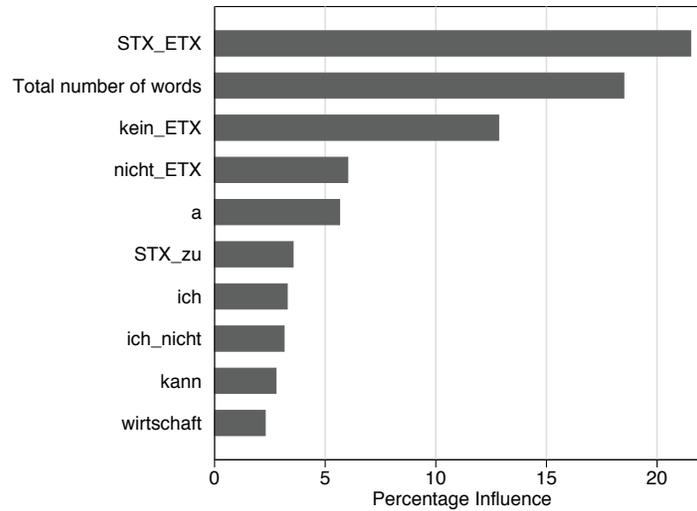


Figure 2. Influential variables for outcome “nonproductive”. Variables with influence of less than 2% are not shown.

In conclusion, with 8 different outcome categories, prediction accuracy on a test dataset was just over 70%. Retaining stopwords was important here.

4 Example: Final comments in surveys

Many surveys include an open-ended question near the end such as “Do you have any other comment?” A subset of such comments in the Longitudinal Internet Studies for the Social Sciences panel and the Dutch Immigrant panel has been categorized into eight mutually exclusive types of comments: survey was difficult, survey contained an error, survey was too long, survey did not apply to me, survey or questions were unclear, negative comments not otherwise categorized, positive comment, and neutral comment (Schonlau 2015). The survey was conducted in Dutch. The data consist of 699 categorized answers and 1,400 uncategorized answers. The frequencies of the categorized answers are given in table 8. Neutral comments dominate; there are 3 categories with low frequency (≤ 11), making it harder to build a model to predict them. The goal is to a) create n -gram from the comments, b) use those n -gram to build a model, and c) use the model to predict the content categories on uncategorized comments.

Table 8. Frequencies of categorized answers shown in descending order by frequency

Comment category	Frequency	Percent(%)
neutral comment	306	43.8
negative comment	196	28.0
survey difficult	73	10.4
questions unclear	72	10.3
does not apply	23	3.3
survey contains error	11	1.6
positive comment	10	1.4
survey too long	8	1.1
total	699	100.0

We first create unigrams of words that occur in at least five answers. Dutch stopwords are removed, the words are stemmed to their Dutch root, and indicator variables (rather than word counts) are used to record when a text contains a word. The texts are substantially longer than in the previous example.

```
set locale_functions nl
ngram eva_comment, degree(1) threshold(5) stemmer  binarize
```

This creates about 800 unigrams as well as the variable `n_token`, which contains the number of words in the answer. Now we are ready to run a statistical learning algorithm on the data. Here we choose support vector machines (Guenther and Schonlau 2016):

```
svmachines category n_token t_* if category!=., type(svc) ///
kernel(linear) c(1) predict cat_pred
```

The dependent variable `category` contains eight different values, one for each category. Uncategorized data are excluded with the `if` statement. Specifying a linear kernel for the support vector machines is usually sufficient in text mining. The linear kernel requires only one tuning parameter, c .

With learning algorithms, it is important to find good values for tuning parameters. Splitting the data into random training data and test datasets, we try a wide range of choices for the c parameter: 100, 10, 1, 0.1, 0.01, 0.001. The c -parameter that gives the best performance on the test dataset happens to be the default value $c = 1$. The accuracy with a training/test split of 500/199 is 0.608; that is, 60.8% of predictions were correct. This is not a fabulous result but still much better than predicting the modal category “neutral comment”, which would have had an accuracy of 43.8%. This highlights the importance of establishing anew how well text mining is working for each application.

We now look at predictions of the uncategorized 1,400 observations. The distribution of the predicted categories is shown in table 9. Relative to the distribution in table 8, more common categories tend to be overpredicted at the expense of rare categories. It

makes sense that, in case of doubt, the more common category is chosen. However, in the aggregate, this distorts the distribution toward such categories.

The distortion in the distribution can be corrected by averaging the probabilities of predicting each category rather than the average number a category is predicted. Even if a rare category is never the most likely for any one answer, small probabilities add up. To obtain the probabilities, we rerun `svmmachines` with the probability option for both model building and prediction:

```
svmmachines category n_token t_* category!=. , type(svc) kernel(linear) c(1) prob
predict p prob
summarize p*
```

The average probabilities are shown in the last column of table 9. The distribution based on the sum of probabilities is strikingly similar to the distribution of the random sample in table 8.

Table 9. Predicted categories of 1,400 previously uncategorized answer texts

Comment category	Frequency	Predictions (%)	Average probabilities (%)
neutral comment	740	52.9	42.0
negative comment	412	29.4	27.9
survey difficult	119	8.5	10.7
questions unclear	78	5.6	10.4
does not apply	32	2.3	3.8
survey too long	9	0.6	1.5
survey contains error	7	0.5	1.7
positive comment	3	0.2	1.7
total	1400	100	100

Sometimes, the categorizations will be used as x variables in subsequent regressions. [McCaffrey and Elliott \(2008\)](#) addressed a similar problem in the context of multiple imputation. Their recommendation was to use estimated probabilities, rather than indicator variables of predicted categories, as x variables. The same recommendation applies here.

5 Discussion

The n -gram approach to text mining is not a panacea. It tends to work better for shorter texts, because the presence and absence of words is more remarkable than in longer texts with many words.

A related command, `txttool` (Williams and Williams 2014), also creates unigrams, removes stopwords, and uses the Porter algorithm for stemming. The command introduced here, `ngram`, is more comprehensive than `txttool`, because it also allows for higher order n -gram variables (that is, bigrams and trigrams in addition to unigrams), stemming in languages other than English, and the use of punctuation (for example, question marks) as n -gram variables. However, unlike `ngram`, the `txttool` command is written in Mata.

We hope we have provided a useful entry point to text mining with this approach. Text mining extends far beyond what we have been able to cover; there are many book-length treatments. Manning and Schütze (1999) is a classic book written by computer scientists. Manning, Raghavan, and Schütze (2008) and Büttcher, Clarke, and Cormack (2016) are two popular books in information retrieval (for example, searches in a search engine such as Google). Ignatow and Mihalcea (2016) is specifically aimed at social scientists. Jockers (2014) gave extensive examples in R aimed at students of literature.

Acknowledgments

This research was supported by the Social Sciences and Humanities Research Council of Canada (SSHRC # 435-2013-0128 and # 430-2013-0301). This article uses data from the Longitudinal Internet Studies for the Social Sciences panel administered by CentERdata (Tilburg University, The Netherlands). Dr. Dorothee Behr kindly provided us with the GESIS immigrant data. We are grateful for the data. We thank an anonymous referee for his or her comments.

6 References

- Braun, M., D. Behr, and L. Kaczmirek. 2013. Assessing cross-national equivalence of measures of xenophobia: Evidence from probing in web surveys. *International Journal of Public Opinion Research* 25: 383–395.
- Büttcher, S., C. L. A. Clarke, and G. V. Cormack. 2016. *Information Retrieval: Implementing and Evaluating Search Engines*. Cambridge, MA: MIT Press.
- Gaustad, T., and G. Bouma. 2002. Accurate stemming of Dutch for text classification. *Language and Computers* 45: 104–117.
- Guenther, N., and M. Schonlau. 2016. Support vector machines. *Stata Journal* 16: 917–937.
- Hastie, T., R. Tibshirani, and J. Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. New York: Springer.
- Hollink, V., J. Kamps, C. Monz, and M. de Rijke. 2004. Monolingual document retrieval for European languages. *Information Retrieval* 7: 33–52.

- Hull, D. A. 1996. Stemming algorithms: A case study for detailed evaluation. *Journal of the American Society of Information Science* 47: 70–84.
- Ignatow, G., and R. Mihalcea. 2016. *Text Mining: A Guidebook for the Social Sciences*. Thousand Oaks, CA: Sage.
- Jockers, M. L. 2014. *Text Analysis with R for Students of Literature*. Heidelberg: Springer.
- Kraaij, W., and R. Pohlmann. 1994. Porter's stemming algorithm for Dutch. In *Informatiewetenschap 1994: Wetenschappelijke bijdragen aan de derde STINFON Conferentie*, ed. N. L. G. M. and W. A. M. de Vroomen, 167–180. Tilburg, Netherlands.
- Madigan, D., A. Genkin, D. D. Lewis, S. Argamon, D. Fradkin, and L. Ye. 2005. Author identification on the large scale. In *Proceedings of the 2005 Meeting of the Classification Society of North America*. St. Louis, MO: Classification Society of North America.
- Manning, C. D., P. Raghavan, and H. Schütze. 2008. *Introduction to Information Retrieval*. Cambridge: Cambridge University Press.
- Manning, C. D., and H. Schütze. 1999. *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press.
- McCaffrey, D. F., and M. N. Elliott. 2008. Power of tests for a dichotomous independent variable measured with error. *Health Services Research* 43: 1085–1101.
- Paice, C. D. 1994. An evaluation method for stemming algorithms. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ed. W. B. Croft and C. J. van Rijsbergen, 42–50. New York: Springer.
- Porter, M. F. 1980. An algorithm for suffix stripping. *Program: Electronic library and information systems* 14: 130–137.
- Savoy, J. 2006. Light stemming approaches for the French, Portuguese, German and Hungarian languages. In *Proceedings of the 2006 ACM Symposium on Applied Computing*, 1031–1035. New York: ACM.
- Schonlau, M. 2005. Boosted regression (boosting): An introductory tutorial and a Stata plugin. *Stata Journal* 5: 330–354.
- . 2015. What do web survey panel respondents answer when asked “Do you have any other comment?”. *Survey Methods: Insights from the Field*. <http://surveyinsights.org/?p=6899>.
- Vapnik, V. N. 2000. *The Nature of Statistical Learning Theory*. 2nd ed. New York: Springer.
- Williams, U., and S. P. Williams. 2014. txttool: Utilities for text analysis in Stata. *Stata Journal* 14: 817–829.

About the authors

Matthias Schonlau, PhD, is a professor in the Department of Statistics and Actuarial Sciences at the University of Waterloo in Canada. His interests include survey methodology and text mining of open-ended questions.

Nick Guenther is an undergraduate in the Department of Statistics and the School of Computer Science at the University of Waterloo in Canada. His interests include feature learning, functional programming, and applying machine learning toward new technologies.

Ilia Sucholutsky is an undergraduate student in the Department of Statistics at the University of Waterloo in Canada. His interests include deep learning and discovering its impact on new problems.

Econometric convergence test and club clustering using Stata

Kerui Du
The Center for Economic Research
Shandong University
Jinan, China
kerrydu@sdu.edu.cn

Abstract. In this article, I introduce a new package with five commands to perform econometric convergence analysis and club clustering as proposed by Phillips and Sul (2007, *Econometrica* 75: 1771–1855). The `logtreg` command performs the log t regression test. The `pssecta` command implements the clustering algorithm to identify convergence clubs. The `scheckmerge` command conducts the log t regression test for all pairs of adjacent clubs. The `imergeclub` command tries to iteratively merge adjacent clubs. The `pfilter` command extracts the trend and cyclical components of a time series of each individual in panel data. I provide an example from Phillips and Sul (2009, *Journal of Applied Econometrics* 24: 1153–1185) to illustrate the use of these commands. Additionally, I use Monte Carlo simulations to exemplify the effectiveness of the clustering algorithm.

Keywords: `st0503`, `logtreg`, `pssecta`, `scheckmerge`, `imergeclub`, `pfilter`, convergence, club clustering, log t test

1 Introduction

Convergence in economics refers to the hypothesis that all economies would eventually converge in terms of per-capita output. This issue has played a central role in the empirical growth literature (Pesaran 2007). A large body of literature (for example, Baumol [1986]; Bernard and Durlauf [1995]; Barro and Sala-I-Martin [1997]; Lee, Pesaran, and Smith [1997]; Luginbuhl and Koopman [2004]) has contributed to developing methods for convergence tests and empirically investigating the convergence hypothesis across different countries and regions. In the past years, convergence analysis has also been applied in other topics such as cost of living (Phillips and Sul 2007), carbon dioxide emissions (Panopoulou and Pantelidis 2009), ecoefficiency (Camarero et al. 2013), house prices (Montañés and Olmos 2013), corporate tax (Regis, Cuestas, and Chen 2015), etc.

Phillips and Sul (2007) proposed a novel approach (termed “log t ” regression test)¹ to test the convergence hypothesis based on a nonlinear time-varying factor model. The proposed approach has the following merits: First, it accommodates heterogeneous agent behavior and evolution in that behavior. Second, the proposed test does not impose any particular assumptions concerning trend stationarity or stochastic nonstation-

1. It is also called “log t test” for short.

arity, thereby being robust to the stationarity property of the series. Phillips and Sul (2009) showed that the traditional convergence tests for economic growth have some pitfalls. For instance, estimation of augmented Solow regression under transitional heterogeneity is biased and inconsistent because of the issues of omitted variables and endogeneity. Conventional cointegration tests typically have low power to detect the asymptotic comovement, because the existence of a unit root in the differential of the series does not necessarily lead to the divergence conclusion.

Another common concern with the convergence analysis is the possible existence of convergence clubs. Regarding this issue, traditional studies typically divided all individuals into subgroups based on some prior information (for example, geographical location, institution), then tested the convergence hypothesis for each subgroup respectively. Phillips and Sul (2007) constructed a new algorithm to identify clusters of convergence subgroups. The developed algorithm is a data-driven method that avoids ex-ante sample separation. The relative transition parameter mechanism that Phillips and Sul (2007) proposed to characterize individual variations fits in with some common models.² It can be used as a general panel method to cluster individuals into groups with similar transition paths.

In practice, Phillips and Sul (2007, 2009) provided Gauss codes to perform their empirical studies. Recently, Schnurbus, Haupt, and Meier (2017) provided a set of R functions to replicate the key results of Phillips and Sul (2009). In this article, I introduce a new package, `psecta`, to perform the econometric convergence test and club clustering developed by Phillips and Sul (2007).

The remainder of this article is organized as follows: section 2 briefly describes the methodology of Phillips and Sul (2007); sections 3–7 explain the syntax and options of the new commands; section 8 performs Monte Carlo simulations to examine the effectiveness of the clustering algorithm; and section 9 illustrates the use of the commands with an example from Phillips and Sul (2009).

2 Econometric convergence test and club clustering

2.1 Time-varying factor representation

The starting point of the model is decomposing the panel data, X_{it} , as

$$X_{it} = g_{it} + a_{it} \quad (1)$$

where g_{it} represents systematic components such as permanent common components and a_{it} embodies transitory components. To separate common components from idiosyncratic components, we further transform (1) as

2. See, for example, economic growth with heterogeneous technological progress (Parente and Prescott 1994; Howitt and Mayer-Foulkes 2005), income processes with heterogeneity (Baker 1997; Moffitt and Gottschalk 2002), and stock price factor models (Menzly, Santos, and Veronesi 2002; Ludvigson and Ng 2007).

$$X_{it} = \left(\frac{g_{it} + a_{it}}{u_t} \right) u_t = \delta_{it} u_t \quad (2)$$

where δ_{it} is a time-varying idiosyncratic element and u_t is a single common component. Equation (2) is a dynamic-factor model where u_t captures some deterministic or stochastically trending behavior, and the time-varying factor-loading coefficient δ_{it} measures the idiosyncratic distance between X_{it} and the common trend component u_t .

In general, we cannot directly fit the model without imposing some restrictions on δ_{it} and u_t . Thus, [Phillips and Sul \(2007\)](#) proposed removing the common factor as follows:

$$h_{it} = \frac{X_{it}}{\frac{1}{N} \sum_{i=1}^N X_{it}} = \frac{\delta_{it}}{\frac{1}{N} \sum_{i=1}^N \delta_{it}} \quad (3)$$

h_{it} is the relative transition parameter, which measures the loading coefficient relative to the panel average at time t . In other words, h_{it} traces out a transition path of individual i in relation to the panel average. Equation (3) indicates that the cross-sectional mean of h_{it} is unity, and the cross-sectional variance of h_{it} satisfies the following condition:

$$H_{it} = \frac{1}{N} \sum_{i=1}^N (h_{it} - 1)^2 \rightarrow 0 \text{ if } \lim_{t \rightarrow \infty} \delta_{it} = \delta, \text{ for all } i$$

2.2 The log t regression test

The convergence of X_{it} requires the following condition:

$$\lim_{t \rightarrow \infty} \frac{X_{it}}{X_{jt}} = 1, \text{ for all } i \text{ and } j$$

[Phillips and Sul \(2007\)](#) defined this condition as the relative convergence. It is equivalent to the convergence of the time-varying factor-loading coefficient

$$\lim_{t \rightarrow \infty} \delta_{it} = \delta, \text{ for all } i$$

Assume the loading coefficient δ_{it} as

$$\delta_{it} = \delta_i + \sigma_{it} \xi_{it}, \quad \sigma_{it} = \frac{\sigma_i}{L(t)t^\alpha}, t \geq 1, \sigma_i > 0 \text{ for all } i$$

where $L(t)$ is a slowly varying function. Possible choices for $L(t)$ can be $\log(t)$, $\log^2(t)$, or $\log\{\log(t)\}$. The Monte Carlo simulations in [Phillips and Sul \(2007\)](#) indicate that $L(t) = \log(t)$ produces the least size distortion and the best test power. Thus, we set $L(t) = \log(t)$ in our Stata codes.

[Phillips and Sul \(2007\)](#) developed a regression t test for the null hypothesis of convergence,

$$\mathcal{H}_0: \delta_i = \delta \text{ and } \alpha \geq 0$$

against the alternative, $\mathcal{H}_A: \delta_i \neq \delta$ or $\alpha < 0$. Specifically, the hypothesis test can be implemented through the following log t regression model:

$$\log\left(\frac{H_1}{H_t}\right) - 2\log\{\log(t)\} = a + b\log(t) + \varepsilon_t$$

for $t = [rT], [rT] + 1, \dots, T$ with $r > 0$

The selection of the initiating sample fraction r might influence the results of the above regression. The Monte Carlo experiments indicate that $r \in [0.2, 0.3]$ achieves a satisfactory performance. Specifically, it is suggested to set $r = 0.3$ for the small or moderate $T (\leq 50)$ sample and set $r = 0.2$ for the large $T (\geq 100)$ sample.

Phillips and Sul (2007) further showed that $b = 2\alpha$ and \mathcal{H}_0 is conveniently tested through the weak inequality null $\alpha \geq 0$. It implies a one-sided t test. Under some technical assumptions, the limit distribution of the regression t statistic is

$$t_b = \frac{\hat{b} - b}{s_b} \Rightarrow N(0, 1)$$

where

$$s_b^2 = \widehat{\text{var}}(\hat{\varepsilon}_t) \left\{ \sum_{t=[rT]}^T \left(\log(t) - \frac{1}{T - [rT] + 1} \sum_{t=[rT]}^T \log(t) \right)^2 \right\}^{-1}$$

and $\widehat{\text{var}}(\hat{\varepsilon}_t)$ is a conventional heteroskedastic and autocorrelated estimate formed from the regression residuals.

2.3 Club convergence test and clustering

Rejection of the null hypothesis of convergence for the whole panel cannot rule out the existence of convergence in subgroups of the panel. To investigate the possibility of convergence clusters, Phillips and Sul (2007) developed a data-driven algorithm. Schnurbus, Haupt, and Meier (2017) advocated making some minor adjustments to the original algorithm. We sketch their ideas as follows:

1. Cross-section sorting

Sort individuals in the panel decreasingly according to their observations in the last period. If there is substantial time-series volatility in the data, the sorting can be implemented based on the time-series average of the last fraction (for example, $1/2, 1/3$) of the sample. Index individuals with their orders $\{1, \dots, N\}$.

2. Core group formation

- 2.1 Find the first k such that the test statistic of the log t regression $t_k > -1.65$ for the subgroup with individuals $\{k, k + 1\}$. If there is no k satisfying $t_k > -1.65$, exit the algorithm, and conclude that there are no convergence subgroups in the panel.
- 2.2 Start with the k identified in step 2.1, perform log t regression for the subgroups with individuals $\{k, k + 1, \dots, k + j\}, j \in \{1, \dots, N - k\}$. Choose j^* such that the subgroup with individuals $\{k, k + 1, \dots, k + j^*\}$ yields the highest value of the test statistic. Individuals $\{k, k + 1, \dots, k + j^*\}$ form a core group.

3. Sieve individuals for club membership

- 3.1 Form a complementary group $G_{j^*}^c$ with all the remaining individuals not included in the core group. Add one individual from $G_{j^*}^c$ at each time to the core group and run the log t test. Include the individual in the club candidate group if the test statistic is greater than the critical value c^* .³
- 3.2 Run the log t test for the club candidate group identified in step 3.1. If the test statistic \hat{t}_b is greater than -1.65 , the initial convergence club is obtained. If not, Phillips and Sul (2007) advocated raising the critical value c^* and repeating steps 3.1 and 3.2 until $\hat{t}_b > -1.65$. Schnurbus, Haupt, and Meier (2017) proposed adjusting this step as follows: If the convergence hypothesis does not hold for the club candidate group, sort the club candidates w.r.t. decreasing \hat{t}_b obtained in step 3.1. If there are some $\hat{t}_b > -1.65$, add the individual with the highest value of \hat{t}_b to form an extended core group. Add one individual from the remaining candidates at a time, run the log t test, and denote the test statistic \hat{t}_b . If the highest value of \hat{t}_b is not greater than -1.65 , stop the procedure; the extended core group will form an initial convergence club. Otherwise, repeat the above procedure to add the individual with the highest \hat{t}_b .

4. Recursion and stopping rule

Form a subgroup of the remaining individuals that are not sieved by step 3. Perform the log t test for this subgroup. If the test statistic is greater than -1.65 , the subgroup forms another convergence club. Otherwise, repeat steps 1–3 on this subgroup.

5. Club merging

Perform the log t test for all pairs of the subsequent initial clubs. Merge those clubs fulfilling the convergence hypothesis jointly. Schnurbus, Haupt, and Meier (2017) advocated conducting club merging iteratively as follows: run the log t test for the initial clubs 1 and 2; if they fulfill the convergence hypothesis jointly,

3. When T is small, the sieve criterion c^* can be set to 0 to ensure that it is highly conservative, whereas for large T , c^* can be set to the asymptotic 5% critical value -1.65 .

merge them to form the new club 1, then run the log t test for the new club 1 and the initial club 3 jointly; if not, run the log t test for initial clubs 2 and 3, etc. The new club classifications would be obtained by the above procedure. After that, one can also repeat the procedure on the newly obtained club classifications until no clubs can be merged, which leads to the classifications with the smallest number of convergence clubs.

3 The logtreg command

`logtreg` performs the log t test using linear regression with heteroskedasticity- and autocorrelation-consistent standard errors.

3.1 Syntax

```
logtreg varname [if] [in] [, kq(#) nomata]
```

3.2 Options

`kq(#)` specifies the first `kq()` proportion of the data to be discarded before regression. The default is `kq(0.3)`.

`nomata` bypasses the use of community-contributed Mata functions; by default, community-contributed Mata functions are used.

3.3 Stored results

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

Scalars

<code>e(N)</code>	number of individuals
<code>e(T)</code>	number of time periods
<code>e(nreg)</code>	number of observations used for the regression
<code>e(beta)</code>	log t coefficient
<code>e(tstat)</code>	t statistic for log t

Macros

<code>e(cmd)</code>	<code>logtreg</code>
<code>e(cmdline)</code>	command as typed
<code>e(varlist)</code>	name of the variable for log t test

Matrices

<code>e(res)</code>	table of estimation results
---------------------	-----------------------------

4 The `psecta` command

`psecta` implements club convergence and clustering analysis using the algorithm proposed by [Phillips and Sul \(2007\)](#).

4.1 Syntax

```
psecta varname [ , name(panelvar) kq(#) gen(newvar) cr(#) incr(#)
    maxcr(#) adjust fr(#) nomata noprtlogtreg ]
```

4.2 Options

`name(panelvar)` specifies a panel variable to be displayed for the clustering results; by default, the panel variable specified by `xtset` is used.

`kq(#)` specifies the first `kq()` proportion of the data to be discarded before regression. The default is `kq(0.3)`.

`gen(newvar)` creates a new variable to store club classifications. For the individuals that are not classified into any convergence club, missing values are generated.

`cr(#)` specifies the critical value for club clustering. The default is `cr(0)`.

`incr(#)` specifies the increment of `cr()` when the initial `cr()` value fails to sieve individuals for clusters. The default is `incr(0.05)`.

`maxcr(#)` specifies the maximum of `cr()` value. The default is `maxcr(50)`.

`adjust` specifies using the adjusted method proposed by [Schnurbus, Haupt, and Meier \(2017\)](#) instead of raising `cr()` when the initial `cr()` value fails to sieve individuals for clusters. See [Schnurbus, Haupt, and Meier \(2017\)](#) for more details.

`fr(#)` specifies sorting individuals by the time-series average of the last `fr()` proportion periods. The default is `fr(0)`, sorting individuals according to the last period.

`nomata` bypasses the use of community-contributed Mata functions; by default, community-contributed Mata functions are used.

`noprtlogtreg` suppresses the estimation results of the `logtreg`.

4.3 Stored results

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

Scalar	
<code>e(nclub)</code>	number of convergent clubs
Macros	
<code>e(cmd)</code>	<code>psecta</code>
<code>e(cmdline)</code>	command as typed
<code>e(varlist)</code>	name of the variable for log t test
Matrices	
<code>e(bm)</code>	log t coefficients
<code>e(tm)</code>	t statistics
<code>e(club)</code>	club classifications

4.4 Dependency of `psecta`

`psecta` depends on the Mata function `mm_which()`. If not already installed, you can install it by typing `ssc install moremata`.

5 The `scheckmerge` command

`scheckmerge` performs the log t test for all pairs of adjacent clubs.

5.1 Syntax

```
scheckmerge varname, club(varname) kq(#) [mdiv nomata]
```

5.2 Options

`club(varname)` specifies the initial club classifications. `club()` is required.

`kq(#)` specifies the first `kq()` proportion of the data to be discarded before regression. `kq()` is required.

`mdiv` specifies including the divergence group for the log t test; by default, the divergence group is excluded.

`nomata` bypasses the use of community-contributed Mata functions; by default, community-contributed Mata functions are used.

5.3 Stored results

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

Macros	
<code>e(cmd)</code>	<code>scheckmerge</code>
<code>e(cmdline)</code>	command as typed
<code>e(varlist)</code>	name of the variable for log t test
Matrices	
<code>e(bm)</code>	log t coefficients
<code>e(tm)</code>	t statistics

6 The `imergeclub` command

`imergeclub` iteratively conducts merging adjacent clubs.

6.1 Syntax

```
imergeclub varname, club(varname) kq(#) [name(panelvar) gen(newvar)
    imore mdiv nomata noprtlogtreg]
```

6.2 Options

`club(varname)` specifies the initial club classifications. `club()` is required.

`kq(#)` specifies the first `kq()` proportion of the data to be discarded before regression. `kq()` is required.

`name(panelvar)` specifies a panel variable to be displayed for the clustering results; by default, the panel variable specified by `xtset` is used.

`gen(newvar)` creates a new variable to store the new club classifications. For the individuals that are not classified into any convergence club, missing values are generated.

`imore` specifies merging clubs iteratively until no clubs can be merged. By default, the procedure is conducted as follows: First, run the log t test for the individuals belonging to the initial clubs 1 and 2 (obtained from club clustering). Second, if they fulfill the convergence hypothesis jointly, merge them to be the new club 1, and then run the log t test for the new club 1 and the initial club 3; if not, run the log t test for initial clubs 2 and 3, etc. If `imore` is chosen, the above procedure is repeated until no clubs can be merged.

`mdiv` specifies including the divergence group during club merging; by default, the divergence group is excluded.

`nomata` bypasses the use of community-contributed Mata functions; by default, community-contributed Mata functions are used.

`noprtlogtreg` suppresses the estimation results of the `logtreg`.

6.3 Stored results

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

Scalars	
<code>e(nclub)</code>	number of convergent clubs
Macros	
<code>e(cmd)</code>	<code>imergeclub</code>
<code>e(cmdline)</code>	command as typed
<code>e(varlist)</code>	name of the variable for log t test
Matrices	
<code>e(bm)</code>	log t coefficients
<code>e(tm)</code>	t statistics

7 The pfilter command

`pfilter` applies the `tsfilter` (see [TS] `tsfilter`) command into a panel-data context. It can be used to extract trend and cyclical components for each individual in the panel, respectively.

7.1 Syntax

```
pfilter varname, method(string) [trend(newvar) cyc(newvar) options]
```

7.2 Options

`method(string)` specifies the filter method. `string` should be chosen from `bk`, `bw`, `cf`, or `hp`. `method()` is required.

`trend(newvar)` creates a new variable for the trend component.

`cyc(newvar)` creates a new variable for the cyclical component.

`options` are any options available for `tsfilter` (see [TS] `tsfilter`).

7.3 Stored results

`pfilter` stores the following in `r()`:

Macros	
<code>r(cmd)</code>	<code>pfilter</code>
<code>r(cmdline)</code>	command as typed
<code>r(varlist)</code>	name of the variable

8 Monte Carlo simulation

Phillips and Sul (2007) performed extensive simulations for the power and size of the “log t ” test. They also showed how the clustering procedure works through Monte Carlo experiments. In this section, we further do simulation exercises to exemplify the effectiveness of the clustering algorithm in the finite sample.

Referring to Phillips and Sul (2007), we consider the following data-generating process,

$$\begin{aligned} X_{it} &= \theta_{it}\mu_t, & \theta_{it} &= \theta_i + \theta_{it}^0 \\ \theta_{it}^0 &= \rho_i\theta_{it-1}^0 + \varepsilon_{it} \end{aligned}$$

where $t = 1, \dots, T$; $\varepsilon_{it} \sim iid N\{0, \sigma_i^2 \log(t+1)^{-2} t^{-2\alpha_i}\}$, $\sigma_i \sim U[0.02, 0.28]$, $\alpha_i \sim U[0.2, 0.9]$; $\rho_i \sim U[0, 0.4]$. Note that the common component μ_t cancels out in the test procedure. It is not needed to generate the realizations of μ_t .

For simulations, we set $T = 20, 40, 60, 100$ and $N = 40, 80, 120$. The number of replications is 1,000. We first consider the following two cases:

Case 1: One convergence club and one divergence subgroup. We consider two equal-sized groups in the panel with numbers $N_1 = N_2 = N/2$. We set $\theta_i = 1$ and $\theta_i \sim U[1.5, 5]$ for the first and second groups, respectively. This implies that the first group forms a convergence club and the second group is divergent.

Case 2: Two convergence clubs. Two groups are set much like case 1, except that $\theta_i = 1$ and $\theta_i = 2$ for the first and second groups, respectively.

Tables 1 and 2 report the false discovery rate (the ratio of the replications that fail to identify the club memberships) for case 1 and case 2, respectively. Generally speaking, the false discovery rate is acceptable. In case 1, the false discovery rate is lower than 10% when $T = 20$, and it becomes lower than 5% when $T \geq 40$. In case 2, the false discovery rate is lower than 5% for all combinations of N and T except for ($N = 120, T = 40$).

Table 1. Simulation result of case 1

$T \backslash N$	40	80	120
20	0.097	0.070	0.082
40	0.048	0.035	0.047
60	0.023	0.032	0.038
100	0.027	0.033	0.039

Table 2. Simulation result of case 2

$T \backslash N$	40	80	120
20	0.026	0.023	0.046
40	0.014	0.031	0.056
60	0.012	0.022	0.040
100	0.015	0.015	0.042

For the experiments of case 1 and case 2, the values of θ_i are given. Here we provide another experiment in which the values of θ_i are unknown, and the data are generated by copying actual data with noises. The experiment is described as follows:

Case 3: We collect per-capita gross domestic product of the United States and Democratic Republic of the Congo and denote them as X_t^U and X_t^C , respectively.⁴ The simulation data are generated by $N/2$ copies of X_t^U and X_t^C with noises as follows:

$$X_{it}^j = X_t^j + \theta_{it}^0 X_t^j, \quad j = \{U, C\}$$

θ_{it}^0 is set as described above except that $\alpha_i = (0.1, 0.3, 0.6, 0.8)$.

The result is given in table 3, which shows that the false discovery rate is lower than 5% for all combinations of N and α_i except for $(N = 80, \alpha_i = 0.1)$. Taking the results presented in tables 1, 2, and 3 together, we can conclude that the clustering algorithm generally achieves a satisfactory performance.

Table 3. Simulation result of case 3

$\alpha_i \backslash N$	40	80	120
0.1	0.034	0.054	0.044
0.3	0.013	0.030	0.044
0.6	0.014	0.030	0.041
0.8	0.010	0.031	0.040

9 Example

The example provided here is a replication of the key results of Phillips and Sul (2009). They collected panel data covering 152 economies during the period of 1970–2003. They first examined whether the convergence hypothesis holds for the whole sample. Then, they investigated the possibility of club convergence using their proposed clustering algorithm. The replication is conducted as follows:

4. The period is 1950–2014, namely, $T = 65$.

```
. use ps2009
(PWT152, from Phillips and Sul (2009) in Journal of Applied Econometrics)
. egen id=group(country)
. xtset id year
      panel variable: id (strongly balanced)
      time variable: year, 1970 to 2003
                delta: 1 unit
. generate lnpgdp=ln(pgdp)
. pfilter lnpgdp, method(hp) trend(lnpgdp2) smooth(400)
```

First, the `pfilter` command is used to wipe out the cyclical component. A new variable, `lnpgdp2`, is generated to store the trend component. We then run the log t regression for the convergence test. The output reports the coefficient, standard error, and t statistic for $\log(t)$. Because the value of the t statistic (calculated as -159.55) is less than -1.65 , the null hypothesis of convergence is rejected at the 5% level.

```
. logtreg lnpgdp2, kq(0.333)
log t test:
```

Variable	Coeff	SE	T-stat
log(t)	-0.8748	0.0055	-159.5544

```
The number of individuals is 152.
The number of time periods is 34.
The first 11 periods are discarded before regression.
```

Furthermore, identifying convergence clubs is conducted by the `psecta` command. The output presents the club classifications. The `noprtlogtreg` option suppresses the estimation details. If not, the results of the log t regression would be displayed following each club. We put all the estimation results together in the matrix `result1` and display it. Additionally, a new variable `club` is generated to store the club classifications.

```
. psecta lnpgdp2, name(country) kq(0.333) gen(club) noprt
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx Club classifications xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
----- Club 1 :(50)-----
| Antigua | Australia | Austria | Belgium | Bermuda | Botswana | |
| Brunei | Canada | Cape Verde | Chile | China | Cyprus | Denmark |
| Dominica | Equatorial Guinea | Finland | France | Germany |
| Hong Kong | Iceland | Ireland | Israel | Italy | Japan |
| Korea, Republic of | Kuwait | Luxembourg | Macao | Malaysia |
| Maldives | Malta | Mauritius | Netherlands | New Zealand |
| Norway | Oman | Portugal | Puerto Rico | Qatar | Singapore |
| Spain | St. Kitts & Nevis | St.Vincent & Grenadines | Sweden |
| Switzerland | Taiwan | Thailand | United Arab Emirates |
| United Kingdom | United States |
-----
```

```

----- Club 2 :(30)-----
| Argentina | Bahamas | Bahrain | Barbados | Belize | Brazil |
| Colombia | Costa Rica | Dominican Republic | Egypt | Gabon |
| Greece | Grenada | Hungary | India | Indonesia | Mexico |
| Netherlands Antilles | Panama | Poland | Saudi Arabia |
| South Africa | Sri Lanka | St. Lucia | Swaziland | Tonga |
| Trinidad &Tobago | Tunisia | Turkey | Uruguay |
-----
----- Club 3 :(21)-----
| Algeria | Bhutan | Cuba | Ecuador | El Salvador | Fiji |
| Guatemala | Iran | Jamaica | Lesotho | Micronesia, Fed. Sts. |
| Morocco | Namibia | Pakistan | Papua New Guinea | Paraguay |
| Peru | Philippines | Romania | Suriname | Venezuela |
-----
----- Club 4 :(24)-----
| Benin | Bolivia | Burkina Faso | Cameroon | Cote d Ivoire | | |
| Ethiopia | Ghana | Guinea | Honduras | Jordan | Korea, Dem. Rep. |
| Laos | Mali | Mauritania | Mozambique | Nepal | Nicaragua | Samoa |
| Solomon Islands | Syria | Tanzania | Uganda | Vanuatu | Zimbabwe |
-----
----- Club 5 :(14)-----
| Cambodia | Chad | Comoros | Congo, Republic of | Gambia, The | |
| Iraq | Kenya | Kiribati | Malawi | Mongolia | Nigeria |
| Sao Tome and Principe | Senegal | Sudan |
-----
----- Club 6 :(11)-----
| Afghanistan | Burundi | Central African Republic | | |
| Guinea-Bissau | Madagascar | Niger | Rwanda | Sierra Leone |
| Somalia | Togo | Zambia |
-----
----- Club 7 :(2)-----
| Congo, Dem. Rep. | Liberia |
-----
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
. matrix b=e(bm)
. matrix t=e(tm)
. matrix result1=(b \ t)
. matlist result1, border(rows) rowtitle("log(t)") format(%9.3f) left(4)

```

log(t)	Club1	Club2	Club3	Club4	Club5
Coeff	0.382	0.240	0.110	0.131	0.190
T-stat	9.282	6.904	3.402	2.055	1.701

log(t)	Club6	Club7
Coeff	1.003	-0.470
T-stat	6.024	-0.559

Finally, we use the `scheckmerge` and `imergeclub` commands to perform possible club merging. We see that the initial clubs 4 and 5 can be merged to form a larger convergent club. The results obtained here are the same as those in [Phillips and Sul \(2009\)](#) and [Schnurbus, Haupt, and Meier \(2017\)](#).

```
. scheckmerge lnpgdp2, kq(0.333) club(club) mdiv
```

```
      The log t test for   Club 1+2
```

```
log t test:
```

Variable	Coeff	SE	T-stat
log(t)	-0.0507	0.0232	-2.1909

```
The number of individuals is 80.
```

```
The number of time periods is 34.
```

```
The first 11 periods are discarded before regression.
```

```
-----
```

```
      The log t test for   Club 2+3
```

```
log t test:
```

Variable	Coeff	SE	T-stat
log(t)	-0.1041	0.0159	-6.5339

```
The number of individuals is 51.
```

```
The number of time periods is 34.
```

```
The first 11 periods are discarded before regression.
```

```
-----
```

```
      The log t test for   Club 3+4
```

```
log t test:
```

Variable	Coeff	SE	T-stat
log(t)	-0.1920	0.0379	-5.0684

```
The number of individuals is 45.
```

```
The number of time periods is 34.
```

```
The first 11 periods are discarded before regression.
```

```
-----
```

```
      The log t test for   Club 4+5
```

```
log t test:
```

Variable	Coeff	SE	T-stat
log(t)	-0.0443	0.0696	-0.6360

```
The number of individuals is 38.
```

```
The number of time periods is 34.
```

```
The first 11 periods are discarded before regression.
```

```
-----
```

The log t test for Club 5+6

log t test:

Variable	Coeff	SE	T-stat
log(t)	-0.2397	0.0612	-3.9178

The number of individuals is 25.

The number of time periods is 34.

The first 11 periods are discarded before regression.

The log t test for Club 6+7

log t test:

Variable	Coeff	SE	T-stat
log(t)	-1.1163	0.0602	-18.5440

The number of individuals is 13.

The number of time periods is 34.

The first 11 periods are discarded before regression.

```
. matrix b=e(bm)
. matrix t=e(tm)
. matrix result2=(b \ t)
. matlist result2, border(rows) rowtitle("log(t)") format(%9.3f) left(4)
```

log(t)	Club1+2	Club2+3	Club3+4	Club4+5	Club5+6
Coeff	-0.051	-0.104	-0.192	-0.044	-0.240
T-stat	-2.191	-6.534	-5.068	-0.636	-3.918

log(t)	Club6+7
Coeff	-1.116
T-stat	-18.544

```
. imergeclub lnpgdp2, name(country) kq(0.333) club(club) gen(finalclub) noprt
xxxxxxxxxxxxxxxxxxxxxxxxxxxx Final Club classifications xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

```
----- Club 1 :(50)-----
| Antigua | Australia | Austria | Belgium | Bermuda | Botswana | |
| Brunei | Canada | Cape Verde | Chile | China | Cyprus | Denmark |
| Dominica | Equatorial Guinea | Finland | France | Germany |
| Hong Kong | Iceland | Ireland | Israel | Italy | Japan |
| Korea, Republic of | Kuwait | Luxembourg | Macao | Malaysia |
| Maldives | Malta | Mauritius | Netherlands | New Zealand |
| Norway | Oman | Portugal | Puerto Rico | Qatar | Singapore |
| Spain | St. Kitts & Nevis | St. Vincent & Grenadines | Sweden |
| Switzerland | Taiwan | Thailand | United Arab Emirates |
| United Kingdom | United States |
```

```

----- Club 2 :(30)-----
| Argentina | Bahamas | Bahrain | Barbados | Belize | Brazil |
| Colombia | Costa Rica | Dominican Republic | Egypt | Gabon |
| Greece | Grenada | Hungary | India | Indonesia | Mexico |
| Netherlands Antilles | Panama | Poland | Saudi Arabia |
| South Africa | Sri Lanka | St. Lucia | Swaziland | Tonga |
| Trinidad &Tobago | Tunisia | Turkey | Uruguay |
-----
----- Club 3 :(21)-----
| Algeria | Bhutan | Cuba | Ecuador | El Salvador | Fiji |
| Guatemala | Iran | Jamaica | Lesotho | Micronesia, Fed. Sts. |
| Morocco | Namibia | Pakistan | Papua New Guinea | Paraguay |
| Peru | Philippines | Romania | Suriname | Venezuela |
-----
----- Club 4 :(38)-----
| Benin | Bolivia | Burkina Faso | Cambodia | Cameroon | Chad | |
| Comoros | Congo, Republic of | Cote d Ivoire | Ethiopia |
| Gambia, The | Ghana | Guinea | Honduras | Iraq | Jordan | Kenya |
| Kiribati | Korea, Dem. Rep. | Laos | Malawi | Mali |
| Mauritania | Mongolia | Mozambique | Nepal | Nicaragua |
| Nigeria | Samoa | Sao Tome and Principe | Senegal |
| Solomon Islands | Sudan | Syria | Tanzania | Uganda | Vanuatu |
| Zimbabwe |
-----
----- Club 5 :(11)-----
| Afghanistan | Burundi | Central African Republic | | |
| Guinea-Bissau | Madagascar | Niger | Rwanda | Sierra Leone |
| Somalia | Togo | Zambia |
-----
----- Club 6 :(2)-----
| Congo, Dem. Rep. | Liberia |
-----
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
. matrix b=e(bm)
. matrix t=e(tm)
. matrix result3=(b \ t)
. matlist result3, border(rows) rowtitle("log(t)") format(%9.3f) left(4)

```

log(t)	Club1	Club2	Club3	Club4	Club5
Coeff	0.382	0.240	0.110	-0.044	1.003
T-stat	9.282	6.904	3.402	-0.636	6.024

log(t)	Club6
Coeff	-0.470
T-stat	-0.559

10 Acknowledgments

For their support, I would like to thank the National Nature Science Foundation of China (No. 71603148), the China Postdoctoral Science Foundation (No. 2016M590627;

No. 2017T100480), and the Shandong Provincial Natural Science Foundation, China (No. ZR2016GB10). The package introduced here benefits greatly from Donggyu Sul's Gauss codes. I would like to thank Donggyu Sul for making his codes available to us. I greatly appreciate H. Joseph Newton (the editor) and an anonymous referee for their helpful comments and suggestions that led to an improved version of this article and the package. I thank Min Deng for her linguistic assistance during the revision of this article.

11 References

- Baker, M. 1997. Growth-rate heterogeneity and the covariance structure of life-cycle earnings. *Journal of Labor Economics* 15: 338–375.
- Barro, R. J., and X. Sala-I-Martin. 1997. Technological diffusion, convergence, and growth. *Journal of Economic Growth* 2: 1–26.
- Baumol, W. J. 1986. Productivity growth, convergence, and welfare: What the long-run data show. *American Economic Review* 76: 1072–1085.
- Bernard, A. B., and S. N. Durlauf. 1995. Convergence in international output. *Journal of Applied Econometrics* 10: 97–108.
- Camarero, M., J. Castillo, A. J. Picazo-Tadeo, and C. Tamarit. 2013. Eco-efficiency and convergence in OECD countries. *Environmental and Resource Economics* 55: 87–106.
- Howitt, P., and D. Mayer-Foulkes. 2005. R&D, implementation, and stagnation: A Schumpeterian theory of convergence clubs. *Journal of Money, Credit and Banking* 37: 147–177.
- Lee, K., M. H. Pesaran, and R. Smith. 1997. Growth and convergence in a multi-country empirical stochastic Solow model. *Journal of Applied Econometrics* 12: 357–392.
- Ludvigson, S. C., and S. Ng. 2007. The empirical risk–return relation: A factor analysis approach. *Journal of Financial Economics* 83: 171–222.
- Luginbuhl, R., and S. J. Koopman. 2004. Convergence in European GDP series: A multivariate common converging trend–cycle decomposition. *Journal of Applied Econometrics* 19: 611–636.
- Menzly, L., T. Santos, and P. Veronesi. 2002. The time series of the cross section of asset prices. NBER Working Paper No. 9217, The National Bureau of Economic Research. <http://www.nber.org/papers/w9217>.
- Moffitt, R. A., and P. Gottschalk. 2002. Trends in the transitory variance of earnings in the United States. *Economic Journal* 112: C68–C73.
- Montañés, A., and L. Olmos. 2013. Convergence in US house prices. *Economics Letters* 121: 152–155.

- Panopoulou, E., and T. Pantelidis. 2009. Club convergence in carbon dioxide emissions. *Environmental and Resource Economics* 44: 47–70.
- Parente, S. L., and E. C. Prescott. 1994. Barriers to technology adoption and development. *Journal of Political Economy* 102: 298–321.
- Pesaran, M. H. 2007. A pair-wise approach to testing for output and growth convergence. *Journal of Econometrics* 138: 312–355.
- Phillips, P. C. B., and D. Sul. 2007. Transition modeling and econometric convergence tests. *Econometrica* 75: 1771–1855.
- . 2009. Economic transition and growth. *Journal of Applied Econometrics* 24: 1153–1185.
- Regis, P. J., J. C. Cuestas, and Y. Chen. 2015. Corporate tax in Europe: Towards convergence? *Economics Letters* 134: 9–12.
- Schnurbus, J., H. Haupt, and V. Meier. 2017. Economic transition and growth: A replication. *Journal of Applied Econometrics* 32: 1039–1042.

About the author

Kerui Du is an assistant professor at the Center for Economic Research, Shandong University. His primary research interests are applied econometrics, energy, and environmental economics.

Automatic portmanteau tests with applications to market risk management

Guangwei Zhu
Southwestern University of Finance and Economics
Chengdu, China
zhugw@swufe.edu.cn

Zaichao Du	Juan Carlos Escanciano
Southwestern University of Finance and Economics	Department of Economics
Chengdu, China	Indiana University
duzc@swufe.edu.cn	Bloomington, IN
	jescanci@indiana.edu

Abstract. In this article, we review some recent advances in testing for serial correlation, provide code for implementation, and illustrate this code’s application to market risk forecast evaluation. We focus on the classic and widely used portmanteau tests and their data-driven versions. These tests are simple to implement for two reasons: First, the researcher does not need to specify the order of the tested autocorrelations, because the test automatically chooses this number. Second, its asymptotic null distribution is chi-squared with one degree of freedom, so there is no need to use a bootstrap procedure to estimate the critical values. We illustrate the wide applicability of this methodology with applications to forecast evaluation for market risk measures such as value-at-risk and expected shortfall.

Keywords: st0504, dbptest, rtau, autocorrelation, consistency, power, Akaike’s information criterion, Schwarz’s Bayesian information criterion, market risk

1 Introduction

Testing for serial correlation has held a central role in time-series analysis since its inception (see the early contributions by [Yule \[1926\]](#) and [Quenouille \[1947\]](#)). Despite the many proposals and variations since the seminal contribution of [Box and Pierce \(1970\)](#), the so-called portmanteau tests are still the most widely used. In its simplest form, the employed statistic is just the sample size times the sum of the first p -squared sample autocorrelations, which is compared with critical values from a chi-squared distribution with p degrees of freedom (with a correction if the test is applied to residuals). The basic Box–Pierce statistic has been slightly modified to improve its finite sample performance; see [Davies, Triggs, and Newbold \(1977\)](#); [Ljung and Box \(1978\)](#); [Davies and Newbold \(1979\)](#); or [Li and McLeod \(1981\)](#). The properties of the classical Box–Pierce tests have been extensively studied in the literature; see, for example, the monograph by [Li \(2004\)](#) for a review of this literature. Much of the theoretical literature on Box–Pierce tests was developed under the independence assumption and hence is generally invalid when applied to dependent data (the asymptotic size of the test is different from the nominal

level); see [Newbold \(1980\)](#) or, more recently, [Francq, Roy, and Zakoïan \(2005\)](#) for valid tests. This limitation of classical Box–Pierce tests is by now well understood. In this article, we focus on a different limitation: the selection of the employed number of autocorrelations is arbitrary. We review the contribution of [Escanciano and Lobato \(2009\)](#), who proposed a data-driven portmanteau statistic where the number of correlations is not fixed but selected automatically from the data. In this article, we give a synthesis of this methodology, introduce new general assumptions for its validity, review new applications in risk management, and provide code for its implementation.

2 Automatic portmanteau tests: A synthesis

Given a strictly stationary process $\{Y_t\}_{t \in \mathbb{Z}}$ with $E(Y_t^2) < \infty$ and $\mu = E(Y_t)$, define the autocovariance of order j as

$$\gamma_j = \text{Cov}(Y_t, Y_{t-j}) = E\{(Y_t - \mu)(Y_{t-j} - \mu)\}, \quad \text{for all } j \geq 0$$

and the j th order autocorrelation as $\rho_j = \gamma_j/\gamma_0$. We aim to test the null hypothesis

$$H_0 : \rho_j = 0, \quad \text{for all } j \geq 1$$

against the fixed alternative hypotheses

$$H_1^K : \rho_j \neq 0, \quad \text{for some } 1 \leq j \leq K$$

and some $K \geq 1$.

Suppose we observe data $\{Y_t\}_{t=1}^n$. γ_j can then be consistently estimated by the sample autocovariance

$$\hat{\gamma}_j = \frac{1}{(n-j)} \sum_{t=1+j}^n (Y_t - \bar{Y})(Y_{t-j} - \bar{Y}), \quad j = 0, \dots, n-1$$

where \bar{Y} is the sample mean, and we can also introduce $\hat{\rho}_j = \hat{\gamma}_j/\hat{\gamma}_0$ to denote the j th order sample autocorrelation.

The Box–Pierce Q_p statistic ([Box and Pierce 1970](#)) is just

$$Q_p = n \sum_{j=1}^p \hat{\rho}_j^2$$

which is commonly implemented via the [Ljung and Box \(LB, 1978\)](#) modification

$$\text{LB}_p = n(n+2) \sum_{j=1}^p (n-j)^{-1} \hat{\rho}_j^2$$

When $\{Y_t\}_{t=1}^n$ are independent and identically distributed (i.i.d.), both Q_p and LB_p converge to a chi-squared distribution with p degrees of freedom, or χ_p^2 . When $\{Y_t\}_{t=1}^n$

are serially dependent, for example, when Y_t is a residual from a fitted model, the asymptotic distribution of Q_p or LB_p is generally different from χ_p^2 and depends on the data-generating process in a complicated way; see [Francq, Roy, and Zakoian \(2005\)](#) and [Delgado and Velasco \(2011\)](#).

In this section, we synthesize the AQ test methodology that was suggested in [Escanciano and Lobato \(2009\)](#) and extend the methodology to other situations. The main ingredients of the methodology are 1) the following asymptotic results for individual autocorrelations: for $j = 1, \dots, d$, where d is a fixed upper bound,

$$\sqrt{n}(\hat{\rho}_j - \rho_j) \xrightarrow{D} N(0, \tau_j) \quad (1)$$

for a positive asymptotic variance $\tau_j > 0$, with¹

$$\hat{\tau}_j \xrightarrow{P} \tau_j \quad (2)$$

and 2) a data-driven construction of p , given below. For i.i.d. observations, $\tau_j = 1$, and trivially, we can take $\hat{\tau}_j = 1$, but in other more general settings with weak dependence or estimation effects, we will have an unknown $\tau_j \neq 1$ that needs to be estimated. Our definitions of portmanteau tests allow for general cases. Define

$$Q_p^* = n \sum_{j=1}^p \tilde{\rho}_j^2$$

where $\tilde{\rho}_j = \hat{\rho}_j / \sqrt{\hat{\tau}_j}$ is called a “generalized autocorrelation” here. Then, the AQ test is given by

$$\text{AQ} = Q_p^* \quad (3)$$

where

$$\tilde{p} = \min\{p : 1 \leq p \leq d; L_p \geq L_h, h = 1, 2, \dots, d\}$$

with

$$L_p = Q_p^* - \pi(p, n, q)$$

$\pi(p, n, q)$ is a penalty term that takes the form

$$\pi(p, n, q) = \begin{cases} p \log n, & \text{if } \max_{1 \leq j \leq d} \sqrt{n} |\tilde{\rho}_j| \leq \sqrt{q \log n} \\ 2p, & \text{if } \max_{1 \leq j \leq d} \sqrt{n} |\tilde{\rho}_j| > \sqrt{q \log n} \end{cases} \quad (4)$$

and $q = 2.4$. The penalty term in (4) has been proposed by [Inglot and Ledwina \(2006b\)](#) for testing the goodness of fit for a distribution. The value of $q = 2.4$ is motivated from extensive simulation evidence in [Inglot and Ledwina \(2006a\)](#) and [Escanciano and Lobato \(2009\)](#). The value of $q = 0$ corresponds to the Akaike information criterion (AIC); see [Akaike \(1974\)](#). The value of $q = \infty$ corresponds to the Bayesian information criterion (BIC); see [Schwarz \(1978\)](#). In the context of testing for serial correlation, AIC

1. In this article, we use \xrightarrow{D} and \xrightarrow{P} to denote convergence in distribution and in probability, respectively.

is good at detecting nonzero correlations at long lags but leads to size distortions. In contrast, BIC controls the size accurately and is good for detecting nonzero correlations at short lags. As shown empirically in figures 1 and 2 in Escanciano and Lobato (2009), the choice of $q = 2.4$ provides a “switching effect” in which one combines the advantages of AIC and BIC. Thus, we recommend $q = 2.4$ in applications. The upper bound d does not affect the asymptotic null distribution of the test, although it may have an impact on power if it is chosen too small. The finite sample performance of the automatic tests is not sensitive to the choice of d for moderate and large values of this parameter, as shown in table 5 of Escanciano and Lobato (2009) and table 6 of Escanciano, Lobato, and Zhu (2013). Extensive simulation experience suggests that the choice of d that is equal to the closest integer around \sqrt{n} performs well in practice.

Theorem 1 *Under the null hypothesis, (1) and (2), $AQ \xrightarrow{D} \chi_1^2$.*

This theorem justifies the rejection region

$$AQ > \chi_{1,1-\alpha}^2$$

where $\chi_{1,1-\alpha}^2$ is the $(1 - \alpha)$ quantile of the χ_1^2 . The following theorem shows the consistency of the test.

Theorem 2 *Assume $\hat{\rho}_j \xrightarrow{P} \rho_j$ for $j = 1, \dots, d$, and let (2) hold. Then, the test based on AQ is consistent against H_1^K , for $K \leq d$.*

Note that joint convergence of the vector of autocorrelations is unnecessary, in contrast to much of the literature. Thus, the methodology of this article does not require estimation of large dimensional asymptotic variances.

The proofs of both theorems follow from straightforward modification of those in Escanciano and Lobato (2009) and are hence omitted.

Remark 1. The methodology can be applied to any setting where (1) and (2) can be established. This includes raw data or residuals from any model. There is an extensive literature proving conditions such as (1) and (2) under different assumptions; see examples below.

Remark 2. The reason for the χ_1^2 limiting distribution of the AQ test is that under the null hypothesis, $\lim_{n \rightarrow \infty} P(\tilde{p} = 1) = 1$. Heuristically, under the null hypothesis, Q_p^* is small, and $\pi(p, n, q)$ increases in p , so the optimal choice selected is the lowest dimensionality $p = 1$ with high probability.

3 Applications to risk management

We illustrate the general applicability of the methodology with new applications in risk management. There is a very extensive literature on the quantification of market

risk for derivative pricing, for portfolio choice and for risk management purposes. This literature has long been particularly interested in evaluating market risk forecasts, or the so-called backtests; see [Jorion \(2007\)](#) and [Christoffersen \(2009\)](#) for comprehensive reviews. A leading market risk measure has been the value at risk (VAR), and more recently, expected shortfall (ES). VAR summarizes the worst loss over a target horizon that will not be exceeded at a given level of confidence called “coverage level”. ES is the expected value of losses beyond a given level of confidence.² We review popular backtests for VAR and ES and derive automatic versions using the general methodology above.

Let R_t denote the revenue of a bank at time t , and let Ω_{t-1} denote the risk manager’s information at time $t-1$, which contains lagged values of R_t and possibly lagged values of other variables, say, X_t . That is, $\Omega_{t-1} = \{X_{t-1}, X_{t-2}, \dots; R_{t-1}, R_{t-2}, \dots\}$. Let $G(\cdot, \Omega_{t-1})$ denote the conditional cumulative distribution function of R_t given Ω_{t-1} , that is, $G(\cdot, \Omega_{t-1}) = \Pr(R_t \leq \cdot | \Omega_{t-1})$. Assume $G(\cdot, \Omega_{t-1})$ is continuous. Let $\alpha \in [0, 1]$ denote the coverage level. The α -level VAR is defined as the quantity $\text{VAR}_t(\alpha)$ such that

$$\Pr\{R_t \leq -\text{VAR}_t(\alpha) | \Omega_{t-1}\} = \alpha \quad (5)$$

That is, the $-\text{VAR}_t(\alpha)$ is the α th percentile of the conditional distribution G ,

$$\text{VAR}_t(\alpha) = -G^{-1}(\alpha, \Omega_{t-1}) = -\inf\{y : G(y, \Omega_{t-1}) \geq \alpha\}$$

Define the α -violation or hit at time t as

$$h_t(\alpha) = 1\{R_t \leq -\text{VAR}_t(\alpha)\}$$

where $1(\cdot)$ denotes the indicator function. That is, the violation takes the value 1 if the loss at time t is larger than or equal to $\text{VAR}_t(\alpha)$, and it takes the value 0 otherwise. Equation (5) implies that violations are Bernoulli variables with mean α and, moreover, that centered violations are a martingale difference sequence (MDS) for each $\alpha \in [0, 1]$; that is,

$$E\{h_t(\alpha) - \alpha | \Omega_{t-1}\} = 0 \text{ for each } \alpha \in [0, 1]$$

This restriction has been the basis for the extensive literature on backtesting VAR. Two of its main implications, the zero mean property of the hit sequence $\{h_t(\alpha) - \alpha\}_{t=1}^{\infty}$ and its uncorrelation, led to the unconditional and conditional backtests of [Kupiec \(1995\)](#) and [Christoffersen \(1998\)](#), respectively, which are the most widely used backtests. More recently, [Berkowitz, Christoffersen, and Pelletier \(2011\)](#) have proposed the Box–Pierce-type test for VAR,

$$C_{\text{VAR}}(p) = n \sum_{j=1}^p \hat{\rho}_j^2$$

with $\hat{\rho}_j = \hat{\gamma}_j / \hat{\gamma}_0$ and $\hat{\gamma}_j = 1/(n-j) \sum_{t=1+j}^n \{\hat{h}_t(\alpha) - \alpha\} \{\hat{h}_{t-j}(\alpha) - \alpha\}$, and where $\{\hat{h}_t(\alpha) = 1\{R_t \leq -\widehat{\text{VAR}}_t(\alpha)\}\}_{t=1}^n$, for an estimator of the VAR, $\widehat{\text{VAR}}_t(\alpha)$. An automatic version of the test statistic in [Berkowitz, Christoffersen, and Pelletier \(2011\)](#) can be

2. Other names for ES are conditional VAR, average VAR, tail VAR, or expected tail loss.

computed following the algorithm above with $\tau_j = 1$. This test is valid only when there are no estimation effects. If T is the in-sample size for estimation and n is the out-of-sample size used for forecast evaluation, the precise condition for no estimation effects in backtesting VAR and ES is that both $T \rightarrow \infty$ and $n \rightarrow \infty$ at a rate such that $n/T \rightarrow 0$ (that is, the in-sample size is much larger than the out-of-sample size). More generally, Escanciano and Olmo (2010) provided primitive conditions for the convergences (1) and (2) to hold in a general setting where there are estimating effects from estimating VAR. When estimation effects are present, τ_j no longer equals 1, but Escanciano and Olmo (2010) provide suitable estimators, $\hat{\tau}_j$, satisfying (2). Let AC_{VAR} denote the AQ version of $C_{\text{VAR}}(p)$.

More recently, there has been a move in the banking sector toward ES as a suitable measure of market risk able to capture “tail risk” (the risk coming from very big losses). ES is defined as the conditional expected loss given that the loss is larger than $\text{VAR}_t(\alpha)$, that is,

$$\text{ES}_t(\alpha) = E\{-R_t | \Omega_{t-1}, -R_t > \text{VAR}_t(\alpha)\}$$

Definition of a conditional probability and a change of variables yield a useful representation of $\text{ES}_t(\alpha)$ in terms of $\text{VAR}_t(\alpha)$,

$$\text{ES}_t(\alpha) = \frac{1}{\alpha} \int_0^\alpha \text{VAR}_t(u) du \quad (6)$$

Unlike $\text{VAR}_t(\alpha)$, which contains information only on one quantile level α , $\text{ES}_t(\alpha)$ contains information from the whole left tail by integrating all VARs from 0 to α . As we did with (6), we define the cumulative violation process,

$$H_t(\alpha) = \frac{1}{\alpha} \int_0^\alpha h_t(u) du$$

Because $h_t(u)$ has mean u , then by Fubini’s theorem, $H_t(\alpha)$ has mean $1/\alpha \int_0^\alpha u du = \alpha/2$. Moreover, again by Fubini’s theorem, the MDS property of the class $\{h_t(\alpha) - \alpha : \alpha \in [0, 1]\}_{t=1}^\infty$ is preserved by integration, which means that $\{H_t(\alpha) - \alpha/2\}_{t=1}^\infty$ is also an MDS. For computational purposes, it is convenient to define $u_t = G(R_t, \Omega_{t-1})$. Because $h_t(u) = 1\{R_t \leq -\text{VAR}_t(u)\} = 1(u_t \leq u)$, we obtain

$$\begin{aligned} H_t(\alpha) &= \frac{1}{\alpha} \int_0^\alpha 1(u_t \leq u) du \\ &= \frac{1}{\alpha} (\alpha - u_t) 1(u_t \leq \alpha) \end{aligned}$$

Like violations, cumulative violations are distribution free because $\{u_t\}_{t=1}^\infty$ comprises a sample of i.i.d. $U[0, 1]$ variables (see Rosenblatt [1952]). Cumulative violations have been recently introduced in Du and Escanciano (2017). The variables $\{u_t\}_{t=1}^\infty$ necessary to construct $\{H_t(\alpha)\}_{t=1}^\infty$ are generally unknown because the distribution of the data G

is unknown. In practice, researchers and risk managers specify a parametric conditional distribution $G(\cdot, \Omega_{t-1}, \theta_0)$, where θ_0 is some unknown parameter in $\Theta \subset \mathbb{R}^p$, and proceed to estimate θ_0 before producing VAR and ES forecasts. Popular choices for distributions $G(\cdot, \Omega_{t-1}, \theta_0)$ are those derived from location-scale models with Student's t distributions, but other choices can be certainly entertained in our setting. With the parametric model at hand, we can define the “generalized errors”

$$u_t(\theta_0) = G(R_t, \Omega_{t-1}, \theta_0)$$

and the associated cumulative violations

$$H_t(\alpha, \theta_0) = \frac{1}{\alpha} \{\alpha - u_t(\theta_0)\} 1(u_t(\theta_0) \leq \alpha)$$

As with VARs, the arguments above provide a theoretical justification for backtesting ES by checking whether $\{H_t(\alpha, \theta_0) - \alpha/2\}_{t=1}^{\infty}$ have zero mean (unconditional ES backtest) and whether $\{H_t(\alpha, \theta_0) - \alpha/2\}_{t=1}^{\infty}$ are uncorrelated (conditional ES backtest).

Let $\hat{\theta}$ be an estimator of θ_0 and construct residuals

$$\hat{u}_t = G(R_t, \Omega_{t-1}, \hat{\theta})$$

and estimated cumulative violations

$$\hat{H}_t(\alpha) = \frac{1}{\alpha} (\alpha - \hat{u}_t) 1(\hat{u}_t \leq \alpha)$$

Then, we obtain

$$\hat{\gamma}_j = \frac{1}{n-j} \sum_{t=1+j}^n \left\{ \hat{H}_t(\alpha) - \alpha/2 \right\} \left\{ \hat{H}_{t-j}(\alpha) - \alpha/2 \right\} \text{ and } \hat{\rho}_j = \frac{\hat{\gamma}_j}{\hat{\gamma}_0}$$

Du and Escanciano (2017) construct the Box–Pierce test statistic

$$C_{\text{ES}}(p) = n \sum_{j=1}^p \hat{\rho}_j^2$$

and derive its asymptotic null distribution. In particular, they establish conditions for (1) and (2) to hold and provide expressions for the corresponding $\hat{\tau}_j$. Let AC_{ES} denote the AQ version of $C_{\text{ES}}(p)$.

Compared with the existing backtests, these automatic backtests select p from the data and require only estimation of marginal asymptotic variances of marginal correlations to obtain known limiting distributions.

4 Implementation

We introduce the `dbptest` command to implement the AQ test (3). Notice that $\tau_j = 1$ for i.i.d. observations and for backtesting VAR and ES without estimation effects.

We also provide the `rtau` command to estimate τ_j for more general cases, including MDS as in Escanciano and Lobato (2009), as well as backtests for VAR and ES with estimation effects as in Escanciano and Olmo (2010) and Du and Escanciano (2017), respectively.

4.1 Syntax

Automatic Q test

```
dbptest varname [if] [in] [, mu(#) q(#) tauvector(matname) nlags(#)]
```

Estimating τ_j

```
rtau varname [if] [in], nlags(#) seriestype(type) [cl(#) nobs(#)]
```

4.2 Options

Automatic Q test

`mu(#)` specifies the mean of the tested variable. The default is the variable's sample mean.

`q(#)` is a fixed positive number to control the switching effect between the AIC and BIC. The default is `q(2.4)`.

`tauvector(matname)` specifies a column vector containing variances of the autocorrelations. The default is a vector of 1s.

`nlags(#)` specifies the maximum number of lags of autocorrelations. The default is the closest integer around \sqrt{n} , where n is the number of observations. If it is larger than the dimension of `tauvector()`, it will be replaced by the dimension of `tauvector()`.

Estimating τ_j

`nlags(#)` specifies the number of lags of autocorrelations. `nlags()` is required.

`seriestype(type)` specifies one of the following types: `mds`, `var`, or `es`. `seriestype()` is required.

`seriestype(mds)` specifies *varname* to be an MDS as in Escanciano and Lobato (2009).

`seriestype(var)` corresponds to backtesting VAR. *varname* assumes a first-order autoregressive mean model and a conditional variance model with squared residuals with lags of order 1 and variance components with lags of order 1 [AR(1)–GARCH(1,1)] model with Student's t innovations when deriving the estimation effects.

`seriestype(es)` corresponds to backtesting ES, and `varname` assumes an AR(1)–GARCH(1,1) model with Student’s t innovations when deriving the estimation effects.

`c1(#)` specifies the coverage level of VAR and ES. The default is `c1(0.05)`.

`nobs(#)` specifies the in-sample size when backtesting VAR and ES.

4.3 Remarks

One needs to `tsset` the data before using `dbptest` and `rtau`.

Automatic Q test

`dbptest` implements a data-driven Box–Pierce test for serial correlations. The test automatically chooses the order of autocorrelations. The command reports not only the usual outputs of the Box–Pierce test as `wntestq`, that is, the Q statistics and the corresponding p -value, but also the automatic number of lags chosen.

Estimating τ_j

`rtau` estimates the asymptotic variances of autocorrelations when necessary. This includes

1. MDS data; and
2. backtesting ES and VAR with estimation effects.

`c1(#)` and `nobs(#)` are required only when `seriestype(var)` or `seriestype(es)` is specified.

4.4 Stored results

Automatic Q test

`dbptest` stores the following in `r()`:

Scalars			
<code>r(stat)</code>	Q statistic	<code>r(lag)</code>	the number of lags
<code>r(p)</code>	probability value		

Estimating τ_j

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

Matrix	
<code>e(tau)</code>	variances of autocorrelations

4.5 Example

To illustrate the use of the two commands, we consider the DAX Index return data from January 1, 1997, to June 30, 2009 as in [Du and Escanciano \(2017\)](#). The in-sample period is from January 1, 1997, to June 30, 2007. The out-of-sample period is from July 1, 2007, to June 30, 2009, which is the financial crisis period.

We use the in-sample data to fit an AR(1)–GARCH(1,1) model with Student's t innovations. After getting the estimates for u_t , $h_t(\alpha)$, and $H_t(\alpha)$ using the out-of-sample data, we implement the conditional backtests for VAR and ES using the new `dbptest` command.

Without estimation effects

Here we carry out the AQ test (3) without considering the estimation effects, that is, $\tau_j = 1$.

```
. set matsize 509
. import delimited "dax.csv", varnames(1)
(3 vars, 3,168 obs)
. scalar nin = 2658
. scalar nout= 509
. scalar ninout = nin + nout
. keep lret date
. drop in 1
(1 observation deleted)
. generate sin = (_n <= nin)
. generate sout= (_n > nin & _n<=ninout)
. keep if _n<=ninout
(0 observations deleted)
. generate date_num=_n
. tsset date_num
      time variable:  date_num, 1 to 3167
      delta: 1 unit
. arch lret if sin==1, noconstant arch(1) garch(1) ar(1) distribution(t)
(output omitted)
. matrix define awab=e(b)
. matrix define covm=e(V)
. scalar ahat  = awab[1,1]
. scalar alphasat = awab[1,2]
. scalar bethat = awab[1,3]
. scalar omghat = awab[1,4]
. scalar vhat = round(e(tdf))
. predict resin, residuals
. generate resout = resin if sin==0
(2,658 missing values generated)
. replace resin=. if sin==0
(509 real changes made, 509 to missing)
```

```

. predict convar, variance
. generate fith = convar if sin==0
(2,658 missing values generated)
. replace convar=. if sin==0
(509 real changes made, 509 to missing)
. generate fitsig = sqrt(fith)
(2,658 missing values generated)
. generate epsj = resout/fitsig
(2,658 missing values generated)
. generate utj = t(vhat, epsj*sqrt(vhat/(vhat-2)))
(2,658 missing values generated)
. scalar jalp = 0.1
. generate h = (utj <= jalp) if sout==1
(2,658 missing values generated)
. generate utalp = utj - jalp if h == 1
(3,108 missing values generated)
. replace utalp=0 if utalp==. & sout==1
(450 real changes made)
. generate H =-utalp/jalp
(2,658 missing values generated)
. dbptest H, mu(0.05)
Automatic Portmanteau test for serial correlation
-----
Variable: H
-----
Portmanteau (Q) statistic           =    2.8417
Prob > chi2(1)                     =    0.0918
The number of lag(s) (from 1 to 23) =         1
-----

```

The displayed results are for cumulative violations at a 10% coverage level, that is, $H_t(0.1)$. Under the correct model specification, we have $E\{H_t(\alpha)\} = \alpha/2$, so we set `mu()` to be 0.05. We get an AQ statistic of 2.8417 and a p -value of 0.0918. Hence, the ES model is rejected at a 10% significance level. It also reports the number of lags chosen, which is 1 in this case.

Likewise, we carry out the conditional backtest for VAR using $h_t(\alpha)$. Following the rule of thumb that the coverage level for ES is twice (or approximately twice) that of VAR, we examine the autocorrelations of $h_t(0.05)$.

```
. capture drop h
. scalar jalp=0.05
. generate h = (utj <= jalp) if sout==1
(2,658 missing values generated)
. dbptest h, mu(0.05)
Automatic Portmanteau test for serial correlation
```

Variable: h

Portmanteau (Q) statistic	=	0.7972
Prob > chi2(1)	=	0.3719
The number of lag(s) (from 1 to 23)	=	1

We now get an AQ statistic of 0.7972 and a p -value of 0.3719, so we fail to reject the VAR model.

With estimation effects

To account for the estimation effects, we use the `rtau` command to estimate τ_j before we run the `dbptest` command.

```
. rtau lret, nlags(15) seriestype(es) cl(0.1) nobs(2658)
```

Asymptotic Variances of Autocorrelations

Order	Tau for ES
1	1.0027636
2	1.0192228
3	1.0192343
4	1.004399
5	1.0030891
6	1.0021455
7	1.0137747
8	1.0016341
9	1.0094143
10	1.0012676
11	1.0011319
12	1.0080588
13	1.0077699
14	1.0033674
15	1.0017961

```
. matrix Tau_ES = e(tau)
. dbptest H, mu(0.05) tauvector(Tau_ES)
Automatic Portmanteau test for serial correlation
```

Variable: H

Portmanteau (Q) statistic	=	2.8338
Prob > chi2(1)	=	0.0923
The number of lag(s) (from 1 to 15)	=	1

```
. rtau lret, nlags(15) seriestype(var) cl(0.05) nobs(2658)
```

Asymptotic Variances of Autocorrelations

Order	Tau for VaR
1	1.01014
2	1.0029985
3	1.0023986
4	1.0023737
5	1.0027832
6	1.0021056
7	1.001556
8	1.0014201
9	1.0011457
10	1.0009844
11	1.001431
12	1.0013224
13	1.0013889
14	1.0009824
15	1.0011676

```
. matrix Tau_VaR = e(tau)
. dbptest h, mu(0.05) tauvector(Tau_VaR)
Automatic Portmanteau test for serial correlation
```

Variable: h

Portmanteau (Q) statistic	=	0.7892
Prob > chi2(1)	=	0.3743
The number of lag(s) (from 1 to 15)	=	1

Notice that the in-sample size here is 2,658. The AQ test statistics for ES and VAR here are slightly lower than those without estimation effects. The test conclusions remain the same, although the p -values are slightly higher than before.

5 References

- Akaike, H. 1974. A new look at the statistical model identification. *IEEE Transactions on Automatic Control* 19: 716–723.
- Berkowitz, J., P. Christoffersen, and D. Pelletier. 2011. Evaluating value-at-risk models with desk-level data. *Management Science* 57: 2213–2227.
- Box, G. E. P., and D. A. Pierce. 1970. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American Statistical Association* 65: 1509–1526.
- Christoffersen, P. 2009. Value-at-risk models. In *Handbook of Financial Time Series*, ed. T. G. Andersen, R. A. Davis, J.-P. Kreiß, and T. Mikosch, 753–766. Berlin: Springer.
- Christoffersen, P. F. 1998. Evaluating interval forecasts. *International Economic Review* 39: 841–862.

- Davies, N., and P. Newbold. 1979. Some power studies of a portmanteau test of time series model specification. *Biometrika* 66: 153–155.
- Davies, N., C. M. Triggs, and P. Newbold. 1977. Significance levels of the Box-Pierce portmanteau statistic in finite samples. *Biometrika* 64: 517–522.
- Delgado, M. A., and C. Velasco. 2011. An asymptotically pivotal transform of the residuals sample autocorrelations with application to model checking. *Journal of the American Statistical Association* 106: 946–958.
- Du, Z., and J. C. Escanciano. 2017. Backtesting expected shortfall: Accounting for tail risk. *Management Science* 63: 940–958.
- Escanciano, J. C., and I. N. Lobato. 2009. An automatic Portmanteau test for serial correlation. *Journal of Econometrics* 151: 140–149.
- Escanciano, J. C., I. N. Lobato, and L. Zhu. 2013. Automatic specification testing for vector autoregressions and multivariate nonlinear time series models. *Journal of Business & Economic Statistics* 31: 426–437.
- Escanciano, J. C., and J. Olmo. 2010. Backtesting parametric value-at-risk with estimation risk. *Journal of Business & Economic Statistics* 28: 36–51.
- Francq, C., R. Roy, and J.-M. Zakoïan. 2005. Diagnostic checking in ARMA models with uncorrelated errors. *Journal of the American Statistical Association* 100: 532–544.
- Inglot, T., and T. Ledwina. 2006a. Data driven score tests of fit for semiparametric homoscedastic linear regression model. Technical Report 665, Institute of Mathematics Polish Academy of Sciences.
- . 2006b. Towards data driven selection of a penalty function for data driven Neyman tests. *Linear Algebra and its Applications* 417: 124–133.
- Jorion, P. 2007. *Value at Risk: The New Benchmark for Managing Financial Risk*. 3rd ed. New York: McGraw-Hill.
- Kupiec, P. H. 1995. Techniques for verifying the accuracy of risk measurement models. *Journal of Derivatives* 3: 73–84.
- Li, W. K. 2004. *Diagnostic Checks in Time Series*. Boca Raton, FL: Chapman & Hall/CRC.
- Li, W. K., and A. I. McLeod. 1981. Distribution of the residual autocorrelations in multivariate ARMA time series models. *Journal of the Royal Statistical Society, Series B* 43: 231–239.
- Ljung, G. M., and G. E. P. Box. 1978. On a measure of lack of fit in time series models. *Biometrika* 65: 297–303.
- Newbold, P. 1980. The equivalence of two tests of time series model adequacy. *Biometrika* 67: 463–465.

Quenouille, M. H. 1947. A large-sample test for the goodness of fit of autoregressive schemes. *Journal of the Royal Statistical Society* 110: 123–129.

Rosenblatt, M. 1952. Remarks on a multivariate transformation. *Annals of Mathematical Statistics* 23: 470–472.

Schwarz, G. 1978. Estimating the dimension of a model. *Annals of Statistics* 6: 461–464.

Yule, G. U. 1926. Why do we sometimes get nonsense-correlations between time-series? A study in sampling and the nature of time-series. *Journal of the Royal Statistical Society* 89: 1–63.

About the authors

Guangwei Zhu is a lecturer in the Institute of Chinese Financial Studies at the Southwestern University of Finance and Economics, Chengdu, China. His research is funded by the Excellent Doctoral Dissertation Foundation of Southwestern University of Finance and Economics.

Zaichao Du is a professor in the Research Institute of Economics and Management at the Southwestern University of Finance and Economics, Chengdu, China. His research is funded by the National Natural Science Foundation of China, 71401140.

Juan Carlos Escanciano is a professor in the Department of Economics at Indiana University, Bloomington, IN. His research is funded by the Spanish Plan Nacional de I+D+I, reference number ECO2014-55858-P.

Two-stage residual inclusion estimation: A practitioners guide to Stata implementation

Joseph V. Terza
Department of Economics
Indiana University–Purdue University Indianapolis
Indianapolis, IN
jvterza@iupui.edu

Abstract. Empirical econometric research often requires implementation of nonlinear models whose regressors include one or more endogenous variables—regressors that are correlated with the unobserved random component of the model. In such cases, conventional regression methods that ignore endogeneity will likely produce biased results that are not causally interpretable. Terza, Basu, and Rathouz (2008, *Journal of Health Economics* 27: 531–543) discuss a relatively simple estimation method (two-stage residual inclusion) that avoids endogeneity bias, is applicable in many nonlinear regression contexts, and can easily be implemented in Stata. In this article, I offer a step-by-step protocol to implement the two-stage residual inclusion method in Stata. I illustrate this protocol in the context of a real-data example. I also discuss other examples and pertinent Stata code.

Keywords: st0505, two-stage residual inclusion, endogeneity

1 Introduction

My objective is to develop a simple but consistent estimation protocol in Stata for the parameters of a generic nonlinear regression model with dependent variable Y , which has a vector of independent variables that includes X_u , an unobservable regressor; X_o , a vector of observable regressors that are not correlated with X_u ; and X_e , an observable regressor that is correlated with X_u —that is, X_e is endogenous.¹ The endogeneity of X_e (that is, the correlation between X_e and X_u) confounds the identification and estimation of the possible causal effect of X_e (or any of the other regressors in the model for that matter) on Y . If, for instance, the presence of X_u is ignored, and a conventional regression method is applied, then the corresponding estimate of the effect of X_e will likely be biased, because it will reflect influence that should instead have been attributed to the unobservables. The general modeling and estimation framework discussed by Terza, Basu, and Rathouz (2008) is designed to control for endogeneity, thereby eliminating consequent bias. Their generic model consists of a regression equation with a dependent variable that is the outcome of interest (the outcome equation) and an auxiliary equation that formalizes the correlation between X_e and X_u . The outcome and auxiliary (O&A) equations can each be defined based on either a mini-

1. X_e and X_u may be made up of more than one regressor. We portray them as being single regressors here to simplify exposition.

mally parametric (MP) or a fully parametric (FP) regression structure. Formally, one can specify the outcome component of the model as either

$$Y = \mu(X_e, X_o, X_u; \beta) + e \quad (\text{MP specification}) \tag{1}$$

or

$$f(Y|X_e, X_o, X_u; \beta) \quad (\text{FP specification}) \tag{2}$$

where $\mu(X_e, X_o, X_u; \beta)$ denotes the conditional mean of Y given X_e , X_o , and X_u ; β is a vector of parameters; and $f(Y|X_e, X_o, X_u; \beta)$ is the conditional probability density function of Y given X_e , X_o , and X_u . Similarly, for the auxiliary component of the model, one can posit either

$$X_e = r(W; \alpha) + X_u \quad (\text{MP specification}) \tag{3}$$

or

$$g(X_e|W; \alpha) \quad (\text{FP specification}) \tag{4}$$

where α is a vector of parameters, $r(W; \alpha)$ denotes the conditional mean of X_e given $W = [X_o \ W^+]$, W^+ is a vector identifying instrumental variables, and $g(X_e|W; \alpha)$ is the conditional probability density function of X_e given W . By definition, the elements of W^+ must satisfy the following three conditions: 1) they are correlated with neither X_u nor e ; 2) they can be legitimately excluded from the outcome regression (1); and 3) they are strongly correlated with X_e . Equation (3) [or (4)] formalizes the correlation between X_u and X_e . The correlation between X_u and Y is formalized in the outcome regression (1) [or (2)]. The general two-stage residual inclusion (2SRI) protocol is the following:

First Stage: Apply the appropriate nonlinear least squares (NLS) [maximum likelihood (ML)] estimator to (3) [or (4)] to consistently estimate α .² The residuals from this regression are

$$\hat{X}_u = X_e - r(W; \hat{\alpha}) \tag{5}$$

where $\hat{\alpha}$ denotes the first-stage consistent estimate of α . Note that the FP specification in (4) will always imply the existence of a regression specification akin to (3), from which residuals, as defined in (5), can be obtained. To complete the first stage of 2SRI, save the residuals defined in (5).

Second Stage: To consistently estimate β , apply the appropriate NLS [ML] estimator to (1) [or (2)], with X_u replaced by \hat{X}_u .³

Note that one can use any combination of MP or FP specifications for the first and second stages of the 2SRI model. Correspondingly, any combination of NLS or ML can be implemented for first- and second-stage estimation. In the majority of applied

2. The first-stage ML estimator is the maximizer of $\sum_{i=1}^n \ln\{g(X_{ei}|W_i; \alpha)\}$ with respect to α , where X_{ei} and W_i denote the observed values of X_e and W for the i th observation in the sample and $i = 1, \dots, n$.

3. The second-stage ML estimator is the maximizer of $\sum_{i=1}^n \ln\{f(Y_i|X_{ei}, X_{oi}, \hat{X}_{ui}; \beta)\}$ with respect to β , where Y_i and X_{oi} denote the observed values of Y and X_o for the i th observation in the sample and where \hat{X}_{ui} is the first-stage residual for the i th observation in the sample.

settings, the 2SRI estimates of α and β are easy to obtain via packaged Stata commands. The asymptotically correct standard errors (ACSE), for use in estimation of confidence intervals and t statistics for testing hypotheses about the elements of β , can be calculated with additional Mata commands.

Before moving on to an example, note that the above model specification and corresponding estimator do not necessarily constitute a control function method (CFM) as defined by [Blundell and Powell \(2003\)](#).⁴ The assumption that I maintain above is that the O&A regressions are correctly specified by the researcher. As [Terza, Basu, and Rathouz \(2008\)](#) show, under this assumption, the 2SRI estimator consistently estimates the model parameters.⁵ To qualify as a CFM with accompanying consistency and robustness properties, the above 2SRI approach must satisfy other conditions. For a detailed discussion of such conditions, see [Wooldridge \(2014, 2015\)](#). To maintain the focus of this article (imparting practical aspects of 2SRI implementation in Stata), we abstract from such issues in the following sections. For simplicity of illustration and didactics, we maintain that for a correctly specified model, 2SRI affords the applied researcher a consistent, coherent but simple way to do empirical analyses for a very general class of nonlinear data-generating processes.

Consider the regression model of [Mullahy \(1997\)](#), in which the objective is to draw causal inferences regarding the effect of prenatal smoking (X_e -CIGSPREG) on infant birthweight (Y -BIRTHWTLB) while controlling for infant birth order (PARITY), race (WHITE), and sex (MALE). The regression model for the birthweight outcome that he proposed can be written in the MP form⁶

$$Y = \exp(X_e\beta_e + X_o\beta_o + X_u\beta_u) + e \quad (\text{outcome regression}) \quad (6)$$

where X_u comprises unobservable variables that are potentially correlated with prenatal smoking (for example, general “health mindedness” of the mother), e is the regression error term, $X_o = [\text{PARITY WHITE MALE}]$ is a row vector of regressors that are uncorrelated with X_u , and e and the β 's are the regression parameters. At issue here is the fact that there exist unobservables (as captured by X_u) that are correlated with both Y and X_e . In other words, X_e is endogenous. For illustrative purposes, we specify an FP version of the auxiliary component of the model in which

$$g(X_e|W; \alpha^*) = \{1 - \Phi(W\alpha_1)\}^{I(X_e=0)} \times \{\Phi(W\alpha_1) \ln \varphi(X_e, W\alpha_2^*, \sigma^2)\}^{\{1-I(X_e=0)\}} \quad (7)$$

4. Under the assumptions of [Blundell and Powell \(2003\)](#) (mainly linearity), in their discussion of CFM, the condition in expression (63) of [Wooldridge \(2014\)](#) is implied. [Wooldridge](#) also notes that, although (63) is not precluded in the nonlinear 2SRI framework, it is also not implied. Therefore, (63) must be imposed if 2SRI is to be interpreted as a CFM as in [Blundell and Powell \(2003\)](#).

5. Under the assumption that the model is correctly specified (and other general conditions), the consistency of the 2SRI estimator follows from the fact that it is a member of the class of two-stage M -estimators (see [Newey and McFadden \[1994, sec. 6\]](#); [White \[1994, chap. 6\]](#); [Wooldridge \[2010, chap. 12\]](#)).

6. [Mullahy \(1997\)](#) does not explicitly specify the model in terms of the unobservable X_u . Nevertheless, (6) is substantively identical to [Mullahy's \(1997\) model](#) (see [Terza \[2006\]](#)).

where $\alpha^{*'} = [\alpha_1' \ \alpha_2^{*'}]$, $\ln \varphi(A, b, c)$ denotes the probability density function of the log-normal random variable A with central tendency parameter b and dispersion parameter c , $W = [X_o \ W^+]$, and $W^+ = [\text{EDFATHER} \ \text{EDMOTHER} \ \text{FAMINCOME} \ \text{CIGTAX}]$, with

EDFATHER = paternal schooling in years
 EDMOTHER = maternal schooling in years
 FAMINCOME = family income

and

CIGTAX = cigarette tax

The specification in (7) indicates that prenatal smoking follows a two-part model with a probit formulation for the extensive margin (EM) and a lognormal intensive margin (IM). This is, in fact, a reasonable specification because a) there is a substantial proportion of nonsmokers in the population (and sample) of pregnant women; and b) the decision to smoke or not probably differs systematically from the decision regarding how much to smoke (among those who have decided to smoke at all). Based on (7), we can write the auxiliary regression as

$$X_p = \Phi(W\alpha_1)\exp(W\alpha_2) + X_u \quad (\text{auxiliary regression}) \tag{8}$$

where α_2 is the same as α_2^* , with the constant term shifted by $+(\sigma^2/2)$, because (7) implies that $E[X_e|W] = \Phi(W\alpha_1)\exp\{W\alpha_2^* + (\sigma^2/2)\}$. From (8), we have that $r(W; \alpha) = \Phi(W\alpha_1)\exp(W\alpha_2)$ and

$$X_u = X_e - \Phi(W\alpha_1)\exp(W\alpha_2) \tag{9}$$

where $\alpha' = [\alpha_1' \ \alpha_2']$. In the sequel, we will refer to the model in (6) through (9) as *the example*. For the generic nonlinear model with endogeneity [(1) through (4)], we offer a step-by-step protocol for using Stata and Mata to obtain the 2SRI estimate of β and the corresponding ACSE.⁷ We use the example to illustrate each of the steps.

2 The step-by-step 2SRI protocol

In detailing this protocol, we assume that the data have been input and that the analysis sample comprises n observations on the following variables: Y , X_e , X_o , and W_{plus} , corresponding to Y , X_e , X_o , and W^+ as generically defined above.

⁷ There are two other ways to calculate the standard errors: bootstrapping and the resampling method proposed by Krinsky and Robb (1986, 1990). For detailed discussions and pro-and-con evaluations of the bootstrapping and Krinsky and Robb (1986, 1990) methods, see Dowd, Greene, and Norton (2014). Dowd, Greene, and Norton (2014) also discuss the ACSE approach, but the formulation they offer [in particular, (17)] is based on an assumption that is usually invalid in econometric applications. See Terza (2016b) for details.

Step a: Specify the O&A components of the 2SRI model.

Any of four O&A combinations is possible based on the choice of MP versus FP specifications for each of the two estimation stages. For the second-stage outcome component, one can use $\mu(X_e, X_o, X_u; \beta)$ [MP specification in (1)] or $f(Y|X_e, X_o, X_u; \beta)$ [FP specification in (2)]. To make their dependence on α and β explicit, and for convenience of exposition, we rewrite the MP and FP versions of the outcome regression, respectively, as

$$\mu^*(X_e, W; \alpha, \beta) = \mu[X_e, X_o, \{X_e - r(W; \alpha)\}; \beta] \quad (10)$$

and

$$f^*(Y|X_e, W; \alpha, \beta) = f[Y|X_e, X_o, \{X_e - r(W; \alpha)\}; \beta]$$

For the first-stage auxiliary component, one can use $r(W; \alpha)$ [MP specification in (3)] or $g(X_e|W; \alpha)$ [FP specification in (4)]. MP (FP) O&A 2SRI components can be estimated via NLS (ML). In the example using (9), the following version of (10) is relevant,

$$\mu^*(X_e, W; \alpha, \beta) = \exp[X_e \beta_p + X_o \beta_o + \{X_e - \Phi(W \alpha_1) \exp(W \alpha_2)\} \beta_u] \quad (11)$$

where $\beta' = [\beta_e \ \beta_o' \ \beta_u]$.

Step b: Derive the requisite analytic components for calculation of the ACSE.

As Terza (2016a) shows, the exact form of the ACSE depends on the estimation method used in the second stage of 2SRI—NLS (for the MP specification) versus a maximum likelihood estimator (MLE) (for the FP specification). When an MLE is used in the second stage, the ACSE for the k th element of β is the square root of the k th diagonal element of the matrix,

$$V(\hat{\beta}) AV(\hat{\alpha}) A' V(\hat{\beta}) + V(\hat{\beta}) \quad (12)$$

where $V(\hat{\alpha})$ and $V(\hat{\beta})$ are the estimates of the covariance matrices output by the relevant Stata commands for the first and second stages of 2SRI, respectively, and

$$A = \sum_{i=1}^n \nabla_{\beta} \ln \hat{f}_i^* \nabla_{\alpha} \ln \hat{f}_i^* \quad (13)$$

with $\nabla_c \ln \hat{f}_i^*$ defined as the gradient of $f^*(Y|X_e, W; \alpha, \beta)$ with respect to c ($c = \alpha$ or β) evaluated at X_{ei} , $W_i[X_{oi} \ W_i^+]$, $\hat{\alpha}$, and $\hat{\beta}$ (“ i ” denotes the i th observation in the sample; $i = 1, \dots, n$). In this case, analytic expressions for $\nabla_{\beta} \ln f^*$ and $\nabla_{\alpha} \ln f^*$ must be derived.

Similarly, Terza (2016a) shows that when NLS is used in the second stage, the ACSE for the k th element of β is the square root of the k th diagonal element of the matrix,

$$B_1^{-1} B_2 V(\hat{\alpha}) B_2' B_1^{-1} + V(\hat{\beta}) \quad (14)$$

where $V(\hat{\alpha})$ and $V(\hat{\beta})$ are the estimated variance–covariance matrices of the first- and second-stage estimators of α and β , respectively, as output by Stata,

$$B_1 = \sum_{i=1}^n \nabla_{\beta} \hat{\mu}_i^* \nabla_{\beta} \hat{\mu}_i^* \tag{15}$$

and

$$B_2 = \sum_{i=1}^n \nabla_{\beta} \hat{\mu}_i^* \nabla_{\alpha} \hat{\mu}_i^* \tag{16}$$

with $\nabla_c \hat{\mu}_i^*$ defined as the gradient of $\mu^*(X_e, W; \alpha, \beta)$ with respect to c ($c = \alpha$ or β) evaluated at X_{ei} , $W_i = [X_{oi} \ W_i^+]$, $\hat{\alpha}$, and $\hat{\beta}$. This step requires that the user supply analytic expressions for $\nabla_{\beta} \mu^*$ and $\nabla_{\alpha} \mu^*$. In the example, it follows from (3) that

$$\nabla_{\beta} \mu^* = \exp(X\beta)X$$

and

$$\nabla_{\alpha} \mu^* = [\nabla_{\alpha_1} \mu^* \quad \nabla_{\alpha_2} \mu^*]$$

where

$$\begin{aligned} \nabla_{\alpha_1} \mu^* &= -\beta_u \exp(X\beta) \exp(W\alpha_2) \varphi(W\alpha_1) W \\ \nabla_{\alpha_2} \mu^* &= -\beta_u \exp(X\beta) \exp(W\alpha_2) \Phi(W\alpha_1) W \\ X &= [X_e \ X_o \ X_u] \quad \text{and} \quad W = [X_o \ W^+] \end{aligned}$$

Therefore,

$$\nabla_{\beta} \hat{\mu}_i^* = \exp(X_i \hat{\beta}) X_i$$

and

$$\nabla_{\alpha} \hat{\mu}_i^* = [\nabla_{\alpha_1} \hat{\mu}_i^* \quad \nabla_{\alpha_2} \hat{\mu}_i^*]$$

where

$$\begin{aligned} \nabla_{\alpha_1} \mu^* &= -\hat{\beta}_u \exp(X_i \hat{\beta}) \exp(W_i \hat{\alpha}_2) \varphi(W_i \hat{\alpha}_1) W_i \\ \nabla_{\alpha_2} \mu^* &= -\hat{\beta}_u \exp(X_i \hat{\beta}) \exp(W_i \hat{\alpha}_2) \Phi(W_i \hat{\alpha}_1) W_i \\ X_i &= [X_{ei} \ X_{oi} \ \hat{X}_{ui}] \quad \text{and} \quad \hat{X}_{ui} = X_{ei} - \Phi(W_i \hat{\alpha}_1) \exp(W_i \hat{\alpha}_2) \end{aligned}$$

Generally (second-stage ML or NLS), based on standard asymptotic theory, the “ t statistic” is

$$\frac{\hat{\beta}(k) - \beta(k)}{\sqrt{\hat{D}(k)}} \tag{17}$$

for the k th element of β , and $[\beta(k)]$ is asymptotically standard normally distributed, where $\hat{\beta}(k)$ is the 2SRI estimator of $\beta(k)$ and $\hat{D}(k)$ denotes the k th diagonal element of (3) or (13). This t statistic can be used to test the hypothesis that $\beta(k) = \beta(k)^0$ for $\beta(k)^0$ —a given null value of $\beta(k)$.

Step c: Apply the appropriate Stata commands for $r(W, \alpha)$ [$g(X_e|W; \alpha)$] when the first stage is NLS [MLE] to obtain the first-stage estimate of α by regressing X_e on X_o and W plus.

In the example, the parameter vector for the first part (EM) of the auxiliary component of the model (α_1) can be estimated by applying the Stata `probit` command to the full sample, with $[1 - I(X_e = 0)]$ as the dependent variable and W as the vector of regressors, where $I(C)$ denotes the indicator function that takes the value 1 if condition C holds and 0 otherwise. The parameters of the second part (IM) of the auxiliary component of the model (α_2) can be consistently estimated by applying the Stata `glm` command to the subsample of nonzero smokers, with X_e as the dependent variable and W as the vector of regressors.

```

/*****
** Generate the binary smoking variable.      **
*****/
gen ANYCIGS=CIGSPREG>0

/*****
** 2SRI first-stage first-part probit estimates.**
*****/
/*Step c*/
probit ANYCIGS PARITY WHITE MALE EDFATHER EDMOTHER ///
      FAMINCOM CIGTAX88
.
.
.

/*****
** 2SRI first-stage second-part probit NLS      **
** estimates.                                  **
*****/
/*Step c*/
glm CIGSPREG PARITY WHITE MALE EDFATHER EDMOTHER ///
    FAMINCOM CIGTAX88 if ANYCIGS==1,      ///
    family(gaussian) link(log) vce(robust)

```

Step d: Use the appropriate command or option to calculate and save the first-stage regression residuals, say, as the additional variable X_{uhat} .

In the context of the example, we have

```

/*Step d*/
predict CIGPROB
.
.
.

/*Step d*/
predict CIGMEAN
.
.
.

```

```

/*****
** Generate the first-stage residuals.      **
*****/
/*Step d*/
gen Xuhat=CIGSPREG-CIGPROB*CIGMEAN

```

The first (second) `predict` is placed immediately after the `probit` (`glm`) command in step c and produces the first-stage first (or second)-part `probit` (exponential regression) predictions $\Phi(W_i\hat{\alpha}_1)$ [$\exp(W_i\hat{\alpha}_2)$].

Step e: Use the appropriate Stata and Mata commands to save the vector of first-stage coefficient estimates and its corresponding estimated covariance matrix (as calculated and output by the relevant Stata commands used in step c) so that they are accessible in Mata; call them, for example, `alphahat` and `Valphahat`, respectively.

In the context of the example, we have

```

/*****
** Save the first-stage first-part probit      **
** estimates and estimated covariance matrix.  **
*****/
/*Step e*/
mata: alpha1hat=st_matrix("e(b)")`
mata: Valpha1hat=st_matrix("e(V)")
.
.
.
/*****
** Save the first-stage second-part NLS      **
** estimates and estimated covariance matrix.  **
*****/
/*Step e*/
mata: alpha2hat=st_matrix("e(b)")`
mata: Valpha2hat=st_matrix("e(V)")

```

The first (second) pair of Mata commands is placed immediately after the `predict` `CIGPROB` (`predict CIGMEAN`) command in step d. The `st_matrix(name)` function turns the Stata matrix `name` into a Mata matrix. In this context, the `probit` and `glm` commands produce the vector of coefficient parameter estimates `e(b)` and estimated covariance matrix `e(V)` among their stored results. The `st_matrix()` function transforms them to Mata-usable format.

Step f: Apply the appropriate Stata commands for $\mu(X_e, X_o, X_u; \beta)$ [$f(Y|X_e, X_o, X_u; \beta)$] when the 2SRI second stage is NLS [ML] to obtain the second-stage estimate of β by regressing `Y` on `Xe`, `Xo`, and `Xuhat`.

In the context of the example, we have

```

/*Step f*/
glm BIRTHWTLB CIGSPREG PARITY WHITE MALE Xuhat, ///
    family(gaussian) link(log) vce(robust)

```

Step g: Use the Stata and Mata commands to save the vector of second-stage coefficient estimates and its corresponding estimated covariance matrix (as calculated and output by the relevant Stata commands used in step f) so that they are accessible in Mata; call them, for example, `betahat` and `Vbetahat`, respectively (you might also have to single out $\hat{\beta}_u$).

In the context of the example, we have

```
/*Step g*/
mata: betahat=st_matrix("e(b)")
mata: Vbetahat=st_matrix("e(V)")
mata: Bu=betahat[5]
```

The last statement uses matrix subscripting and the fact that $\hat{\beta}_u$ is the fifth element of the estimated coefficients of the exponential outcome regression.

Step h: Construct X and W matrices in Mata, where X is the matrix that has columns that are X_e , X_o , and a constant term (a column vector of 1s); and W has columns X_o , W_{plus} , and a constant term.⁸

In the context of the example, we have

```
/*Step h*/
putmata BIRTHWTLB CIGSPREG ANYCIGS PARITY WHITE ///
        MALE EDFATHER EDMOTHER FAMINCOM CIGTAX88 Xuhat

/*Step h*/
mata: X=CIGSPREG, PARITY, WHITE, MALE,          ///
        Xuhat, J(rows(PARITY),1,1)
mata: W=PARITY, WHITE, MALE, EDFATHER, EDMOTHER, ///
        FAMINCOM, CIGTAX88, J(rows(PARITY),1,1)
```

The `putmata` command converts designated variables in the relevant Stata dataset to vectors in Mata-usable format.

Step i: Use `alphahat`, `betahat`, X , W , and the analytic results obtained in step b to construct the two gradient matrices needed to calculate the correct standard errors for `betahat`, say, `gradbeta` and `gradalpha`. Note that `gradbeta` will have n rows and K columns, where K is the column dimension of X , and `gradalpha` will have n rows and S columns, where S is the column dimension of W . The exact forms of these gradient matrices will depend on whether ML or NLS was implemented in the second stage of the 2SRI estimator. If ML was used, then the i th rows of `gradbeta` and `gradalpha` will be $\nabla_{\beta} \ln \hat{f}_i^*$ and $\nabla_{\alpha} \ln \hat{f}_i^*$, respectively, as defined in (12). If the 2SRI second stage is NLS, then the i th rows of `gradbeta` and `gradalpha` will be $\nabla_{\beta} \hat{\mu}_i^*$ and $\nabla_{\alpha} \hat{\mu}_i^*$, respectively, as defined in (14) and (16).

8. Be sure that the ordering of the columns of X and W (including the constant term) conforms to the ordering of the estimated coefficients in `betahat` and `alphahat`.

In the context of the example, we have

```

/*Step i*/
mata: gradbeta=exp(X*betahat):*X
mata: gradalpha1=
-Bu:*exp(X*betahat):*normalden(W*alpha1hat):*exp(W*alpha2hat):*W
mata: gradalpha2=
-Bu:*exp(X*betahat):*normal(W*alpha1hat):*exp(W*alpha2hat):*W
mata: gradalpha=gradalpha1,gradalpha2

```

Step j: Calculate A [B₁ and B₂] as defined in (12) [(14) and (16)].

If the 2SRI second stage is ML, then calculate the A matrix as⁹

```
A = gradbeta' * gradbeta
```

based on (12). Because the 2SRI second stage in the example is NLS, we calculate the B1 and B2 matrices as

```

/*Step j*/
mata: B1=gradbeta'*gradbeta
mata: B2=gradbeta'*gradalpha

```

based on (14) and (16), respectively.

Step k: Calculate the asymptotic covariance matrix of $\hat{\beta}$.

If the 2SRI second stage is ML, then calculate the estimated asymptotic covariance matrix of betahat as

```
AVARBeta = Vbetahat * A * Valphahat * A' Vbetahat' + Vbetahat
```

based on (3). Because the 2SRI second stage in the example is NLS, we calculate the estimated asymptotic covariance matrix of betahat as

```

/*Step k*/
mata: Valphahat=blockdiag(Valpha1hat,Valpha2hat)
mata: Dhat=invsym(B1)*B2*Valphahat*B2'*invsym(B1)+Vbetahat

```

based on (13). Note that we first had to stack up the full estimated covariance matrix of $\hat{\alpha} = [\hat{\alpha}'_1 \hat{\alpha}'_2]$ from the first- and second-part outputs for the first-stage 2SRI estimate of α .

9. Here we use the following summation or matrix equality: Let \mathbf{Z}_i and \mathbf{Q}_i be the K and S dimensional row vectors, respectively ($I = 1, \dots, n$), and let \mathbf{Z} and \mathbf{Q} be the $n \times K$ and $n \times S$ matrices with i th rows that are \mathbf{Z}_i and \mathbf{Q}_i , respectively; then

$$\sum_{i=1}^n \mathbf{Z}'_i \mathbf{Q}_i = \mathbf{Z}' \mathbf{Q}$$

Step l: Calculate the vector of asymptotic standard errors for $\hat{\beta}$.

Regardless of the estimator used in the 2SRI second stage, use

```
mata: ACSE=sqrt(diagonal(AVARBeta))
```

Step m: Calculate the vector of asymptotic t statistics to be used to test the conventional null hypothesis regarding the elements of β (namely, $H_0: \beta_k = 0$, where β_k denotes the kth element of β).

Regardless of the estimator used in the 2SRI second stage, use

```
/*Step m*/
mata: Betatstats=Betahat:/ACSE.
```

The k th element of `Betatstats` corresponds with (17). The full Stata code for this protocol as it pertains to the example is given in the appendix.

I applied the above 2SRI estimation protocol to the same dataset analyzed by Mullahy (1997). The estimation results for α and β are reported in tables 1 and 2, respectively. The correct asymptotic t statistics for the 2SRI estimate of β , reported in column 3 of table 2, were calculated using (13). In table 2, we also display Mullahy's generalized method of moments (GMM) estimates and, as a baseline, report the conventional NLS estimates that ignore potential endogeneity. As an indicator of the strength of the instrumental variables (that is, the elements of W^+), we conducted a Wald test of their joint significance. The value of the chi-squared test statistic is 49.33, so the null hypothesis that their coefficients are jointly zero is roundly rejected at any reasonable level of significance. The second-stage 2SRI estimates shown in table 2 (column 2) are virtually identical to Mullahy's GMM estimates (column 5), but the former, unlike the latter, provide a direct test of the endogeneity of the prenatal smoking variable via the asymptotic t statistic (5th element of $\hat{\beta}$) for the coefficient of $X_u[\hat{\beta}_u = \hat{\beta}(5)]$ with $H_0: \beta_u = \beta(5) = 0$. According to the results of this test, the exogeneity null hypothesis is rejected at nearly the 1% significance level. To get a sense of the bias from neglecting to account for the two-stage nature of the estimator in the calculation of the asymptotic standard errors, in table 2 (last column), we also display the "packaged" second-stage `glm` t statistics as reported in the Stata output. The mean absolute bias across these uncorrected asymptotic t statistics for the four control variables and X_u is nearly 9%.

Table 1. 2SRI first-stage estimates

Variable	Estimate	Asymptotic <i>t</i> statistic	<i>p</i> -value
First-stage estimate of α_1			
PARITY	0.02	0.39	0.696
WHITE	0.25	2.16	0.031
MALE	-0.16	-1.88	0.060
EDFATHER	-0.02	-2.38	0.017
EDMOTHER	-0.12	-5.54	0.000
FAMINCOM	-0.01	-2.87	0.004
CIGTAX	0.01	2.25	0.024
Constant	0.56	1.93	0.054
First-stage estimate of α_2			
PARITY	0.10	1.34	0.182
WHITE	0.00	0.00	0.998
MALE	0.21	2.13	0.033
EDFATHER	-0.02	-1.43	0.153
EDMOTHER	-0.03	-0.87	0.386
FAMINCOM	0.00	0.28	0.778
CIGTAX	0.00	-0.39	0.697
Constant	2.82	6.00	0.000

$n = 1388$

Table 2. 2SRI second-stage, GMM, and NLS estimates

Variable	Estimate	2SRI		GMM		NLS	
		Correct asymptotic t statistic	Uncorrected asymptotic t statistic	Estimate	Asymptotic t statistic	Estimate	Asymptotic t statistic
CIGS	-0.01	-4.07	-4.41	-0.01	-3.46	0.00	-5.62
PARITY	0.02	3.36	3.66	0.02	3.33	0.01	2.99
WHITE	0.05	4.45	4.61	0.05	4.44	0.06	4.75
MALE	0.03	2.80	2.90	0.03	2.95	0.03	2.90
X_u	0.01	2.66	2.89	-	-	-	-
Constant	1.94	124.67	129.70	1.94	121.71	1.93	133.70

 $n = 1388$

3 Other oft-encountered O&A combinations

Nonlinearity in regression modeling is typically implied by limitations on the support of the dependent variable. For instance, the linear specification is clearly unappealing for models with binary or fractional support. Another commonly encountered dependent variable type that prompts nonlinear modeling is one whose support is the nonnegative half of the real line (including zero). In the previous section, in the context of the example, we discussed a particular version of this case, in which there is i) a nontrivial proportion of zeros in the population (sample); and ii) a reason to believe that the EM (zero or not) should be modeled differently from the IM (value of the dependent variable conditional on it being nonzero). In a simpler (nested) version of this model, there is no need to distinguish between the EM and IM in modeling. In the example, if there were no reason to believe that the decision regarding whether or not to smoke during pregnancy (IM) is systematically different from one's choice of how much to smoke (EM), then we would replace (7) with

$$X_e = \exp(W\alpha) + X_u$$

and implement NLS for 2SRI first-stage estimation of α . We leave it to the reader to supply the details of the above step-by-step 2SRI protocol for this case. In the remainder of this section, we discuss binary and fractional O&A specifications.

Consider the details of the step-by-step protocol when X_e is binary and Y is fractional. From the following discussion of this case, the reader should be able to infer the details of the protocol for the remaining three possible O&A specifications involving these two variable types.

Step a: In this case, the first- and second-stage estimators are ML and NLS, respectively. The conditional pdf for ML in the first stage is

$$g(X_e|W; \alpha) = \Phi(W\alpha)^{X_e} \{1 - \Phi(W\alpha)\}^{1-X_e} \tag{18}$$

where $\Phi(\cdot)$ denotes the standard normal cumulative distribution function.¹⁰ Note that (15) implies that $r(W, \alpha) = \Phi(W\alpha)$. The functional form for the outcome regression in (1) and (10) is

$$\begin{aligned} \mu^*(X_e, W; \alpha, \beta) &= \mu(X_e, X_o, \{X_e - r(W; \alpha)\}; \beta) \\ &= \Phi(X\beta) = \Phi[X_e\beta_p + X_o\beta_o + \{X_e - \Phi(W\alpha)\}\beta_u] \end{aligned}$$

Step b:

$$\nabla_{\beta} \mu^* = \varphi(X\beta)X$$

and

$$\nabla_{\alpha} \mu^* = -\beta_u \varphi(X\beta) \varphi(W\alpha) W$$

10. $\Phi(\cdot)$ can be replaced here by any convenient (packaged) cumulative distribution function.

Step c:

```
/*step c*/
probit Xe Xo Wplus
```

Step d:

```
/*step d*/
predict phiWalpha, p
gen Xuhat=Y-phiWalpha
```

Step e:

```
/*step e*/
mata: alphahat=st_matrix("e(b)")´
mata: Valphahat=st_matrix("e(V)")
```

Step f:

```
/*step f*/
glm Y Xe Xo Xuhat,family(gaussian) link(probit) vce(robust)
```

Step g:

```
/*step g*/
mata: betahat=st_matrix("e(b)")´
mata: Vbetahat=st_matrix("e(V)")
mata: Bu=betahat[3]
```

Step h:

```
/*step h*/
putmata Y Xe Xo Wplus Xuhat
mata: X=Xe, Xo, Xuhat, J(rows(Xo),1,1)
mata: W=Xo, Wplus, J(rows(Xo),1,1)
```

Step i:

```
/*step i*/
mata: gradbeta=normalden(X*betahat):*X
mata: gradalpha=-Bu:*normalden(X*betahat):*/*
*/normalden(W*alphahat):*W
```

Step j:

```
/*step j*/
mata: B1 = gradbeta´*gradbeta
mata: B2 = gradbeta´*gradalpha
```

Step k:

```
/*step k*/
mata: AVARBeta=invsym(B1)*B2*Valphahat*B2´invsym(B1)/*
*/+ Vbetahat
```

Step l:

```
/*step l*/
mata: ACSE = sqrt(diagonal(AVARBeta))
```

Step m:

```
/*step m*/
mata: ACtstats=betahat:/ACSE
```

4 Summary and discussion

I reviewed the 2SRI method for nonlinear models with endogenous regressors and offered a step-by-step protocol for its implementation in Stata. I illustrated its application with real data for when both X_e and Y are nonnegative. In empirical practice, cases in which X_e , Y , or both are binary or fractional often arise. I detailed Stata and Mata implementation of the protocol for the version of the model in which X_e is binary and Y is fractional. I hope that these examples will serve to demonstrate the ease with which the protocol can be extended to models involving other variable-type configurations not explicitly covered here. In particular, the class of nonnegative dependent variables encompasses important subtypes; for example, count variables and continuous variables with support that does not include 0. For instance, one might seek to fit a model with an endogenous count regressor and an outcome whose distribution is skewed with 0 excluded. In this case, $g(X_e|W; \alpha)$ might be specified as Poisson and $f(Y|X_e, W, X_u; \alpha, \beta)$ as generalized Gamma. In Stata, the first-stage MLE of α would be obtained using the `poisson` command. The `streg` command with the `distribution(ggamma)` option would be used to obtain the second-stage MLE of β . The ACSEs for the elements of β would be obtained using our proposed protocol.

5 Acknowledgments

This research was supported by a grant from the Agency for Healthcare Research and Quality (R01 HS017434-01). This article was presented at the Stata Conference in Chicago, IL, July 28–29, 2016. Please do not quote without the author's permission.

6 References

- Blundell, R., and J. L. Powell. 2003. Endogeneity in nonparametric and semiparametric regression models. In *Advances in Economics and Econometrics: Theory and Applications, Eight World Congress, vol. 2*, ed. M. Dewatripont, L. P. Hansen, and S. J. Turnovsky, 312–357. Cambridge: Cambridge University Press.
- Dowd, B. E., W. H. Greene, and E. C. Norton. 2014. Computation of standard errors. *Health Services Research* 49: 731–750.

- Krinsky, I., and A. Robb. 1990. On approximating the statistical properties of elasticities: A correction. *Review of Economics and Statistics* 72: 189–190.
- Krinsky, I., and A. L. Robb. 1986. On approximating the statistical properties of elasticities. *Review of Economics and Statistics* 68: 715–719.
- Mullahy, J. 1997. Instrumental-variable estimation of count data models: Applications to models of cigarette smoking behavior. *Review of Economics and Statistics* 79: 586–593.
- Newey, W. K., and D. McFadden. 1994. Large sample estimation and hypothesis testing. In *Handbook of Econometrics*, vol. 4, ed. R. F. Engle and D. L. McFadden, 2111–2245. Amsterdam: Elsevier.
- Terza, J. V. 2006. Estimation of policy effects using parametric nonlinear models: A contextual critique of the generalized method of moments. *Health Services and Outcomes Research Methodology* 6: 177–198.
- . 2016a. Simpler standard errors for two-stage optimization estimators. *Stata Journal* 16: 368–385.
- . 2016b. Inference using sample means of parametric nonlinear data transformations. *Health Services Research* 51: 1109–1113.
- Terza, J. V., A. Basu, and P. J. Rathouz. 2008. Two-stage residual inclusion estimation: Addressing endogeneity in health econometric modeling. *Journal of Health Economics* 27: 531–543.
- White, H. 1994. *Estimation, Inference and Specification Analysis*. New York: Cambridge University Press.
- Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.
- . 2014. Quasi-maximum likelihood estimation and testing for nonlinear models with endogenous explanatory variables. *Journal of Econometrics* 182: 226–234.
- . 2015. Control function methods in applied econometrics. *Journal of Human Resources* 50: 420–445.

About the author

Joseph V. Terza is a health economist and econometrician in the Department of Economics at Indiana University–Purdue University Indianapolis. His research focuses on the development and application of methods for estimating qualitative and limited dependent variable models with endogeneity. Two of his methods have been implemented as Stata commands. He was a keynote speaker at the Stata Users Group meeting in Mexico City in November 2014.

Appendix: Stata and Mata do-files and log files for the example

Stata and Mata code

```

. /*****
> ** Read in the data.          **
> *****/
. use mullahy-birthweight-data-lbs-not-oz
.
. /*****
> ** Generate the binary smoking variable.  **
> *****/
. generate ANYCIGS=CIGSPREG>0
.
. /*****
> ** 2SRI first-stage first-part probit estimates.**
> *****/
. /*Step c*/
. probit ANYCIGS PARITY WHITE MALE EDFATHER EDMOTHER
>       FAMINCOM CIGTAX88
Iteration 0:   log likelihood = -593.2711
Iteration 1:   log likelihood = -539.2207
Iteration 2:   log likelihood = -537.93241
Iteration 3:   log likelihood = -537.9313
Iteration 4:   log likelihood = -537.9313

Probit regression                               Number of obs   =       1,388
                                                LR chi2(7)      =       110.68
                                                Prob > chi2     =        0.0000
Log likelihood = -537.9313                    Pseudo R2      =        0.0933

```

ANYCIGS	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
PARITY	.0183594	.0470494	0.39	0.696	-.0738558	.1105746
WHITE	.2484636	.1148504	2.16	0.031	.023361	.4735663
MALE	-.1628769	.0864755	-1.88	0.060	-.3323658	.006612
EDFATHER	-.0239095	.0100267	-2.38	0.017	-.0435614	-.0042576
EDMOTHER	-.1199751	.0216733	-5.54	0.000	-.162454	-.0774962
FAMINCOM	-.0092103	.0032144	-2.87	0.004	-.0155104	-.0029101
CIGTAX88	.0127688	.0056673	2.25	0.024	.0016611	.0238766
_cons	.5600838	.2908317	1.93	0.054	-.0099359	1.130104

```

.
. /*****
> ** Save the 2SRI first-stage first-part probit **
> ** predicted values for use in calculating      **
> ** the first stage residuals.                  **
> *****/
. /*Step d*/
. predict CIGPROB
(option pr assumed; Pr(ANYCIGS))
.
. /*****
> ** Save the first-stage first-part probit      **
> ** estimates and estimated covariance matrix.  **
> *****/

```

```

. /*Step e*/
. mata: alphaihat=st_matrix("e(b)")`
. mata: Valphaihat=st_matrix("e(V)")
.
. /*****
> ** 2SRI first-stage second-part probit NLS      **
> ** estimates.                                  **
> *****/
. /*Step c*/
. glm CIGSPREG PARITY WHITE MALE EDFATHER EDMOTHER
>     FAMINCOM CIGTAX88 if ANYCIGS==1,
>     family(gaussian) link(log) vce(robust)

Iteration 0:  log pseudolikelihood = -768.10967
Iteration 1:  log pseudolikelihood = -751.55365
Iteration 2:  log pseudolikelihood = -750.05496
Iteration 3:  log pseudolikelihood = -750.05493

Generalized linear models                    No. of obs    =      212
Optimization      : ML                      Residual df   =      204
Scale parameter = 71.99373
Deviance          = 14686.72175              (1/df) Deviance = 71.99373
Pearson           = 14686.72175              (1/df) Pearson  = 71.99373
Variance function: V(u) = 1                  [Gaussian]
Link function     : g(u) = ln(u)             [Log]
AIC               = 7.151462
Log pseudolikelihood = -750.0549266         BIC           = 13593.98

```

CIGSPREG	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
PARITY	.1004253	.0752068	1.34	0.182	-.0469773	.2478279
WHITE	.0002311	.11928	0.00	0.998	-.2335533	.2340156
MALE	.2066734	.0968097	2.13	0.033	.0169298	.396417
EDFATHER	-.0157006	.0109983	-1.43	0.153	-.0372569	.0058557
EDMOTHER	-.027413	.031649	-0.87	0.386	-.0894439	.034618
FAMINCOM	.0011098	.0039345	0.28	0.778	-.0066017	.0088212
CIGTAX88	-.0028822	.0074149	-0.39	0.697	-.0174151	.0116507
_cons	2.821627	.4702037	6.00	0.000	1.900044	3.743209

```

. /*****
> ** Save the 2SRI first-stage second-part NLS      **
> ** (glm) predicted values for use in calculating**
> ** the first-stage residuals.                    **
> *****/
. /*Step d*/
. predict CIGMEAN
(option mu assumed; predicted mean CIGSPREG)
.
. /*****
> ** Generate the first-stage residuals.            **
> *****/
. /*Step d*/
. generate Xuhat=CIGSPREG-CIGPROB*CIGMEAN
.

```

```

. /*****
> ** Save the first-stage second-part NLS      **
> ** estimates and estimated covariance matrix. **
> *****/
. /*Step e*/
. mata: alpha2hat=st_matrix("e(b)")`
. mata: Valpha2hat=st_matrix("e(V)")
.
. /*****
> ** Descriptive statistics.      **
> *****/
. summ

```

Variable	Obs	Mean	Std. Dev.	Min	Max
BIRTHWT	1,388	118.6996	20.35396	23	271
CIGSPREG	1,388	2.087176	5.972688	0	50
PARITY	1,388	1.632565	.8940273	1	6
WHITE	1,388	.7845821	.4112601	0	1
MALE	1,388	.5208934	.4997433	0	1
EDFATHER	1,388	11.32421	5.251299	0	18
EDMOTHER	1,388	12.92651	2.401109	0	18
FAMINCOM	1,388	29.02666	18.73928	.5	65
CIGTAX88	1,388	19.55295	7.795598	2	38
BIRTHWTLB	1,388	7.418723	1.272123	1.4375	16.9375
ANYCIGS	1,388	.1527378	.3598642	0	1
CIGPROB	1,388	.1520482	.1038465	.0049521	.7636681
CIGMEAN	1,388	12.86834	2.512522	7.946904	28.78438
Xuhat	1,388	.0063805	5.818791	-15.09198	46.96746

```

. /*****
> ** 2SRI second-stage NLS estimates.      **
> *****/
. /*Step f*/
. glm BIRTHWTLB CIGSPREG PARITY WHITE MALE Xuhat,
>       family(gaussian) link(log) vce(robust)

Iteration 0:  log pseudolikelihood = -2271.1401
Iteration 1:  log pseudolikelihood = -2263.4591
Iteration 2:  log pseudolikelihood = -2263.4109
Iteration 3:  log pseudolikelihood = -2263.4109

Generalized linear models                No. of obs    =    1,388
Optimization      : ML                   Residual df   =    1,382
                                                Scale parameter =  1.533962
Deviance          =  2119.935722          (1/df) Deviance =  1.533962
Pearson           =  2119.935722          (1/df) Pearson  =  1.533962
Variance function: V(u) = 1              [Gaussian]
Link function     : g(u) = ln(u)         [Log]
                                                AIC           =  3.270045
Log pseudolikelihood = -2263.410887     BIC           = -7879.69

```

BIRTHWTLB	Robust		z	P> z	[95% Conf. Interval]	
	Coef.	Std. Err.				
CIGSPREG	-.0119672	.0027167	-4.41	0.000	-.0172918	-.0066427
PARITY	.0183912	.0050259	3.66	0.000	.0085405	.0282419
WHITE	.0542038	.0117566	4.61	0.000	.0311614	.0772463
MALE	.0259255	.0089519	2.90	0.004	.0083802	.0434708
Xuhat	.0077064	.0026665	2.89	0.004	.00248	.0129327
_cons	1.942015	.0149736	129.70	0.000	1.912667	1.971363

```

.
. /*****
> ** Save second-stage estimates and covariance **
> ** matrix. Single out the coefficient estimate **
> ** for Xu.                                     **
> *****/
. /*Step g*/
. mata: betahat=st_matrix("e(b)")
. mata: Vbetahat=st_matrix("e(V)")
. mata: Bu=betahat[5]
.
. /*****
> ** Send the requisite variables to Mata as      **
> ** vectors.                                     **
> *****/
. /*Step h*/
. putmata BIRTHWTLB CIGSPREG ANYCIGS PARITY WHITE
>       MALE EDFATHER EDMOTHER FAMINCOM CIGTAX88 Xuhat
(11 vectors posted)
.

```

```

. /*****
> ** Use these vectors to concatenate the needed **
> ** matrices. **
> *****/
. /*Step h*/
. mata: X=CIGSPREG, PARITY, WHITE, MALE,
>       Xuhat, J(rows(PARITY),1,1)
. mata: W=PARITY, WHITE, MALE, EDFATHER, EDMOTHER,
>       FAMINCOM, CIGTAX88, J(rows(PARITY),1,1)
.
. /*****
> ** Set up the two gradient matrices for the ACSE**
> *****/
. /*Step i*/
. mata: gradbeta=exp(X*betahat):*X
. mata: gradalpha1=-Bu*exp(X*betahat):*normalden(W*alpha1hat):*exp(W*alpha2hat):*W
. mata: gradalpha2=-Bu*exp(X*betahat):*normal(W*alpha1hat):*exp(W*alpha2hat):*W
. mata: gradalpha=gradalpha1,gradalpha2
.
. /*****
> ** Set up the B1 and B2 matrices for the ACSE. **
> *****/
. /*Step j*/
. mata: B1=gradbeta`*gradbeta
. mata: B2=gradbeta`*gradalpha
.
. /*****
> ** Set up the full estimated asymptotic **
> ** covariance matrix for alpha (first-stage **
> ** two-part model covariance matrix estimator as**
> ** output by Stata). **
> *****/
. /*Step k*/
. mata: Valphahat=blockdiag(Valpha1hat,Valpha2hat)
.
. /*****
> ** Construct the covariance matrix of the **
> ** second-stage Beta estimates. **
> *****/
. /*Step k*/
. mata: Dhat=invsym(B1)*B2*Valphahat*B2`*invsym(B1)+Vbetahat
.
. /*****
> ** Extract the vector of asymptotically correct **
> ** standard errors for betahat. **
> *****/
. /*Step l*/
. mata: ACSE=sqrt(diagonal(Dhat))
.

```

```

. /*****
> ** Calculate the corresponding vector of      **
> ** asymptotically correct t-stats.          **
> *****/
. /*Step m*/
. mata: ACtstats=betahat:/ACSE
.
. /*****
> ** Compute the corresponding vector of p-values.
> *****/
. mata: ACpvalues=2:*(1:-normal(abs(ACtstats)))
.
. /*****
> ** Display results.                          **
> *****/
. mata: header="Variable","Estimate","ACSE","AC t-stat","pvalue"
. mata: varnames="CIGSPREG", "PARITY", "WHITE", "MALE", "Xuhat","Constant"
. mata: results=betahat,ACSE,ACtstats,ACpvalues
. mata: resview=stroofreal(results)
. mata: "2SRI Results with ACSE"
      2SRI Results with ACSE
. mata: header \ (varnames`,resview)
           1           2           3           4           5
1  Variable  Estimate      ACSE  AC t-stat  pvalue
2  CIGSPREG  -.0119672    .002939  -4.071839  .0000466
3  PARITY    .0183912    .0054684  3.363166  .0007705
4  WHITE     .0542038    .0121787  4.450694  8.56e-06
5  MALE      .0259255    .009266  2.797918  .0051433
6  Xuhat     .0077064    .0028991  2.658169  .0078566
7  Constant  1.942015    .0155771  124.6715  0

```

Causal effect estimation and inference using Stata

Joseph V. Terza
Department of Economics
Indiana University–Purdue University Indianapolis
Indianapolis, IN
jvterza@iupui.edu

Abstract. [Terza \(2016b\)](#), *Health Services Research* 51: 1109–1113) gives the correct generic expression for the asymptotic standard errors of statistics formed as sample means of nonlinear data transformations. In this article, I assess the performance of the Stata `margins` command as a relatively simple alternative for calculating such standard errors. I note that `margins` is not available for all packaged nonlinear regression commands in Stata and cannot be implemented in conjunction with user-defined-and-coded nonlinear estimation protocols that do not make a `predict` command available. When `margins` is available, however, I establish (using a real-data example) that it produces standard errors that are asymptotically equivalent to those obtained from the formulations in [Terza \(2016b\)](#) and the appendix available with this article. This result favors using `margins` (with its relative coding simplicity) when available. In all other cases, use Mata to code the standard-error formulations in [Terza \(2016b\)](#). I discuss examples, and I give corresponding Stata do-files in appendices.

Keywords: st0506, margins, causal effect estimation, causal inference

1 Introduction

[Terza \(2016b\)](#) gives the correct generic expression for the asymptotic standard errors of statistics formed as sample means of nonlinear data transformations. In this article, I offer guidance to empirical researchers for implementing such statistics (and their standard errors) in Stata for causal inference—for example, the estimation of the average treatment effect (ATE), average incremental effect (AIE), or average marginal effect (AME). I discuss (and detail in the appendixes) simple Stata and Mata commands that can be used to calculate these causal estimators and their correct standard errors in any nonlinear modeling context. I also explore the potential use of `margins` as a simpler alternative for standard-error calculation. Pursuant to this, I ask and answer the following:

Q1: For which nonlinear regression estimation protocols coded in Stata is `margins` available (or unavailable)?

and

Q2: For the cases in which `margins` is available, are there relevant versions of `margins` (RM) that will return the correct standard errors as obtained using the [Terza \(2016b\)](#) formulations (TF) coded in Mata?

The answer to Q1 can be viewed as defining necessary conditions for using `margins`. Meanwhile, the existence and correctness of RM, as defined by the answer to Q2 if affirmative, supplies a sufficient condition under which `margins` would be appropriate. In the answer to Q2, we find that in any particular empirical study—conditional on availability as in Q1—although the values of the standard errors returned by RM are not identical to those produced by TF, the two approaches are asymptotically equivalent (that is, virtually the same in large samples). Therefore, when `margins` is available, one should use the relatively easy-to-code RM. In all other cases, one can use TF.¹

The remainder of the article is organized as follows: In the next section, I discuss availability of `margins` for Stata-based causal inference in nonlinear empirical settings (therefore answering Q1). In section 3, I review the TF—the generic formulation of the asymptotically correct standard errors in this context suggested by Terza (2016b). I apply the TF to data from Fishback and Terza (1989) and estimate the following causal effects on employment (the binary outcome variable of interest) and their standard errors: the ATE of disability (a binary variable), the AME of income (a continuous variable), and the AIE of an additional year of work experience (a continuous variable). I then detail the RM that should be used in this empirical context, for which `margins` is available. By answering Q2, I re-estimate the ATE for disability, the AME for income, and the AIE for experience using the RM and the Fishback and Terza (1989) data. The TF and RM standard-error results differ but are shown to be asymptotically equivalent (theoretically the same in large samples). In section 4, I consider estimation of the AIE of smoking during pregnancy on infant birthweight using a model akin to that of Mullahy (1997). Because of the endogeneity of the smoking variable, unconventional regression methods are required, for which `margins` is either unavailable or incorrect. In this example, one must revert to the TF to calculate standard errors. In section 5, I summarize and conclude the article.

2 The availability of the Stata `margins` command

This section focuses on the necessary condition (as reflected in Q1) for the appropriate use of `margins` in the present context—availability. `margins` may be used after most of Stata’s packaged estimation commands to estimate an ATE, AME, or AIE. However, there are important cases in which `margins` is not available. For example, attempting to invoke `margins` with the `biprobit` command (bivariate probit) will produce the following error message:

```
"default prediction is a function of possibly stochastic quantities other than  
> e(b)"
```

Moreover, `margins` cannot be used in cases where the asymptotic covariance matrix of the underlying parameter estimation protocol requires community-contributed programming code (for example, covariance matrices calculated via community-contributed

1. Our focus on causal estimation and inference does not substantially limit the generality of the results presented below, because we can think of no interesting empirical applications of TF or RM in empirical econometrics that would not aim to produce causally interpretable results.

Mata code), such as two-stage estimation protocols (Terza 2016a). A prominent example is the two-stage residual inclusion (2SRI) estimator (see Terza, Basu, and Rathouz [2008] and Terza [Forthcoming]); although 2SRI parameter estimates can be obtained using packaged Stata commands that would otherwise permit `margins`, the estimated covariance matrices output by those commands would be incorrect and therefore should not be used as inputs to the standard-error calculations in `margins`.² As Terza (Forthcoming) shows, estimation of the correct asymptotic covariance matrix of the 2SRI estimator is one such case that requires special programming. There are, of course, many similar cases.

3 Calculation of the asymptotically correct standard errors of causal effects estimators

Statistics aimed at estimating causal effects of interest often use the following general form,

$$\hat{\gamma} = \sum_{i=1}^N \frac{g(\hat{\boldsymbol{\theta}}, \mathbf{X}_i)}{N} \quad (1)$$

where $\gamma = E\{g(\boldsymbol{\theta}, \mathbf{X})\}$ is the parameter of ultimate interest to be estimated by (1), $g(\cdot)$ is a known (possibly nonlinear) transformation, $\hat{\boldsymbol{\theta}}$ is a preestimate of $\boldsymbol{\theta}$ —a vector of “deeper” model parameters—and \mathbf{X}_i denotes a vector of observed data on \mathbf{X} for the i th member of a sample of size n ($i = 1, \dots, N$). The three most common versions of (1)—ATE, AME, and AIE—correspond to the following formulations of $g(\cdot)$, respectively,

$$g(\boldsymbol{\theta}, \mathbf{X}) = m(\boldsymbol{\theta}, \mathbf{1}, \mathbf{X}_o) - m(\boldsymbol{\theta}, \mathbf{0}, \mathbf{X}_o) \quad (2)$$

$$g(\boldsymbol{\theta}, \mathbf{X}) = \frac{\partial m(\boldsymbol{\theta}, \mathbf{X}_p, \mathbf{X}_o)}{\partial \mathbf{X}_p} \quad (3)$$

$$g(\boldsymbol{\theta}, \mathbf{X}) = m(\boldsymbol{\theta}, \mathbf{X}_p + \boldsymbol{\Delta}, \mathbf{X}_o) - m(\boldsymbol{\theta}, \mathbf{X}_p, \mathbf{X}_o) \quad (4)$$

where $m(\boldsymbol{\theta}, \mathbf{X}_p, \mathbf{X}_o) = E(\mathbf{Y}|\mathbf{X}_p, \mathbf{X}_o)$ is a regression function written to highlight the distinction between a policy-relevant regressor of interest, \mathbf{X}_p , and a vector of regression controls, \mathbf{X}_o ; $\mathbf{X} = [\mathbf{X}_p \ \mathbf{X}_o]$; $\boldsymbol{\theta}$ is a vector of regression parameters; and $\boldsymbol{\Delta}$ is a known exogenous (usually policy-driven) increment to \mathbf{X}_p . After the regression parameter estimates are obtained (for example, $\hat{\boldsymbol{\theta}}$ —estimated via the nonlinear least [NLS] method), under fairly general conditions, in conjunction with (1), the formulations in (2), (3), and (4), respectively, yield consistent estimators of the ATE when \mathbf{X}_p is binary; the AME when \mathbf{X}_p is continuous and interest is in the effect attributable to an infinitesimal policy change; and the AIE when \mathbf{X}_p is discrete or continuous and the relevant policy increment is $\boldsymbol{\Delta}$. The focus of this article is not only estimation of these causal effects but also attendant inference, usually drawn from the value of a “ t statistic” for (1) as derived from standard asymptotic theory. Such a t statistic has the following general form,

² In section 4, I give an example of such an incorrect covariance matrix in the 2SRI context.

$$\frac{\sqrt{N}(\hat{\gamma} - \gamma^\dagger)}{\text{se}(\hat{\gamma})} \quad (5)$$

where γ^\dagger is the relevant “null” value of γ (as in a test of the null hypothesis $H_0 : \gamma = \gamma^\dagger$) and $\text{se}(\hat{\gamma})$ is the asymptotic standard error of (1) defined as $\text{se}(\hat{\gamma}) \equiv \sqrt{\widehat{\text{a var}}(\hat{\gamma})}$, with $\widehat{\text{a var}}(\hat{\gamma})$ being a consistent estimator of the asymptotic variance of $\hat{\gamma}$. Under slightly stronger conditions than those required for the consistency of (1), it can be shown that (5) is asymptotically standard normal distributed.

The key to using (5) for inference in this context is correct specification of $\text{se}(\hat{\gamma})$ [that is, correct specification of $\widehat{\text{a var}}(\hat{\gamma})$]. Terza (2016b) shows that

$$\widehat{\text{a var}}(\hat{\gamma}) = A + B \quad (6)$$

where

$$A = \left\{ \frac{\sum_{i=1}^N \nabla_{\theta} g(\hat{\theta}, \mathbf{X}_i)}{N} \right\} \widehat{\text{AVAR}}(\hat{\theta}) \left\{ \frac{\sum_{i=1}^N \nabla_{\theta} g(\hat{\theta}, \mathbf{X}_i)}{N} \right\}' \quad (7)$$

$$B = \frac{\sum_{i=1}^N \{g(\hat{\theta}, \mathbf{X}_i) - \hat{\gamma}\}^2}{N} \quad (8)$$

$\nabla_{\theta} g(\hat{\theta}, \mathbf{X}_i)$ (a row vector) denotes the gradient of $g(\theta, \mathbf{X})$ evaluated at \mathbf{X}_i and $\hat{\theta}$, and $\widehat{\text{AVAR}}(\hat{\theta})$ is an estimator of the asymptotic covariance matrix of $\hat{\theta}$. Equations (1) through (8) and their supplemental discussion constitute what we referred to above (in Q2) as the TF.

As an example, and for the purpose of comparing results obtained from the TF with those produced via `margins`, we consider the following model of the likelihood of employment. Let \mathbf{Y} be the indicator of an individual’s employment status such that

$$\mathbf{Y}(\text{employed}) \equiv 1 \text{ if the individual is employed, } 0 \text{ if not}$$

and

$$\mathbf{X}_p \equiv \text{the particular employment determinant of interest}$$

We will, in turn, estimate the causal effect of each of the following three employment determinants using the relevant versions of the statistic given in (1).

ATE with (1) defined as in (2)

$$\mathbf{X}_{p1}(\text{disabil}) \equiv 1 \text{ if the individual has a disability, } 0 \text{ if not (binary variable)} \quad (9)$$

AME with (1) defined as in (3)

$$\begin{aligned} \mathbf{X}_{p2}(\text{othhinc}) &\equiv \text{other household income} \\ &= \text{income earned by others in the household} \\ &\text{(continuous variable)} \end{aligned} \quad (10)$$

and

AIE with (1) defined as in (4) with $\Delta = 1$

$$\mathbf{X}_{p3}(\text{exper}) \equiv \text{years of work experience} = \text{age} - \text{grade} - 6 \text{(count variable)} \quad (11)$$

The elements of the vector of additional control variables (\mathbf{X}_o) are³

male \equiv 1 if male, 0 if not
black \equiv 1 if black, 0 if not
grade \equiv years of schooling completed
inschool \equiv 1 if enrolled in school in 1980, 0 if not
vet \equiv 1 if veteran of military service, 0 if not
neverm \equiv 1 if never married, 0 otherwise
income4 \equiv interest, dividend, and rental income
worlft75 \equiv 1 if worked less than full time in 1975, 0 if not
spanish \equiv 1 if of hispanic descent, 0 if not
indian \equiv 1 if Native American, 0 if not
foreignb \equiv 1 if foreign born, 0 if not
nonengl \equiv 1 if speaks English poorly or not at all, 0 if not
smsa \equiv 1 if resides in a Standard Metropolitan Area, 0 if not
regso \equiv 1 if resides in Southern Census Region, 0 if not
regwe \equiv 1 if resides in Western Census Region, 0 if not
regnc \equiv 1 if resides in North Central Census Region, 0 if not
evermus \equiv 1 if current state differs from state of birth, 0 if not
npershh \equiv number of persons in the household
othhinc \equiv income earned by others in the household

3. In each case, the nonfeatured causal variables will be included among the elements of \mathbf{X}_o (for example, when estimating the ATE of **disabil**, **exper** and **othhinc** will be included in \mathbf{X}_o).

The data for this illustration were drawn from the database analyzed by Fishback and Terza (1989). The analysis sample comprises 31,507 observations. The descriptive statistics of the analysis sample are given in table 1.

Table 1. Descriptive statistics

Variable	Mean	Min	Max
employed	0.68	0.00	1.00
disabil	0.10	0.00	1.00
exper	30.41	9.00	59.00
othhinc	13231.87	0.00	74839.90
male	0.47	0.00	1.00
black	0.07	0.00	1.00
grade	12.01	0.00	20.00
inschool	0.01	0.00	1.00
vet	0.27	0.00	1.00
neverm	0.05	0.00	1.00
income4	543.80	0.00	62005.00
worlft75	0.37	0.00	1.00
spanish	0.04	0.00	1.00
indian	0.00	0.00	1.00
foreignb	0.07	0.00	1.00
nonlengl	0.02	0.00	1.00
smsa	0.81	0.00	1.00
regso	0.30	0.00	1.00
regwe	0.18	0.00	1.00
regnc	0.27	0.00	1.00
evermus	0.37	0.00	1.00
npershh	3.40	1.00	16.00

We assume a probit regression specification for the employment outcome, so the parameters of the model can be estimated using the Stata `probit` command—for which `margins` is available. Using the TF, we estimated the ATE, AME, and AIE as formalized in (2)–(4) and detailed in (9)–(11), respectively.

For the purpose of illustration, consider the formulation and Stata coding of the TF standard errors and asymptotic t statistics for the estimated ATE of disability. In this example, the estimated causal effect of interest is given by the following versions of (1) and (2),

$$\hat{\gamma} = \sum_{i=1}^N \frac{g(\hat{\beta}, \mathbf{X}_i)}{N}$$

with

$$g(\beta, \mathbf{X}_p, \mathbf{X}_o) = m(\beta, \mathbf{1}, \mathbf{X}_o) - m(\beta, \mathbf{0}, \mathbf{X}_o)$$

where $\hat{\beta}' = [\hat{\beta}_p \ \hat{\beta}'_o]$ is the probit estimate of $\beta' = [\beta_p \ \beta'_o]$,

$$m(\theta, \mathbf{X}_p, \mathbf{X}_o) = \Phi(\mathbf{X}_p\beta_p + \mathbf{X}_o\beta_o)$$

and $\Phi(\cdot)$ denotes the standard normal cumulative distribution function. The asymptotically correct standard error is given in (6) with

$$A = \left\{ \frac{\sum_{i=1}^N \nabla_{\beta} g(\hat{\beta}, \mathbf{X}_i)}{N} \right\} \widehat{\text{AVAR}}(\hat{\beta}) \left\{ \frac{\sum_{i=1}^N \nabla_{\beta} g(\hat{\beta}, \mathbf{X}_i)}{N} \right\}' \tag{12}$$

$$B = \frac{\sum_{i=1}^N \{g(\hat{\beta}, \mathbf{X}_i) - \hat{\gamma}\}^2}{N} \tag{13}$$

and

$$\nabla_{\beta} g(\beta, \mathbf{X}) = \varphi(\mathbf{X}^1\beta)\mathbf{X}^1 - \varphi(\mathbf{X}^0\beta)\mathbf{X}^0 \tag{14}$$

where $\mathbf{X}^1 = [\mathbf{1} \ \mathbf{X}_o]$ and $\mathbf{X}^0 = [\mathbf{0} \ \mathbf{X}_o]$. The Stata and Mata do-file used to obtain the estimate of the ATE of disability on employment and calculate its TF-based asymptotically correct standard error using (12)–(14) is

```

/*****
** This do-file estimates ATE of disability on
** employment, along with its correct asymptotic
** standard error.
** Stata probit procedure used to fit the model.
** Fishback-Terza data are used.
** Program implements Mata, not margins.
*****/
/*****
** Preliminary Stuff.
*****/
clear mata
clear matrix
clear
set more off
capture log close

/*****
** Set the default directory.
*****/
cd <PATH FOR THE DEFAULT DIRECTORY>

/*****
** Set up the output file.
*****/
log using ATE-Simple-disabil.log, replace
    
```

```

/*****
** Read in the data.
*****/
use fishback-terza-male-female-data-clean.dta

/*****
** Compute descriptive statistics.
*****/
summarize employed disabil exper othhinc male black grade ///
inschool vet neverm income4 worlft75 spanish indian ///
foreignb nonlengl smsa regso regwe regnc ///
evermus npersh

/*****
** Probit.
*****/
probit employed disabil male black grade inschool ///
vet neverm income4 worlft75 spanish indian foreignb ///
nonlengl smsa regso regwe regnc evermus npersh exper othhinc

/*****
** Start mata.
*****/
mata:

/*****
** Save probit coefficient estimates and covariance
** matrix estimate.
*****/
beta=st_matrix("e(b)")
AVARbeta=st_matrix("e(V)")

/*****
** End mata.
*****/
end

/*****
** Post data into mata.
*****/
putmata employed disabil neverm male grade exper ///
vet worlft75 foreignb nonlengl black indian ///
spanish regnc regso regwe smsa inschool evermus ///
npersh othhinc income4

/*****
** MATA Start-up.
*****/
mata:

/*****
** Sample size.
*****/
N=rows(employed)

```

```

/*****
** Define the matrix of regressors, including
** Xp, Xo and a constant term.
*****/
X=disabil, male, black, grade, inschool, vet, neverm,   ///
  income4, worlft75, spanish, indian, foreignb,       ///
  nonlengl, smsa, regso, regwe, regnc, evermus,       ///
  npershh, exper, othhinc, J(N,1,1)

/*****
** Create the X sup 1 vector (Xp=1).
*****/
Xsup1=X
Xsup1[. ,1]=J(N,1,1)

/*****
** Create the X sup 0 vector (Xp=0).
*****/
Xsup0=X
Xsup0[. ,1]=J(N,1,0)

/*****
** Compute the index vector for (Xp=1).
*****/
Xsup1Beta=Xsup1*beta

/*****
** Compute the index vector for (Xp=0).
*****/
Xsup0Beta=Xsup0*beta

/*****
** Compute the estimated effect for each
** individual in the sample.
*****/
g=normal(Xsup1Beta)-normal(Xsup0Beta)

/*****
** Compute the average treatment effect.
*****/
ATE=mean(g)

/*****
** Compute the gradient of g with respect
** to beta.
*****/
pbetag=normalden(Xsup1Beta):*Xsup1:-normalden(Xsup0Beta):*Xsup0

/*****
** Average the gradient of g with respect to beta.
*****/
pbetag=mean(pbetag)

/*****
** Compute the estimated asymptotic variance.
*****/
varATE=pbetag*(N:*(AVARbeta))*pbetag'+mean((g:-ATE):^2)

```

```

/*****
** The corresponding standard error.
*****/
seATE=sqrt(varATE/N)

/*****
** Compute the relevant asymptotic t-statistic.
*****/
tstatATE=ATE/sqrt(varATE/N)

/*****
** Compute the corresponding p-value.
*****/
pvalue=2*(1:-normal(abs(tstatATE)))

/*****
** Display effect results obtained via Mata.
*****/
header="ATE","asy-se","asy-t-stat","p-value" \ "","",""
results=ATE,seATE,tstatATE,pvalue
resview=stofreal(results)
"disabil"
header \ resview

/*****
** MATA end.
*****/
end

log close
```

The relevant output from this do-file is

```
. probit employed disabil male black grade inschool
> vet neverm income4 worlft75 spanish indian foreignb
> nonlengl smsa regso regwe regnc evermus npershh exper othhinc
Iteration 0: log likelihood = -19823.769
Iteration 1: log likelihood = -14122.42
Iteration 2: log likelihood = -14057.314
Iteration 3: log likelihood = -14057.182
Iteration 4: log likelihood = -14057.182

Probit regression                               Number of obs   =    31,507
                                                LR chi2(21)    =   11533.17
                                                Prob > chi2    =    0.0000
                                                Pseudo R2     =    0.2909

Log likelihood = -14057.182
```

employed	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
disabil	-.7658884	.0284741	-26.90	0.000	-.8216966	-.7100803
male	-.0182789	.0264679	-0.69	0.490	-.0701551	.0335973
black	.1075914	.0351067	3.06	0.002	.0387836	.1763992
grade	-.0024472	.0034459	-0.71	0.478	-.009201	.0043066
inschool	.3675846	.0800549	4.59	0.000	.2106799	.5244893
vet	.1636881	.0269121	6.08	0.000	.1109414	.2164349
neverm	.0455236	.0441799	1.03	0.303	-.0410674	.1321145
income4	-.000053	4.29e-06	-12.35	0.000	-.0000614	-.0000446
worlft75	-1.396906	.0209017	-66.83	0.000	-1.437872	-1.355939
spanish	-.0472764	.0467021	-1.01	0.311	-.1388108	.044258
indian	.0268261	.1385675	0.19	0.846	-.2447613	.2984135
foreignb	-.0268723	.0379451	-0.71	0.479	-.1012433	.0474986
nonlengl	.0947007	.0685965	1.38	0.167	-.0397459	.2291473
smsa	.1789895	.0217065	8.25	0.000	.1364455	.2215335
regso	-.0358343	.023932	-1.50	0.134	-.0827402	.0110717
regwe	-.0157091	.0275966	-0.57	0.569	-.0697974	.0383792
regnc	.0244632	.0240862	1.02	0.310	-.022745	.0716714
evermus	.0064609	.0192114	0.34	0.737	-.0311928	.0441146
npershh	-.0137427	.0062981	-2.18	0.029	-.0260868	-.0013986
exper	-.0275907	.0011352	-24.30	0.000	-.0298157	-.0253657
othhinc	-5.72e-06	7.64e-07	-7.48	0.000	-7.21e-06	-4.22e-06
_cons	2.015496	.081264	24.80	0.000	1.856222	2.17477

```
: header \ resview
```

	1	2	3	4
1	ATE	asy-se	asy-t-stat	p-value
2				
3	-.216556	.0086262	-25.10448	0

The ATE estimate is displayed in the header over the first two columns of table 2, and its TF-based asymptotic standard error and *t* statistic are shown in the first two columns of the first row of that table. The analogous TF for the AME of household income and the AIE of experience were derived and coded in Stata and Mata. The effect estimates are given in the respective headers of table 2, and the TF-based standard errors and *t* statistics are correspondingly displayed in the first row of that table.⁴

4. The full do-files, available from the *Stata Journal* software package, are given in appendix A.

Table 2. ATE, AME, and AIE estimates and their standard errors

Standard-error calculation method	Disability $\hat{\gamma} = \widehat{ATE} = 0.217$		Other household income $\hat{\gamma} = \widehat{AME} = 1.43e-06$		Experience $\hat{\gamma} = \widehat{AIE} = 0.007$	
	Asymptotic standard error	Asymptotic <i>t</i> statistic	Asymptotic standard error	Asymptotic <i>t</i> statistic	Asymptotic standard error	Asymptotic <i>t</i> statistic
TF	0.0086262	-25.10	1.91e-07	-7.49	0.0002813	-24.64
RM	0.0086215	-25.12	1.88e-07	-7.61	0.0002831	-24.48
ATF	0.0086215	-25.12	1.88e-07	-7.61	0.0002831	-24.48

Alternatively, the ATE, AME, and AIE estimates and their standard errors (asymptotic *t* statistics) can be obtained via `margins`. Note that there are two options in `margins` for calculating standard errors: a) `vce(delta)`, which is only appropriate for cases in which the matrix of observations on the independent variables (say, χ) is assumed to be fixed in repeated samples; and b) `vce(unconditional)`, which is used in all other cases. Generally, χ is not assumed to be fixed in repeated samples.⁵ Therefore, the `vce(unconditional)` option is the most relevant to the current discussion. [StataCorp \(2017, 1455–1466\)](#) describes the three cases in which the `vce(unconditional)` option is applicable: case I—you have a representative sample and have not `svyset` your data; case II—you have a weighted sample and have not `svyset` your data; and case III—you have `svyset` your data. Case I is relevant here. In this case, the `margins` command for the ATE and AME [as defined in (2) and (3)] is

```
margins, dydx(varname) vce(unconditional) (a)
```

and for the AIE [as defined in (4)] with $\Delta = 1$, the proper use of `margins` is

```
margins, at((asobserved)_all) ///
at(varname=generate(varname+1)) ///
contrast(atcontrast(r)) vce(unconditional) (b)
```

where *varname* is the name of the causal variable. The Stata statements in (a) and (b) constitute what we referred to above (in Q2) as the RM. We estimated the ATE (disability), AME (other household income), and AIE (experience) using the RM in (a) and (b), accordingly. For instance, the Stata do-file used to estimate the ATE of disability on employment using the RM is⁶

5. [Dowd, Greene, and Norton \(2014\)](#) derive standard-error formulations under the fixed-in-repeated-samples assumption. This is a very restrictive and unrealistic assumption that is usually invalid for empirical econometrics.

6. Note that, to invoke the `vce(unconditional)` option in `margins`, you must use the `vce(robust)` option in the relevant Stata estimation command.

```

/*****
** This program estimates ATE of disability on
** employment, along with its correct asymptotic
** standard error.
** Stata probit procedure used to fit the model.
** Fishback-Terza data are used.
** Program implements margins, not Mata.
*****/
/*****
** Preliminary Stuff.
*****/
clear mata
clear matrix
clear
set more off
capture log close

/*****
** Set the default directory.
*****/
cd <PATH FOR THE DEFAULT DIRECTORY>

/*****
** Set up the output file.
*****/
log using ATE-MARGINS-disabil.log, replace

/*****
** Read in the data (Full dataset -- Males and
** Females).
*****/
use fishback-terza-male-female-data-clean.dta

/*****
** Compute descriptive statistics.
*****/
summarize employed disabil exper othhinc male ///
black grade inschool vet neverm ///
income4 worlft75 spanish indian foreignb ///
nonlengl smsa regso regwe regnc ///
evermus npershh

/*****
** Probit.
*****/
probit employed i.disabil male ///
black grade inschool vet neverm ///
income4 worlft75 spanish indian foreignb ///
nonlengl smsa regso regwe regnc ///
evermus npershh exper othhinc, vce(robust)

/*****
** Calculate and store the margins results.
*****/
margins, dydx(disabil) vce(unconditional)

log close

```

The relevant output from this do-file is

```
. probit employed i.disabil male
> black grade inschool vet neverm
> income4 worlft75 spanish indian foreignb
> nonlengl smsa regso regwe regnc
> evermus npershh exper othhinc, vce(robust)

Iteration 0:  log pseudolikelihood = -19823.769
Iteration 1:  log pseudolikelihood = -14122.42
Iteration 2:  log pseudolikelihood = -14057.314
Iteration 3:  log pseudolikelihood = -14057.182
Iteration 4:  log pseudolikelihood = -14057.182

Probit regression                               Number of obs   =    31,507
                                                Wald chi2(21)   =    9046.46
                                                Prob > chi2     =    0.0000
                                                Pseudo R2      =    0.2909

Log pseudolikelihood = -14057.182
```

employed	Robust					[95% Conf. Interval]	
	Coef.	Std. Err.	z	P> z			
1.disabil	-.7658884	.0284655	-26.91	0.000	-.8216797	-.7100971	
male	-.0182789	.0265917	-0.69	0.492	-.0703978	.0338399	
black	.1075914	.0324009	3.32	0.001	.0440868	.171096	
grade	-.0024472	.0034933	-0.70	0.484	-.0092939	.0043994	
inschool	.3675846	.0826749	4.45	0.000	.2055448	.5296244	
vet	.1636881	.0277747	5.89	0.000	.1092507	.2181255	
neverm	.0455236	.0432359	1.05	0.292	-.0392172	.1302643	
income4	-.000053	4.80e-06	-11.03	0.000	-.0000624	-.0000436	
worlft75	-1.396906	.0203743	-68.56	0.000	-1.436839	-1.356973	
spanish	-.0472764	.0466468	-1.01	0.311	-.1387024	.0441496	
indian	.0268261	.1193102	0.22	0.822	-.2070177	.2606699	
foreignb	-.0268723	.0392518	-0.68	0.494	-.1038044	.0500597	
nonlengl	.0947007	.0667308	1.42	0.156	-.0360892	.2254906	
smsa	.1789895	.0220443	8.12	0.000	.1357835	.2221955	
regso	-.0358343	.02374	-1.51	0.131	-.0823639	.0106953	
regwe	-.0157091	.0280164	-0.56	0.575	-.0706202	.0392202	
regnc	.0244632	.0240745	1.02	0.310	-.0227219	.0716483	
evermus	.0064609	.0191481	0.34	0.736	-.0310687	.0439905	
npershh	-.0137427	.0062653	-2.19	0.028	-.0260224	-.001463	
exper	-.0275907	.0011489	-24.01	0.000	-.0298425	-.0253389	
othhinc	-5.72e-06	7.54e-07	-7.58	0.000	-7.19e-06	-4.24e-06	
_cons	2.015496	.0820759	24.56	0.000	1.85463	2.176362	

```
. /*****
> ** Calculate and store the margins results.
> *****/
. margins, dydx(disabil) vce(unconditional)
```

```
Average marginal effects                               Number of obs   =    31,507
Expression      : Pr(employed), predict()
dy/dx w.r.t.   : 1.disabil
```

1.disabil	Unconditional					[95% Conf. Interval]	
	dy/dx	Std. Err.	z	P> z			
1.disabil	-.216556	.0086215	-25.12	0.000	-.2334538	-.1996582	

Note: dy/dx for factor levels is the discrete change from the base level.

We first note that the ATE estimate produced by this code is identical to that which was obtained earlier via the above Mata code (shown in the header above the first two columns of table 2). The results for the RM-based asymptotic standard error and t statistic for the ATE, displayed in the second row (first two columns) of table 2, differ from the TF-based results (albeit slightly). Similar RM-based calculations for the AME of household income and the AIE of experience were conducted.⁷ As was the case for the ATE, the corresponding estimates of the AME and AIE are identical to those obtained using the TF (they are given in the respective headers of table 2). RM-based standard errors and t statistics are correspondingly displayed in the first row of that table. They also differ slightly from the TF-based values.

We suspected that, asymptotically, the difference between the TF and RM results is nil. To verify our conjecture, we note, as the appendix available with Terza (2016b) shows, that the following asymptotic variance estimator is asymptotically equivalent to (6),

$$\widehat{\text{a var}}(\hat{\gamma}) = A + B + 2C \quad (15)$$

where the scalar C is a function of $\hat{\theta}$, \mathbf{X}_i , and $\hat{\gamma}$. The asymptotic variance estimators in (6) and (15) are asymptotically equivalent in that C converges to zero in the limit as N approaches ∞ . To assess our conjecture empirically, we re-estimated the standard errors of the three effect parameters (and their asymptotic t statistics) using (15). We call this the adjusted TF (ATF) approach to estimating the standard errors.⁸ The ATF and RM results are identical. From this, we conclude that RM should be used when the `margins` command is available and correct.

4 A case for which margins is unavailable or incorrect: Causal inference in the 2SRI context

Consider the regression model of Mullahy (1997), in which the objective is to draw causal inferences regarding the effect of prenatal smoking (\mathbf{X}_p -CIGSPREG) on infant birthweight (Y -BIRTHWTLB), while controlling for infant birth order (PARITY), race (WHITE), and sex (MALE). The regression model for the birthweight outcome that he proposed can be written as

$$\mathbf{Y} = \exp(\mathbf{X}_p\boldsymbol{\beta}_p + \mathbf{X}_o\boldsymbol{\beta}_o + \mathbf{X}_u\boldsymbol{\beta}_u) + \mathbf{e} \quad (16)$$

where \mathbf{X}_u comprises unobservable variables that are potentially correlated with prenatal smoking (for example, general “health mindedness” of the mother), \mathbf{e} is the regression error term, $\mathbf{X}_o = [\text{PARITY WHITE MALE}]$ is a row vector of regressors that are uncorrelated with \mathbf{X}_u , and \mathbf{e} and the $\boldsymbol{\beta}$'s are the regression parameters.⁹ At issue here is the fact that there exist unobservables (as captured by \mathbf{X}_u) that are correlated with both \mathbf{Y} and \mathbf{X}_p . In other words, \mathbf{X}_p is endogenous. To account for said endogeneity, we

7. The full do-files, available from the *Stata Journal* software package, are given in appendix B.

8. The full do-files used to obtain the ATF estimates and their asymptotically correct standard errors, available from the *Stata Journal* software package, are given in appendix C.

9. Mullahy (1997) does not explicitly specify the model in terms of the unobservable \mathbf{X}_u . Nevertheless, (16) is substantively identical to Mullahy's model (see Terza [2006]).

follow the 2SRI approach suggested by Terza, Basu, and Rathouz (2008) and formalize the correlation between \mathbf{X}_u and \mathbf{X}_p as

$$\mathbf{X}_p = \exp(\mathbf{W}\boldsymbol{\alpha}) + \mathbf{X}_u$$

where $\boldsymbol{\alpha}$ is a vector of regression parameters $\mathbf{W} = [\mathbf{X}_o \mathbf{W}^+]$ and \mathbf{W}^+ is a vector of identifying instrumental variables specified in this case as

$$\mathbf{W}^+ = [\text{EDFATHER EDMOTHER FAMINCOME CIGTAX}]$$

with

$$\begin{aligned} \text{EDFATHER} &\equiv \text{paternal schooling in years} \\ \text{EDMOTHER} &\equiv \text{maternal schooling in years} \\ \text{FAMINCOME} &\equiv \text{family income} \end{aligned}$$

and

$$\text{CIGTAX} \equiv \text{cigarette tax}$$

Suppose that the ultimate objective here is estimation of the causal effect of a policy that completely prevents and eliminates smoking during pregnancy. In this example, given a consistent estimate of $\boldsymbol{\theta}' = [\boldsymbol{\alpha}' \boldsymbol{\beta}']$ (say, $\hat{\boldsymbol{\theta}}$) with $\boldsymbol{\beta}' = [\boldsymbol{\beta}_p \boldsymbol{\beta}_o \boldsymbol{\beta}_u']$, the estimated causal effect of interest is given by the following versions of (1) and (4),

$$\hat{\gamma} = \sum_{i=1}^N \frac{g(\hat{\boldsymbol{\theta}}, \mathbf{Z}_i)}{N} \quad (17)$$

with

$$g(\hat{\boldsymbol{\theta}}, \mathbf{Z}_i) = m(\hat{\boldsymbol{\theta}}, \mathbf{X}_{pi} + \boldsymbol{\Delta}_i, \mathbf{W}_i) - m(\hat{\boldsymbol{\theta}}, \mathbf{X}_{pi}, \mathbf{W}_i) \quad (18)$$

where $\boldsymbol{\Delta} = \boldsymbol{\Delta}_i = -\mathbf{X}_{pi}$, $\mathbf{Z}_i = [\mathbf{X}_{pi} \mathbf{W}_i]$ and

$$m(\hat{\boldsymbol{\theta}}, \mathbf{X}_{pi}, \mathbf{W}_i) = \exp[\mathbf{X}_{pi}\hat{\boldsymbol{\beta}}_p + \mathbf{X}_{oi}\hat{\boldsymbol{\beta}}_o + \{\mathbf{X}_{pi} - \exp(\mathbf{W}_i\hat{\boldsymbol{\alpha}})\}\hat{\boldsymbol{\beta}}_u]$$

The asymptotically correct standard error is obtained from (6), with

$$A = \left\{ \frac{\sum_{i=1}^N \nabla_{\boldsymbol{\theta}} g(\hat{\boldsymbol{\theta}}, \mathbf{Z}_i)}{N} \right\} \widehat{\text{AVAR}}(\hat{\boldsymbol{\theta}}) \left\{ \frac{\sum_{i=1}^N \nabla_{\boldsymbol{\theta}} g(\hat{\boldsymbol{\theta}}, \mathbf{Z}_i)}{N} \right\}' \quad (19)$$

$$B = \frac{\sum_{i=1}^N \{g(\hat{\boldsymbol{\theta}}, \mathbf{Z}_i) - \hat{\gamma}\}^2}{N} \quad (20)$$

and

$$\nabla_{\boldsymbol{\theta}} g(\hat{\boldsymbol{\theta}}, \mathbf{Z}_i) = \left\{ \nabla_{\boldsymbol{\alpha}} g(\hat{\boldsymbol{\theta}}, \mathbf{Z}_i) \quad \nabla_{\boldsymbol{\beta}} g(\hat{\boldsymbol{\theta}}, \mathbf{Z}_i) \right\} \quad (21)$$

where $\mathbf{Z}_i = [\mathbf{X}_{pi} \ \mathbf{W}_i]$ and $\mathbf{W}_i = [\mathbf{X}_{oi} \ \mathbf{W}_i^+]$

$$\begin{aligned} \nabla_{\alpha} g(\hat{\boldsymbol{\theta}}, \mathbf{Z}_i) &= -\hat{\beta}_u \exp(\mathbf{W}_i \hat{\boldsymbol{\alpha}}) g(\hat{\boldsymbol{\theta}}, \mathbf{Z}_i) \mathbf{W}_i \\ \nabla_{\beta} g(\hat{\boldsymbol{\theta}}, \mathbf{Z}_i) &= \exp(\mathbf{V}_i^{\Delta} \hat{\boldsymbol{\beta}}) \mathbf{V}_i^{\Delta} - \exp(\mathbf{V}_i \boldsymbol{\beta}) \mathbf{V}_i \\ \mathbf{V}_i &= [\mathbf{X}_{pi} \ \mathbf{X}_{oi} \ \{\mathbf{X}_{pi} - \exp(\mathbf{W}_i \hat{\boldsymbol{\alpha}})\}] \end{aligned}$$

and

$$\mathbf{V}_i^{\Delta} = [\mathbf{X}_{pi} + \boldsymbol{\Delta} \ \mathbf{X}_{oi} \ \{\mathbf{X}_{pi} - \exp(\mathbf{W}_i \hat{\boldsymbol{\alpha}})\}]$$

We first estimated $\boldsymbol{\theta}' = [\boldsymbol{\alpha}' \ \boldsymbol{\beta}']$ using the 2SRI method as detailed in section 4 of Terza (2016a). The first- and second-stage 2SRI results are shown in tables 1 and 2 of that article. We then used the 2SRI result $(\hat{\boldsymbol{\theta}})$ and (17) to obtain a consistent estimate of the desired AIE. The relevant asymptotic standard error and t statistic were calculated using (6) and (19)–(21).

Note that (19) requires the correct expression for $\widehat{\text{AVAR}}(\boldsymbol{\theta})$, and this cannot be obtained directly from the outputs from the `glm` commands used to produce consistent estimates of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. For instance, the covariance matrix estimate of second-stage `glm` for $\boldsymbol{\beta}$ would be consistent for

$$E(\nabla_{\beta\beta} q)^{-1} E(\nabla_{\beta} q' \nabla_{\beta} q) E(\nabla_{\beta\beta} q)^{-1} \tag{22}$$

where $\nabla_{\beta} q$ and $\nabla_{\beta\beta} q$ are shorthand notation for the gradient and Hessian of

$$q(\boldsymbol{\theta}, \mathbf{X}_p, \mathbf{W}) = -(\mathbf{Y} - \exp[\mathbf{X}_p \boldsymbol{\beta}_p + \mathbf{X}_o \boldsymbol{\beta}_o + \{\mathbf{X}_p - \exp(\mathbf{W}\boldsymbol{\alpha})\} \boldsymbol{\beta}_u])^2$$

respectively. But (22) is incomplete—an additional term is required to account for the fact that the estimator has two stages (see Terza [2016a]). It is this incorrect covariance matrix estimate that would be used by `margins` in calculating the standard error of the AIE. Therefore, `margins` cannot be used in this case. Following Terza (2016a), we have that the correct estimated asymptotic covariance matrix of $\hat{\boldsymbol{\theta}}$ (the 2SRI estimator of $\boldsymbol{\theta}$) is

$$\widehat{\text{AVAR}}(\hat{\boldsymbol{\theta}}) = \begin{bmatrix} N \times \widehat{\text{AVAR}}^*(\hat{\boldsymbol{\alpha}}) & \hat{\mathbf{D}}_{12} \\ \hat{\mathbf{D}}'_{12} & N \times \hat{\mathbf{D}}_{22}^{\dagger} \end{bmatrix}$$

where $\widehat{\text{AVAR}}^*(\hat{\boldsymbol{\alpha}})$ is the estimated covariance matrix output by `glm` for the first-stage estimate of $\hat{\boldsymbol{\alpha}}$, and $\hat{\mathbf{D}}_{12}$ and $\hat{\mathbf{D}}_{22}^{\dagger}$ are as defined in (22) and (26) of Terza (2016a), respectively.

The key Stata and Mata code for calculating the AIE estimate, its correct standard error, and its t statistic is

```

/*****
** Purpose: Mullahy (1997) Birth Weight model.
** Estimation of the model using the 2SRI.
** The outcome variable (birth weight) is
** non-negative and continuous, and the policy
** variable (cigarette smoking) is nonnegative
** and a count.
*****/

/*****
** Initial Set-up.
*****/
clear mata
clear matrix
clear
set more off
capture log close

/*****
** Set up default directory.
*****/
cd <PATH FOR THE DEFAULT DIRECTORY>

/*****
** Set up the output file.
*****/
log using Mullahy-Birthweight-2SRI.log, replace

/*****
** Read in the data.
*****/
use mullahy-birthweight-data.dta

/*****
** Descriptive Statistics.
*****/
summ

/*****
** First-stage NLS for alphahat and residuals.
*****/
glm CIGSPREG PARITY WHITE MALE EDFATHER EDMOTHER    ///
    FAMINCOM CIGTAX88,                               ///
    family(gaussian) link(log) vce(robust)

/*****
** Save residuals.
*****/
predict Xuhat, response

/*****
** Save alphahat and its covariance matrix.
*****/
mata: alphahat=st_matrix("e(b)")
mata: COValphahat=st_matrix("e(V)")

```

```

/*****
** Second-stage NLS for betahat.
*****/
glm BIRTHWT CIGSPREG PARITY WHITE MALE Xuhat,    ///
    family(gaussian) link(log) vce(robust)

/*****
** Save betahat, its covariance matrix and
** single out the coefficient of Xu.
*****/
mata: betahat=st_matrix("e(b)")
mata: COVbetahat=st_matrix("e(V)")
mata: Bu=betahat[5]

/*****
** Post needed variables to Mata.
*****/
putmata BIRTHWT CIGSPREG PARITY WHITE MALE EDFATHER    ///
    EDMOTHER FAMINCOM CIGTAX88 Xuhat

/*****
** Start mata.
*****/
mata:

/*****
** Set up V and W matrices.
*****/
V=CIGSPREG, PARITY, WHITE, MALE,    ///
    Xuhat, J(rows(PARITY),1,1)
W=PARITY, WHITE, MALE, EDFATHER, EDMOTHER,    ///
    FAMINCOM, CIGTAX88, J(rows(PARITY),1,1)

/*****
** Set up the vector of rhs variable names.
*****/
VNames="CIGSPREG", "PARITY", "WHITE", "MALE",    ///
    "Xuhat", "Constant"

/*****
** Set up Balpha and Bbeta.
*****/
Ba=-Bu:*exp(V*betahat)/*
*/:*exp(W*alphahat):*W
Bb=exp(V*betahat):*V

/*****
** Set Bbetabeta and Bbetaalpha.
*****/
Bbb=Bb`*Bb
Bba=Bb`*Ba

/*****
** Estimate the asymptotic covariance matrix.
*****/
D22hat=invsym(Bbb)*Bba*COValphahat*Bba`*invsym(Bbb)+COVbetahat

```

```

/*****
** Calculate the asymptotically correct
** standard errors.
*****/
ACSE=sqrt(diagonal(D22hat))

/*****
** Calculate the asymptotically correct
** t-stats.
*****/
tstats=betahat:/ACSE

/*****
** Compute the corresponding p-values.
*****/
pvalues=2*(1:-normal(abs(tstats)))

/*****
** Display the results.
*****/
header="Variable","Coeff-Estimate","Std-errs","t-stat","p-value" \ "","","",""
results=betahat,ACSE,tstats,pvalues
resview=stroofreal(results)
header \ (VNAMEs',resview)

/*****
** Computation of the AIE begins here.
*****/

/*****
** Set delta (the policy increment).
*****/
Xp=CIGSPREG
delta=-Xp

/*****
** Calculate AIE.
*****/
Vdelta=V
Vdelta[.,1]=Vdelta[.,1]:+delta
expVdelta=exp(Vdelta*betahat)
expV=exp(V*betahat)
expW=exp(W*alphahat)
g=expVdelta:-expV
AIE=mean(g)

/*****
** Calculate components of estimated asymptotic
** variance of estimated AIE.
*****/
gradthetag=-Bu:*expW:*g:*W,expVdelta:*Vdelta:-expV:*V
N=rows(V)
D11hat=N:*COValphahat
D22hatstar=N:*D22hat
D12hat=-D11hat*Bba`*invsym(Bbb)
Dhat=D11hat,D12hat \ D12hat',D22hatstar

```

```

/*****
** Compute the estimated asymptotic variance.
*****/
avarAIEhat=mean(gradthetag)*Dhat*mean(gradthetag)'/*
*/:+mean((AIE:-g):^2)

/*****
** Compute the corresponding standard error.
*****/
seAIEhat=sqrt(avarAIEhat/N)

/*****
** Compute the relevant asymptotic t statistic.
*****/
tstatavarAIEhat=AIE/seAIEhat

/*****
** Compute the corresponding p-value.
*****/
pvalue=2*(1:-normal(abs(tstatavarAIEhat)))

/*****
** Display effect results obtained via Mata.
*****/
header="AIEhat","asy-se","asy-t-stat","p-value" \ " "," "," "
results=AIE,seAIEhat,tstatavarAIEhat,pvalue
resview=stroofreal(results)
header \ resview

/*****
** End Mata.
*****/
end

log close

```

As shown in the header of table 3, the resultant AIE estimate is 3.68, implying that our hypothetical policy would serve to increase infant birthweight by nearly 4 ounces on average. The asymptotically correct standard errors and t statistics obtained via the TF using (19)–(21) are displayed in the first row of table 3.

Table 3. Smoking during pregnancy ($\widehat{\gamma} = \widehat{\text{AIE}} = 3.68$)

Standard-error calculation method	Asymptotic standard error	Asymptotic t statistic
TF	1.167	3.153
RM	1.046	3.517

For comparison, in table 3, we also give the incorrect results obtained using the following Stata code, which implements the RM in (b):¹⁰

```

/*****
** First-stage NLS for alphahat and residuals.
*****/
glm CIGSPREG PARITY WHITE MALE EDFATHER EDMOTHER    ///
    FAMINCOM CIGTAX88,                               ///
    family(gaussian) link(log) vce(robust)

/*****
** Save residuals.
*****/
predict Xuhat, response

/*****
** Second-stage NLS for alphahat and residuals.
*****/
glm BIRTHWT CIGSPREG PARITY WHITE MALE Xuhat,       ///
    family(gaussian) link(log) vce(robust)

/*****
** Calculate and store the margins results.
*****/
margins, at((asobserved) _all) at(CIGSPREG=generate(0)) ///
contrast(atcontrast(r)) vce(unconditional)

```

This code produces an incorrect value for the standard error of the AIE estimate, because `margins` implements a covariance matrix estimate that is consistent for (22), which ignores first-stage estimation of α and the concomitant requisite covariance matrix correction factor.

5 Summary and conclusion

Terza (2016b) gives the generic expression for the asymptotically correct standard errors of statistics formed as sample means of nonlinear data transformations. In this article, we assessed the performance of `margins` as a relatively simple alternative for calculating such standard errors. We noted that `margins` is not available for all packaged nonlinear

10. The full do-file for the RM estimates, available from the *Stata Journal* software package, is given in appendix D.

regression commands in Stata and cannot be implemented in conjunction with user-defined-and-coded nonlinear estimation protocols that do not make `predict` available. When `margins` is available, however, we established (using a real-data example) that it produces standard errors that are asymptotically equivalent to those obtained from the formulations in Terza (2016b). Given its relative coding simplicity, this result favors using `margins` when available. In all other cases, one can use Mata to code the standard-error formulations in Terza (2016b) [namely, (6) through (8) above].

6 References

- Dowd, B. E., W. H. Greene, and E. C. Norton. 2014. Computation of standard errors. *Health Services Research* 49: 731–750.
- Fishback, P. V., and J. V. Terza. 1989. Are estimates of sex discrimination by employers robust? The use of never-marrieds. *Economic Inquiry* 27: 271–285.
- Mullahy, J. 1997. Instrumental-variable estimation of count data models: Applications to models of cigarette smoking behavior. *Review of Economics and Statistics* 79: 586–593.
- StataCorp. 2017. *Stata 15 Base Reference Manual*. College Station, TX: Stata Press.
- Terza, J. V. 2006. Estimation of policy effects using parametric nonlinear models: A contextual critique of the generalized method of moments. *Health Services and Outcomes Research Methodology* 6: 177–198.
- . 2016a. Simpler standard errors for two-stage optimization estimators. *Stata Journal* 16: 368–385.
- . 2016b. Inference using sample means of parametric nonlinear data transformations. *Health Services Research* 51: 1109–1113.
- . Forthcoming. Two-stage residual inclusion estimation in health services research and health economics. *Health Services Research*.
- Terza, J. V., A. Basu, and P. J. Rathouz. 2008. Two-stage residual inclusion estimation: Addressing endogeneity in health econometric modeling. *Journal of Health Economics* 27: 531–543.

About the author

Joseph V. Terza is a health economist and econometrician in the Department of Economics at Indiana University–Purdue University Indianapolis. His research focuses on the development and application of methods for estimating qualitative and limited dependent variable models with endogeneity. Two of his methods have been implemented as Stata commands. He was keynote speaker at the Stata Users Group meeting in Mexico City in November 2014.

A simple command to calculate travel distance and travel time

Sylvain Weber
University of Neuchâtel
Institute of Economic Research
Neuchâtel, Switzerland
sylvain.weber@unine.ch

Martin Péclat
University of Neuchâtel
Institute of Economic Research
Neuchâtel, Switzerland
martin.peclat@unine.ch

Abstract. Obtaining the routing distance between two addresses should not be a hassle with current technology. Unfortunately, this is more complicated than it first seems. Recently, several commands have been implemented for this purpose (`traveltime`, `traveltime3`, `mvertime`, `osrmtime`), but most of them became obsolete only a few months after their introduction or appear complicated to use. In this article, we introduce the community-contributed command `georoute`, which retrieves travel distance and travel time between two points defined either by their addresses or by their geographical coordinates. Compared with other existing commands, it is simple to use, efficient in terms of computational speed, and versatile regarding the information that can be provided as input.

Keywords: dm0092, `georoute`, `georoutei`, geocoding, travel distance, travel time

1 Introduction

The demand for calculating routing distance between two geographical points is growing. Researchers in energy economics (such as the authors of this article) might be interested in knowing the travel distance between two places for various reasons. Numerous applications in spatial econometrics also rely on such data. The development of large surveys containing addresses (for example, of homes and workplaces) has increased the usefulness of systems allowing to retrieve distances based on such information.

In this article, we follow a series of publications on the topic of geocoding in the *Stata Journal* (Ozimek and Miles 2011; Voorheis 2015; Huber and Rust 2016) and several community-contributed commands available via Statistical Software Components (Anderson 2013; Ansari 2015; Heß 2015; Picard 2010; Zeigermann 2016). However, because the geocoding field is progressing quickly, most of these commands are now obsolete (see Huber and Rust [2016] for a detailed account about what commands are obsolete and why).

In this article, we introduce the community-contributed command `georoute`, which retrieves travel distance and travel time between two points defined by their addresses or their geographical coordinates. Travel distance is the number of miles (or kilometers) one should drive by car to join the first point to the second. Travel time is how long it takes to drive the latter distance under normal traffic conditions. These definitions clarify that the purpose of `georoute` is to provide relevant information for socioeconomic

research. Other existing commands, such as `geodist` (Picard 2010), calculate straight-line distance between two geographical coordinates, which might be relevant in different contexts.

The command `georoute` is close to `mqtime` (Voorheis 2015), which is principally also capable of retrieving travel distances and geographical coordinates from addresses. However, `mqtime` does not always function correctly. Huber and Rust (2016) apparently tested `mqtime` on a bad day and concluded that “`mqtime` no longer works”. The inconsistency of `mqtime` is probably due to MapQuest’s open application programming interface (API), which previously allowed for an unlimited number of requests but has now altered its policy. For requests facing problems with MapQuest’s Open API, `mqtime` uses the HERE API as an alternative to try to retrieve a distance. In `georoute`, we rely directly and only on the HERE API, which is managed by a commercial provider but offers free plans that allow large numbers of requests per month. Thus, `georoute` is many times faster than `mqtime`. `georoute` cannot be guaranteed to remain operational in the long run, because it depends on the stability of the HERE API. The risk of depreciation is minimized by forcing users to create and use their own HERE accounts.

Our command is also closely related to `osrmtime`, implemented by Huber and Rust (2016), but there are two major differences: First, `osrmtime` accepts only geographical coordinates (latitude, longitude) as input, while `georoute` also accepts addresses. Second, we argue that `osrmtime` is quite complicated to use. Indeed, before running `osrmtime`, the user has to follow a series of prerequisites (particularly downloading and preparing the map files), some of which are quite involved and imply a substantial upfront time investment from the user.¹ Contrarily, `georoute` is user friendly and delivers reliable results. Starting from a database of addresses, a user simply needs a one-line command. The only prerequisites are to register for a HERE account and obtain an *App ID* and an *App Code*.

2 The `georoute` and `georoutei` commands

2.1 Prerequisite: Get an HERE account

To use `georoute`, one needs a HERE account (<https://developer.here.com/>). HERE is a commercial provider, but it offers a 90-day free trial of its entire platform, which permits 100,000 requests per month, and a free public basic plan, which is not limited in time and permits 15,000 requests per month.² Such accounts should be largely sufficient for most researchers and most empirical applications. However, `georoute`

1. When we tried to `osrmprepare` the maps for Europe as a whole, the process got stuck while trying to extend the external memory space... and then froze completely while building `node id map` ... Preparing the maps for a single country (in our case, a small one, Switzerland) was less problematic. Nevertheless, `osrmtime` yielded some strange (not to say harmful) outcomes when coordinates outside the country were specified. Instead of excluding such observations, `osrmtime` calculated a clearly incorrect travel distance and duration. The return code, supposed to signal any issue, was nevertheless set as OK for these observations.

2. Moreover, in our experience, it is possible to reactivate a new 90-day free trial once it expires, using the same HERE account.

will make three requests for computing a single routing distance when addresses are provided (one geocoding request per address plus one routing request for calculating the distance between the two points). Thus, the maximal number of travel distances that can be calculated in this case is one-third of the above-mentioned limits. If geographical coordinates are directly specified instead of addresses, only one routing request will be necessary.

After registering in HERE, the user should create a project and get an *App ID* and *App Code* (“JavaScript/REST”). These two elements are necessary for `georoute`.³ Note that a delay of around two hours may occur between the creation of the HERE project and its activation.

2.2 The `georoute` command

Syntax

The syntax of `georoute` is as follows:

```
georoute [if] [in], hereid(string) herecode(string)
  {startaddress(varlist) | startxy(varlist)}
  {endaddress(varlist) | endxy(varlist)} [km distance(newvar) time(newvar)
  diagnostic(newvar) coordinates(str1 str2) replace herepaid timer pause]
```

Options

`hereid(string)` and `herecode(string)` indicate the *App ID* and *App Code* of the user. `hereid()` and `herecode()` are required.

`startaddress(varlist)` and `endaddress(varlist)` specify the addresses of the starting and ending points. Addresses can be inserted as a single variable or as a list of variables. Alternatively, `startxy()` and `endxy()` can be used. Either `startaddress()` or `startxy()` is required. Either `endaddress()` or `endxy()` is required. Note that the presence of special characters (for example, French accents) in addresses might cause errors in the geocoding process. Such characters should be transformed before running `georoute`, for example, using `subinstr()`.

`startxy(varlist)` and `endxy(varlist)` specify the coordinates in decimal degrees of the starting and ending points. They can be used as an alternative to `startaddress()` and `endaddress()`. Two numeric variables containing *x* (latitude) and *y* (longitude) coordinates of the starting and ending points should be provided in `startxy()` and

3. The *App ID* should be a 20-character series such as `BfSfwS1KMCPHj5WbVJ1g`, and the *App Code* a 22-character series such as `bFw1UDZM3Zgc4QM8lyknVg`. We find it useless to provide our own *App ID* and *App Code*, because the maximal number of requests would be exceeded quickly if these were made available to all Stata users.

`endxy()`. Note that x (latitude) must be between -90 and 90 and that y (longitude) must be between -180 and 180 . Examples:

- United States Capitol: 38.8897, -77.0089
- Eiffel Tower: 48.8584, 2.2923
- Cape Horn: -55.9859 , -67.2743
- Pearl Tower: 31.2378, 121.5225

`km` specifies that distances should be returned in kilometers. The default is to return distances in miles.

`distance(newvar)` creates a new variable containing the travel distances between pairs of addresses or geographical points. By default, travel distances will be stored in a variable named `travel_distance`.

`time(newvar)` creates a new variable containing the travel times (by car and under normal traffic conditions) between pairs of addresses or geographical points. By default, travel times will be stored in a variable named `travel_time`.

`diagnostic(newvar)` creates a new variable containing a diagnostic code for the geocoding and georouting outcome of each observation in the database: 0 = OK, 1 = No route found, 2 = Start and/or end not geocoded, 3 = Start and/or end coordinates missing. By default, diagnostic codes will be stored in a variable named `georoute_diagnostic`.

`coordinates(str1 str2)` creates the new variables `str1_x`, `str1_y`, `str1_match`, `str2_x`, `str2_y`, and `str2_match`, which contain the coordinates and the match code of the starting (`str1_x`, `str1_y`, `str1_match`) and ending (`str2_x`, `str2_y`, `str2_match`) addresses. By default, coordinates and match codes are not saved. The match code indicates how well the result matches the request in a 4-point scale: 1 = exact, 2 = ambiguous, 3 = upHierarchy, 4 = ambiguousUpHierarchy.

`replace` specifies that the variables in `distance()`, `time()`, `diagnostic()`, and `coordinates()` be replaced if they already exist in the database. It should be used cautiously because it might cause some data to be lost.

`herepaid` allows the user who owns a paid HERE plan to specify it. This option will simply alter the URL used for the API requests to comply with HERE policy (see <https://developer.here.com/rest-apis/documentation/geocoder/common/request-cit-environment-rest.html>).

`timer` requests that a timer be printed while geocoding. If specified, a dot is printed for every geocoded centile of the dataset, and the number corresponding to every decile is printed. If distances are calculated based on addresses (and not geographical coordinates), two different timers will appear successively: one while geocoding addresses and one while geocoding routes. When geocoding large numbers of observations, this option will inform the user on the expected end time.

`pause` can be used to slow the geocoding process by asking Stata to sleep for 30 seconds every 100th observation. This could be useful for large databases, which might overload the HERE API and result in missing values for batches of observations.

2.3 The `georoutei` command

Syntax

For quick requests for a single pair of addresses or coordinates, we implemented the immediate command `georoutei`, where all arguments must be specified interactively. The syntax of `georoutei` is

```
georoutei, hereid(string) herecode(string)
    {startaddress(string) | startxy(#x, #y)}
    {endaddress(string) | endxy(#x, #y)} [km herepaid]
```

Options

`hereid()`, `herecode()`, `km`, and `herepaid` are exactly as described in section 2.2.

`startaddress(string)` and `endaddress(string)` specify the addresses of the starting and ending points. Addresses must simply be typed within the parentheses. Alternatively, `startxy()` and `endxy()` can be used. Either `startaddress()` or `startxy()` is required. Either `endaddress()` or `endxy()` is required.

`startxy(#x, #y)` and `endxy(#x, #y)` specify the coordinates in decimal degrees of the starting and ending points. They can be used as an alternative to `startaddress()` and `endaddress()`. Coordinates (latitude and longitude) must be specified as two numbers separated by a comma.

Stored results

`georoutei` stores the following in `r()`:

Scalars				
	<code>r(dist)</code>	travel distance	<code>r(time)</code>	travel time
Macros				
	<code>r(start)</code>	coordinates of starting point	<code>r(end)</code>	coordinates of ending point

3 Examples

To illustrate `georoute`, let's build a small dataset:⁴

```
. * Starting points
. input str25 str1 zip1 str15 city1 str11 cntry1
           str11      zip1      city1      cntry1
1. "Rue de la Tambourine 17" 1227 "Carouge" "Switzerland"
2. "" 1003 "Lausanne" "Switzerland"
3. "" . "Paris" "France"
4. "" 1003 "Lausanne" "Switzerland"
5. end

. * Ending points
. input str25 str2 zip2 str15 city2 str11 cntry2
           str11      zip2      city2      cntry2
1. "Rue Abram-Louis Breguet 2" 2000 "Neuchatel" "Switzerland"
2. "" 74500 "Evian" "France"
3. "" . "New York" "USA"
4. "" 1203 "Geneva" "Switzerland"

. *Compute distances using georoute
. georoute, hereid(BfSfwSlKMCPHj5WbVJ1g) herecode(bFw1UDZM3Zgc4QM8lyknVg)
> startad(str1 zip1 city1 cntry1)
> endad(str2 zip2 city2 cntry2) km distance(dist) time(time) coordinates(p1 p2)
. format dist time %7.2f
. list city1 cntry1 city2 cntry2 dist time
```

	city1	cntry1	city2	cntry2	dist	time
1.	Carouge	Switzerland	Neuchatel	Switzerland	135.68	87.02
2.	Lausanne	Switzerland	Evian	France	73.22	77.73
3.	Paris	France	New York	USA	.	.
4.	Lausanne	Switzerland	Geneva	Switzerland	64.53	47.12

For the record, the first observation contains the office addresses of the two authors of this article. Both of them live close to the city where the other works; the outcome essentially reveals their daily back-and-forth travel distance.

The second observation was chosen to demonstrate an essential feature of `georoute`. Both the cities of Lausanne and Evian are located on the shores of Geneva's Lake: Lausanne in the north and Evian in the south. By car, one would have to drive around the lake, a 73.22-kilometer distance. However, connecting these 2 cities by a straight line would result in 17.75 kilometers, as shown by `geodist`'s output:

```
. geodist p1_x p1_y p2_x p2_y, gen(distlin)
. format distlin %7.2f
. format p?_? %5.2f
```

4. Be warned that simply introducing the following lines in Stata will result in an error message, because the *App ID* and *App Code* displayed here are invalid. To replicate the results, include your own *App ID* and *App Code* (see section 2.1).

```
. list city1 p1_x p1_y city2 p2_x p2_y dist distlin
```

	city1	p1_x	p1_y	city2	p2_x	p2_y	dist	distlin
1.	Carouge	46.18	6.14	Neuchatel	46.99	6.94	135.68	109.76
2.	Lausanne	46.52	6.63	Evian	46.36	6.65	73.22	17.75
3.	Paris	48.86	2.34	New York	40.71	-74.01	.	5852.14
4.	Lausanne	46.52	6.63	Geneva	46.21	6.12	64.53	52.17

On the other hand, one may notice with the third observation that no distance is computed by `georoute` between Paris and New York (for obvious reasons), but `geodist` indicates the geodetic distance as being almost 6,000 km. The purposes of these two commands are different, and which distance (travel distance from `georoute` or geodetic distance from `geodist`) to use depends on the goal of the user. In less obvious cases, where doubts might remain about the reason why no distance was obtained with `georoute`, the variable `georoute_diagnostic` could offer some guidance:

```
. list city1 city2 georoute_diagnostic
```

	city1	city2	georoute_dia-c
1.	Carouge	Neuchatel	OK
2.	Lausanne	Evian	OK
3.	Paris	New York	No route found
4.	Lausanne	Geneva	OK

Note also that `geodist` could be used thanks to the latitudes and longitudes (variables `p1_x`, `p1_y`, `p2_x`, and `p2_y`) previously produced by `georoute` with the option `coordinates`. In that sense, these two commands are complementary. Furthermore, note that `georoute` is quite versatile regarding how addresses can be specified. If several variables should be combined to produce the entire address, these different variables, be they string or numeric, can be simply introduced in the `startaddress()` and `endaddress()` options as a variable list.

By comparing the second and fourth observations, we see another interesting feature of `georoute`. While routing distances are comparable for Lausanne–Evian and Lausanne–Geneva, one may notice that travel time is much lower for the latter. This is because most travel between Lausanne and Geneva can be done on a highway, while a large share of the travel between Lausanne and Evian takes place on regional roads with much lower speed limits. Distance and time are thus two different dimensions of travel, and both might be useful in empirical applications.

Finally, let us assume we want to check one of the results obtained above. In this case, the immediate command `georoutei` would be convenient. For instance, one could obtain the results for the first observation as follows:

```
. georoutei, hereid(BfSfwSlKMCPHj5WbVJ1g) herecode(bFw1UDZM3Zgc4QM8lyknVg)
> startad(Rue de la Tambourine 17, 1227 Carouge, Switzerland)
> endad(Rue Abram-Louis Breguet 2, 2000 Neuchatel, Switzerland) km
-----
From: Rue de la Tambourine 17, 1227 Carouge, Switzerland (46.17556,6.13906)
To:   Rue Abram-Louis Breguet 2, 2000 Neuchatel, Switzerland (46.99382,6.94049)
-----
Travel distance:    135.68 kilometers
Travel time:        87.02 minutes
```

Given that we also know latitudes and longitudes corresponding to the addresses from the previous call of `georoute`, we could provide this information to `georoutei`:⁵

```
. georoutei, hereid(BfSfwSlKMCPHj5WbVJ1g) herecode(bFw1UDZM3Zgc4QM8lyknVg)
> startxy(46.1761413,6.1393099) endxy(46.99382,6.94049) km
-----
From: (46.1761413,6.1393099)
To:   (46.99382,6.94049)
-----
Travel distance:    135.68 kilometers
Travel time:        87.02 minutes
```

We emphasize that both travel distance and time are the same as before. This is an important feature of the HERE API: it provides travel time under normal traffic conditions. Said otherwise, the results will not be influenced by current traffic conditions, which is essential in terms of reproducibility. Whenever `georoute` and `georoutei` are run, results will be identical (unless, of course, roads have been built or closed in the meantime).

4 Conclusion

The techniques for geocoding evolve at a rapid pace. Consequently, new commands appear and depreciate rapidly. In this article, we introduce the community-contributed command `georoute`, which computes travel distance and time between two points defined by their addresses or their geographical coordinates. Like its predecessors, the longevity of `georoute` depends on whether the commercial provider HERE will maintain its API unchanged or alter its terms of use. Nevertheless, we have tried to minimize the risk of obsolescence by forcing users to use their own HERE accounts, which are free and benefit from a substantial number of requests.

Compared with existing commands that have a similar purpose and are still in operation, `georoute` possesses several advantages. Compared with `mqtime`, it is computationally efficient and versatile regarding how addresses can be specified, and it encompasses many additional options. Moreover, `mqtime` does not always work, whereas many checks have not revealed any inconsistency in `georoute`. Compared with `osrmtime`, `georoute` is objectively simpler to use and can be used on addresses, while `osrmtime` can obtain distances only between coordinates and might thus require a first step to geocode ad-

5. Note that to get strictly identical results when using coordinates instead of addresses, one must include all available digits in the latitudes and longitudes.

addresses if this is the only information initially available. Hopefully, `georoute` should aid researchers for a long time.

5 Acknowledgments

This research was supported by the Swiss National Science Foundation Grant 100018-144310 and is part of the activities of SCCER CREST, which is financially supported by the Swiss Commission for Technology and Innovation (CTI).

6 References

- Anderson, M. L. 2013. `geocodeopen`: Stata module to geocode addresses using MapQuest Open Geocoding Services and Open Street Maps. Statistical Software Components S457733, Department of Economics, Boston College. <https://ideas.repec.org/c/boc/bocode/s457733.html>.
- Ansari, M. R. 2015. `gcode`: Stata module to download Google geocode data. Statistical Software Components S457969, Department of Economics, Boston College. <https://ideas.repec.org/c/boc/bocode/s457969.html>.
- Heß, S. 2015. `geocodehere`: Stata module to provide geocoding relying on Nokia's Here Maps API. Statistical Software Components S457969, Department of Economics, Boston College. <https://ideas.repec.org/c/boc/bocode/s458048.html>.
- Huber, S., and C. Rust. 2016. Calculate travel time and distance with OpenStreetMap data using the Open Source Routing Machine (OSRM). *Stata Journal* 16: 416–423.
- Ozimek, A., and D. Miles. 2011. Stata utilities for geocoding and generating travel time and travel distance information. *Stata Journal* 11: 106–119.
- Picard, R. 2010. `geodist`: Stata module to compute geodetic distances. Statistical Software Components S457147, Department of Economics, Boston College. <https://ideas.repec.org/c/boc/bocode/s457147.html>.
- Voorheis, J. 2015. `mqtime`: A Stata tool for calculating travel time and distance using MapQuest web services. *Stata Journal* 15: 845–853.
- Zeigermann, L. 2016. `opencagegeo`: Stata module for forward and reverse geocoding using the OpenCage Geocoder API. Statistical Software Components S458155, Department of Economics, Boston College. <https://ideas.repec.org/c/boc/bocode/s458155.html>.

About the authors

Sylvain Weber is a postdoctoral researcher at the Institute of Economic Research at the University of Neuchâtel (Switzerland). His main field of research is energy economics. In particular, he studies private mobility, hence his interest for calculating travel distances precisely and easily.

Martin Péclat is a PhD student at the University of Neuchâtel (Switzerland) and at the Geneva School of Business Administration HES-SO, University of Applied Sciences of Western Switzerland (Switzerland). His thesis is about the determinants of spatial diffusion and adoption of solar photovoltaic technology in Switzerland.

Testing for Granger causality in panel data

Luciano Lopez
University of Neuchâtel
Neuchâtel, Switzerland
luciano.lopez@unine.ch

Sylvain Weber
University of Neuchâtel
Neuchâtel, Switzerland
sylvain.weber@unine.ch

Abstract. With the development of large and long panel databases, the theory surrounding panel causality evolves quickly, and empirical researchers might find it difficult to run the most recent techniques developed in the literature. In this article, we present the community-contributed command `xtgcause`, which implements a procedure proposed by [Dumitrescu and Hurlin \(2012, *Economic Modelling* 29: 1450–1460\)](#) for detecting Granger causality in panel datasets. Thus, it constitutes an effort to help practitioners understand and apply the test. `xtgcause` offers the possibility of selecting the number of lags to include in the model by minimizing the Akaike information criterion, Bayesian information criterion, or Hannan–Quinn information criterion, and it offers the possibility to implement a bootstrap procedure to compute p -values and critical values.

Keywords: st0507, `xtgcause`, Granger causality, panel datasets, bootstrap

1 Introduction

Panel datasets composed of many individuals and many time periods are becoming widely available. A particularly salient case is the growing availability of cross-country data over time. As a consequence, the focus of panel-data econometrics is shifting from micropanel, with large N and small T , to macropanel, with large N and large T . In this setting, classical issues of time-series econometrics, such as (non)stationarity and (non)causality, also arise. In this article, we present the community-contributed command `xtgcause`, which implements a procedure developed by [Dumitrescu and Hurlin \(2012\)](#) (the DH test) to test for Granger causality in panel datasets.

Considering the fast evolution of the literature, practitioners may find it difficult to implement the latest econometric tests. Therefore, in this article, we summarize the test built by [Dumitrescu and Hurlin \(2012\)](#) and present `xtgcause` using examples based on simulated and real data. The objective of our contribution is to support the empirical literature using panel causality techniques. One recurrent concern is the selection of the number of lags to be included in the estimations; we have implemented an extension of the test based on Akaike information criteria (AIC), Bayesian information criteria (BIC), and Hannan–Quinn information criteria (HQIC) to facilitate this task. Finally, to deal with the empirical issue of cross-sectional dependence, we have implemented an option to compute p -values and critical values based on a bootstrap procedure.

2 The Dumitrescu–Hurlin test

In a seminal article, [Granger \(1969\)](#) developed a methodology for analyzing the causal relationships between time series. Suppose x_t and y_t are two stationary series. The model

$$y_t = \alpha + \sum_{k=1}^K \gamma_k y_{t-k} + \sum_{k=1}^K \beta_k x_{t-k} + \varepsilon_t \quad \text{with } t = 1, \dots, T \quad (1)$$

can then be used to test whether x causes y . Essentially, if past values of x are significant predictors of the current value of y even when past values of y have been included in the model, then x exerts a causal influence on y . Using (1), one might easily investigate this causality based on an F test with the following null hypothesis:

$$H_0: \beta_1 = \dots = \beta_K = 0$$

If H_0 is rejected, one can conclude that causality from x to y exists. The x and y variables can be interchanged to test for causality in the other direction, and it is possible to observe bidirectional causality (also called feedback).

[Dumitrescu and Hurlin \(2012\)](#) provide an extension designed to detect causality in panel data. The underlying regression is

$$y_{i,t} = \alpha_i + \sum_{k=1}^K \gamma_{ik} y_{i,t-k} + \sum_{k=1}^K \beta_{ik} x_{i,t-k} + \varepsilon_{i,t} \quad \text{with } i = 1, \dots, N \text{ and } t = 1, \dots, T \quad (2)$$

where $x_{i,t}$ and $y_{i,t}$ are the observations of two stationary variables for individual i in period t . Coefficients are allowed to differ across individuals (note the i subscripts attached to coefficients) but are assumed to be time invariant. The lag order K is assumed to be identical for all individuals, and the panel must be balanced.

As in [Granger \(1969\)](#), the procedure to determine the existence of causality is to test for significant effects of past values of x on the present value of y . The null hypothesis is therefore defined as

$$H_0: \beta_{i1} = \dots = \beta_{iK} = 0 \quad \forall i = 1, \dots, N \quad (3)$$

which corresponds to the absence of causality for all individuals in the panel.

The DH test assumes there can be causality for some individuals but not necessarily for all. Thus, the alternative hypothesis is

$$\begin{aligned} H_1: & \beta_{i1} = \dots = \beta_{iK} = 0 & \forall i = 1, \dots, N_1 \\ & \beta_{i1} \neq 0 \text{ or } \dots \text{ or } \beta_{iK} \neq 0 & \forall i = N_1 + 1, \dots, N \end{aligned}$$

where $N_1 \in [0, N - 1]$ is unknown. If $N_1 = 0$, there is causality for all individuals in the panel. N_1 must be strictly smaller than N ; otherwise, there is no causality for all individuals, and H_1 reduces to H_0 .

Against this backdrop, [Dumitrescu and Hurlin \(2012\)](#) propose the following procedure: run the N individual regressions implicitly enclosed in (2), perform F tests of the

K linear hypotheses $\beta_{i1} = \dots = \beta_{iK} = 0$ to retrieve the individual Wald statistic W_i , and finally compute the average Wald statistic \bar{W} :¹

$$\bar{W} = \frac{1}{N} \sum_{i=1}^N W_i$$

We emphasize that the test is designed to detect causality at the panel level, and rejecting H_0 does not exclude noncausality for some individuals. Using Monte Carlo simulations, Dumitrescu and Hurlin (2012) show that \bar{W} is asymptotically well behaved and can genuinely be used to investigate panel causality.

Under the assumption that the Wald statistics W_i are independently and identically distributed across individuals, it can be shown that the standardized statistic \bar{Z} when $T \rightarrow \infty$ first and then $N \rightarrow \infty$ (sometimes interpreted as “ T should be large relative to N ”) follows a standard normal distribution:

$$\bar{Z} = \sqrt{\frac{N}{2K}} \times (\bar{W} - K) \xrightarrow[T, N \rightarrow \infty]{d} \mathcal{N}(0, 1) \quad (4)$$

Also, for a fixed T dimension with $T > 5 + 3K$, the approximated standardized statistic \tilde{Z} follows a standard normal distribution:

$$\tilde{Z} = \sqrt{\frac{N}{2K}} \times \frac{T - 3K - 5}{T - 2K - 3} \times \left(\frac{T - 3K - 3}{T - 3K - 1} \times \bar{W} - K \right) \xrightarrow[N \rightarrow \infty]{d} \mathcal{N}(0, 1) \quad (5)$$

The testing procedure of the null hypothesis in (3) is finally based on \bar{Z} and \tilde{Z} . If these are larger than the standard critical values, then one should reject H_0 and conclude that Granger causality exists. For large N and T panel datasets, \bar{Z} can be reasonably considered. For large N but relatively small T datasets, \tilde{Z} should be favored. Using Monte Carlo simulations, Dumitrescu and Hurlin (2012) have shown that the test exhibits good finite sample properties, even when both T and N are small.

The lag order (K) selection is an empirical issue for which Dumitrescu and Hurlin (2012) provide no guidance. One way to tackle this issue is to select the number of lags based on an information criterion (AIC/BIC/HQIC). In this process, all estimations have to be conducted on a common sample to be nested and therefore comparable.² Practically, this implies that the first K_{\max} ³ time periods must be omitted during the entire lag-selection process.

Another empirical issue to consider in panel data is cross-sectional dependence. To this end, a block bootstrap procedure is proposed in section 6.2 of Dumitrescu and

1. See Dumitrescu and Hurlin (2012, 1453) for the mathematical definition of W_i . Note, however, that T in their formulas must be understood as the number of observations remaining in the estimations, that is, the number of periods minus the number of lags included. To be consistent with our notation, we therefore replaced Dumitrescu and Hurlin’s (2012) T by $T - K$ in the following formulas of the present article.

2. We thank Gareth Thomas (IHS Markit EViews) for bringing this point to our attention.

3. K_{\max} stands for the maximum possible number of lags to be considered in the entire procedure.

Hurlin (2012) to compute bootstrapped critical values for \bar{Z} and \tilde{Z} instead of asymptotic critical values. The procedure has the following steps:⁴

1. Fit (2) and obtain \bar{Z} and \tilde{Z} as defined in (4) and (5).
2. Fit the model under $H_0: y_{i,t} = \alpha_i^0 + \sum_{k=1}^K \gamma_{ik}^0 y_{i,t-k} + \varepsilon_{i,t}$, and collect the residuals in matrix $\hat{\varepsilon}_{(T-K) \times N}$.
3. Build a matrix $\varepsilon_{(T-K) \times N}^*$ by resampling (overlapping blocks of) rows (that is, time periods) of matrix $\hat{\varepsilon}$. Block bootstrap is useful when there is autocorrelation.
4. Generate a random draw $(\mathbf{y}_1^*, \dots, \mathbf{y}_K^*)'$, with $\mathbf{y}_t^* = (y_{1,t}^*, y_{2,t}^*, \dots, y_{N,t}^*)$, by randomly selecting a block of K consecutive time periods with replacement (see [Stine \[1987\]](#) and [Berkowitz and Kilian \[2000\]](#)).
5. Construct the resampled series $y_{i,t}^* = \hat{\alpha}_i^0 + \sum_{k=1}^K \hat{\beta}_{ik}^0 y_{i,t-k}^* + \varepsilon_{i,t}^*$ conditional on the random draw for the first K periods.
6. Fit the model $y_{i,t}^* = \alpha_i^b + \sum_{k=1}^K \gamma_{ik}^b y_{i,t-k}^* + \sum_{k=1}^K \beta_{ik}^b x_{i,t-k} + \varepsilon_{i,t}$ and compute \bar{Z}^b and \tilde{Z}^b .
7. Run B replications of steps 3 to 6.
8. Compute p -values and critical values for \bar{Z} and \tilde{Z} based on the distributions of \bar{Z}^b and \tilde{Z}^b , $b = 1, \dots, B$.

3 The xtgcause command

3.1 Syntax

The syntax of `xtgcause` is

```
xtgcause devar indepvar [if] [in] [, lags(#|aic [#]|bic [#]|hqic [#])
regress bootstrap breps(#) blevel(#) blength(#) seed(#) nodots]
```

3.2 Options

`lags(#|aic [#]|bic [#]|hqic [#])` specifies the lag structure to use for the regressions performed in computing the test statistic. The default is `lags(1)`.

4. The procedure we present here differs slightly from that proposed by [Dumitrescu and Hurlin \(2012\)](#) in the numbering of the steps but, more importantly, in the definition of the initial conditions (our step 4), which is not addressed by [Dumitrescu and Hurlin \(2012\)](#), and in the construction of the resampled series (our step 5). We are indebted to David Ardia (University of Neuchâtel) for his valuable advice on the bootstrap procedure.

Specifying `lags(#)` requests that `#` lags of the series be used in the regressions. The maximum authorized number of lags is such that $T > 5 + 3 \times \#$.

Specifying `lags(aic|bic|hqic [#])` requests that the number of lags of the series be chosen such that the average AIC/BIC/HQIC for the set of regressions is minimized. Regressions with 1 to `#` lags will be conducted, restricting the number of observations to $T - \#$ for all estimations to make the models nested and therefore comparable. Displayed statistics come from the set of regressions for which the average AIC/BIC/HQIC is minimized (reestimated using the total number of observations available). If `#` is not specified in `lags(aic|bic|hqic [#])`, then it is set to the maximum number of lags authorized.

`regress` can be used to display the results of the N individual regressions on which the test is based. This option is useful to have a look at the coefficients of individual regressions. When the number of individuals in the panel is large, this option will result in a very long output.

`bootstrap` requests p -values and critical values be computed using a bootstrap procedure as proposed in [Dumitrescu and Hurlin \(2012, sec. 6.2\)](#). The bootstrap procedure is useful when there is cross-sectional dependence.

`breps(#)` indicates the number of bootstrap replications to perform. The default is `breps(1000)`.

`blevel(#)` indicates the significance level (in %) for computing the bootstrapped critical values. The default is `blevel(95)`.

`blength(#)` indicates the size of the block length to be used in the bootstrap. By default, each time period is sampled independently with replacement `blength(1)`.

`blength()` allows the user to implement the bootstrap by dividing the sample into overlapping blocks of `#` time periods and sampling the blocks independently with the replacement. Using blocks of more than one time period is useful if autocorrelation is suspected.

`seed(#)` can be used to set the random-number seed. By default, the seed is not set.

`nodots` suppresses replication dots. By default, a dot is printed for each replication to provide an indication of the evolution of the bootstrap.

`breps()`, `blevel()`, `blength()`, `seed()`, and `nodots` are `bootstrap` suboptions. They can be used only if `bootstrap` is also specified.

3.3 Stored results

`xtgcause` stores the following in `r()`:

Scalars

<code>r(wbar)</code>	average Wald statistic	<code>r(zbart)</code>	Z-bar tilde statistic
<code>r(lags)</code>	number of lags used for the test	<code>r(zbart_pv)</code>	<i>p</i> -value of the Z-bar tilde statistic
<code>r(zbar)</code>	Z-bar statistic		
<code>r(zbar_pv)</code>	<i>p</i> -value of the Z-bar statistic		

Matrices

<code>r(Wi)</code>	individual Wald statistics	<code>r(PVi)</code>	<i>p</i> -values of the individual Wald statistics
--------------------	----------------------------	---------------------	--

`xtgcause` with the `bootstrap` option also stores the following additional results in `r()`:

Scalars

<code>r(zbarb_cv)</code>	critical value for the Z-bar statistic	<code>r(blevel)</code>	significance level for bootstrap critical values
<code>r(zbartb_cv)</code>	critical value for the Z-bar tilde statistic	<code>r(blenght)</code>	size of the block length
<code>r(breps)</code>	number of bootstrap replications		

Matrices

<code>r(ZBARb)</code>	Z-bar statistics from the bootstrap procedure	<code>r(ZBARTb)</code>	Z-bar tilde statistics from the bootstrap procedure
-----------------------	---	------------------------	---

4 Examples

Before presenting some examples, we recall that the test implemented in `xtgcause` assumes that the variables are stationary. We will not go through this first step here, but it is the user's responsibility to verify that the data satisfy this condition. To this end, the user might consider `xtunitroot`, which provides various panel stationarity tests with alternative null hypotheses (in particular, Breitung [2000]; Hadri [2000]; Harris and Tzavalis [1999]; Im, Pesaran, and Shin [2003]; Levin, Lin, and Chu [2002]). The user may also want to perform second-generation panel unit-root tests such as the one proposed by Pesaran (2007) to control for cross-sectional dependence.

4.1 Example based on simulated data

To illustrate `xtgcause`, we first use simulated data provided by Dumitrescu and Hurlin (2012) at <http://www.execandshare.org/execandshare/htdocs/data/MetaSite/upload/companionSite51/data/> in the file `data-demo.csv`.⁵ Then, we import the original Excel dataset directly from the website. In the original CSV file, the dataset is organized as a matrix, with all observations for each individual in a single cell. Within this cell, the 10 values of variable x are separated by tabs, a comma separates the last value of x and the first value of y , and the 10 values of variable y are then separated by tabs. Hence, the following lines of code allow shaping the data to be understood as a panel by Stata.

⁵ Data and MATLAB code are also available at <http://www.runmycode.org/companion/view/42> in a ZIP file.

```

. import delimited using "http://www.execandshare.org/execandshare/htdocs/data/
> MetaSite/upload/companionSite51/data/data-demo.csv", delimiter(",")
> colrange(1:2) varnames(1)
(2 vars, 20 obs)

. quietly: split x, parse(`=char(9)`) destring
. quietly: split y, parse(`=char(9)`) destring
. drop x y
. generate t = _n
. reshape long x y, i(t) j(id)
(note: j = 1 2 3 4 5 6 7 8 9 10)
Data                                wide  ->  long
-----
Number of obs.                      20   ->   200
Number of variables                  21   ->    4
j variable (10 values)              ->   id
xij variables:
                                x1 x2 ... x10 ->  x
                                y1 y2 ... y10 ->  y
-----

. xtset id t
      panel variable:  id (strongly balanced)
      time variable:  t, 1 to 20
                delta: 1 unit

. list id t x y in 1/5

```

	id	t	x	y
1.	1	1	.55149203	.81872837
2.	1	2	.64373514	-.42077179
3.	1	3	-.58843258	-.40312278
4.	1	4	-.55873336	.14674849
5.	1	5	-.32486386	.42924677

```

. list id t x y in 21/25

```

	id	t	x	y
21.	2	1	-1.4703536	1.2586422
22.	2	2	1.3356281	-.71173904
23.	2	3	-.21564623	-.73264199
24.	2	4	.08435614	-.67841901
25.	2	5	1.5766581	-.2562083

Some sections of the above code are quite involved and require explanations. We started by importing the data as if values were separated by commas, which is only partly true. This created two string variables, named `x` and `y`, each containing 10 values (separated by tabs) in each observation. We then invoked `split`, using `char(9)` (which indeed corresponds to a tab) as the parse string. We used the prefix `quietly` to avoid a long output indicating that 2 sets of 10 variables (x_1, \dots, x_{10} , and y_1, \dots, y_{10}) were created. These variables were immediately converted from string to numeric thanks to `split`'s `destring` option. To have a well-shaped panel that Stata can correctly interpret, we combined these 2 sets of 10 variables into only 2 variables using `reshape`.

A few observations (the first five for individuals 1 and 2) are displayed to show how the data are finally organized.

After we format and `xtset` the data, we can now run `xtgcause`. The simplest possible test to investigate whether x causes y would be

```
. xtgcause y x

Dumitrescu & Hurlin (2012) Granger non-causality test results:
-----
Lag order: 1
W-bar =          1.2909
Z-bar =          0.6504   (p-value = 0.5155)
Z-bar tilde =    0.2590   (p-value = 0.7956)
-----

H0: x does not Granger-cause y.
H1: x does Granger-cause y for at least one panelvar (id).
```

Because we did not specify any lag order, `xtgcause` introduced a single lag by default. In this case, the outcome of the test does not reject the null hypothesis. The output reports the values obtained for \bar{W} (`W-bar`), \bar{Z} (`Z-bar`), and \tilde{Z} (`Z-bar tilde`). For the latter two statistics, p -values are provided based on the standard normal distribution.

One could additionally display the individual Wald statistics and their corresponding values by displaying the stored matrices `r(Wi)` and `r(PVi)` (which we first combine into a single matrix for the sake of space):

```
. matrix Wi_PVi = r(Wi), r(PVi)
. matrix list Wi_PVi
Wi_PVi[10,2]
      Wi          PVi
id1  .56655945  .46256089
id2  .11648998  .73731411
id3  .09081952  .76701924
id4  8.1263612  .01156476
id5  .18687517  .67129995
id6  .80060395  .38417583
id7  .53075859  .47681675
id8  .00158371  .96874825
id9  .43635413  .5182858
id10 2.0521113  .17124367
```

Using the `lags()` option, we run a similar test introducing two lags of the variables `x` and `y`:

```
. xtgcause y x, lags(2)
-----
Dumitrescu & Hurlin (2012) Granger non-causality test results:
-----
Lag order: 2
W-bar =          1.7302
Z-bar =         -0.4266   (p-value = 0.6696)
Z-bar tilde =    -0.7052   (p-value = 0.4807)
-----
H0: x does not Granger-cause y.
H1: x does Granger-cause y for at least one panelvar (id).
```

The conclusion of the test is similar to before.

Alternatively, the test could also be conducted using a bootstrap procedure to compute p -values and critical values:

```
. xtgcause y x, bootstrap lags(1) breps(100) seed(20171020)
-----
Bootstrap replications (100)
-----
..... 50
..... 100
-----
Dumitrescu & Hurlin (2012) Granger non-causality test results:
-----
Lag order: 1
W-bar =          1.2909
Z-bar =          0.6504   (p-value* = 0.4700, 95% critical value = 1.7316)
Z-bar tilde =     0.2590   (p-value* = 0.7100, 95% critical value = 1.3967)
-----
H0: x does not Granger-cause y.
H1: x does Granger-cause y for at least one panelvar (id).
*p-values computed using 100 bootstrap replications.
```

In this case, the bootstrapped p -values are relatively close to the asymptotic ones displayed in the first test above.

4.2 Example based on real data

To provide an example based on real data, we searched for articles reporting Dumitrescu and Hurlin's (2012) tests and published in journals that make authors' datasets available. We found several such articles (for example, Paramati, Ummalla, and Apergis [2016]; Paramati, Apergis, and Ummalla [2017]; Salahuddin, Alam, and Ozturk [2016]). In particular, Paramati, Ummalla, and Apergis (2016) investigate the effect of foreign direct investment and stock market growth on clean energy use. In their table 8, they report a series of pairwise panel causality tests between variables such as economic output, CO₂ emissions, or foreign direct investment. As indicated in their online supplementary data (file `Results.xlsx`), they conduct the tests using EViews 8. We replicate some of their results:

```

. import excel using data-wdi.xlsx, clear first case(lower) cellrange(A1:I421)
> sheet(FirstDif-Data)

. xtset id year
      panel variable: id (strongly balanced)
      time variable: year, 1992 to 2012
      delta: 1 unit

. xtgcause co2 output, lags(2)

Dumitrescu & Hurlin (2012) Granger non-causality test results:
-----
Lag order: 2
W-bar =          2.4223
Z-bar =          0.9442 (p-value = 0.3451)
Z-bar tilde =    0.1441 (p-value = 0.8855)
-----
H0: output does not Granger-cause co2.
H1: output does Granger-cause co2 for at least one panelvar (id).

. xtgcause fdi output, lags(2)

Dumitrescu & Hurlin (2012) Granger non-causality test results:
-----
Lag order: 2
W-bar =          4.6432
Z-bar =          5.9103 (p-value = 0.0000)
Z-bar tilde =    3.7416 (p-value = 0.0002)
-----
H0: output does not Granger-cause fdi.
H1: output does Granger-cause fdi for at least one panelvar (id).

```

The above code imports the dataset constructed by Paramati, Ummalla, and Apergis (2016) (file `Data-WDI.xlsx`, sheet `FirstDif-Data`) as a first step. We then use `xtgcause` to test for the causality from `output` to `co2` and from `output` to `fdi`, which correspond to some tests reported in their table 8. We use two lags in both cases to match the numbers indicated by Paramati, Ummalla, and Apergis (2016) in their accompanying appendix file. Compared with their output, it turns out that the denomination “Zbar-Stat” used in EViews corresponds to the `Z-bar tilde` statistic (while the `Z-bar` statistic is not provided in EViews).

Optionally, `xtgcause` allows the user to request the lag order to be chosen so that the AIC, BIC, or HQIC be minimized. Given that Dumitrescu and Hurlin (2012) offer no guidance regarding the choice of the lag order, this feature might be appealing to practitioners. We can, for example, test the causality from `output` to `fdi` specifying the option `lags(bic)`:

```

. xtgcause fdi output, lags(bic)

Dumitrescu & Hurlin (2012) Granger non-causality test results:
-----
Optimal number of lags (BIC): 1 (lags tested: 1 to 5).
W-bar =          1.3027
Z-bar =          0.9572 (p-value = 0.3385)
Z-bar tilde =    0.4260 (p-value = 0.6701)
-----
H0: output does not Granger-cause fdi.
H1: output does Granger-cause fdi for at least one panelvar (id).

```

In practice, `xtgcause` runs all sets of regressions with a lag order from 1 to the highest possible number (that is, such that $T > 5 + 3K$ or optionally specified by the user below this limit), maintaining a common sample. Said otherwise, if at most five lags are to be considered, the first five observations of the panel will never be considered in the estimations, even if it would be possible to do so with fewer than five lags. This ensures nested models, which can then be appropriately compared using AIC, BIC, or HQIC. After this series of estimations, `xtgcause` selects the optimal outcome (that is, such that the average AIC/BIC/HQIC of the N individual estimations is the lowest) and reruns all estimations with the optimal number of lags and using all observations available. Statistics based on the latter are reported as output.

In the above example, the optimal lag order using BIC appears to be 1, which is different from the lag order selected by [Paramati, Ummalla, and Apergis \(2016\)](#) for this test.⁶ This difference is not without consequences, because the conclusion of the test is then reversed. More precisely, the null hypothesis is not rejected with the optimally selected single lag, but [Paramati, Ummalla, and Apergis \(2016\)](#) use two lags and therefore reject the null hypothesis. Considering that empirical research in economics is used to formulate policy recommendations, such inaccurate conclusions may potentially be harmful. We therefore consider `xtgcause`'s option allowing the user to select the number of lags based on AIC/BIC/HQIC as an important improvement. It will allow researchers to rely on these widely accepted criteria and make the selection in a transparent way.

Finally, `xtgcause` makes it possible to compute the p -values and critical values associated with the Z -bar and Z -bar tilde via a bootstrap procedure. Computing bootstrapped critical values (rather than asymptotic ones) may be useful when there is cross-sectional dependence. Based on the [Paramati, Ummalla, and Apergis \(2016\)](#) data, we test the causality from `output` to `fdi` by adding the `bootstrap` option (we also use `seed` for replicability reasons and `nodots` for the sake of space):

```
. xtgcause fdi output, lags(bic) bootstrap seed(20171020) nodots
-----
Bootstrap replications (1000)
-----

Dumitrescu & Hurlin (2012) Granger non-causality test results:
-----
Optimal number of lags (BIC): 1 (lags tested: 1 to 5).
W-bar =          1.3027
Z-bar =          0.9572 (p-value* = 0.4530, 95% critical value = 3.0746)
Z-bar tilde =    0.4260 (p-value* = 0.7080, 95% critical value = 2.1234)
-----
H0: output does not Granger-cause fdi.
H1: output does Granger-cause fdi for at least one panelvar (id).
*p-values computed using 1000 bootstrap replications.
```

Here `xtgcause` first computes the Z -bar and Z -bar tilde statistics using the optimal number of lags as in previous series of estimations; then, it computes the boot-

6. The number of lags would be three using HQIC and four using AIC. Therefore, while [Paramati, Ummalla, and Apergis \(2016\)](#) state in their table 8 that “the appropriate lag length is chosen based on SIC”, we do not find the same number with any of the information criterion considered.

strapped p -values and critical values. By default, 1,000 bootstrap replications are performed. We observe that the bootstrapped p -value for the `Z-bar` increases substantially compared with the asymptotic p -value obtained before (from 0.34 to 0.45), while that for the `Z-bar tilde` remains closer. This should be interpreted as a signal that the estimations suffer from small sample biases, so asymptotic p -values are underestimated. Bootstrapped p -values indicate that the null hypothesis is far from being rejected, which strengthens the concerns about Paramati, Ummalla, and Apergis's (2016) conclusions based on the asymptotic p -values and obtained with two lags.

5 Conclusion

In this article, we presented the community-contributed command `xtgcause`, which automates a procedure introduced by Dumitrescu and Hurlin (2012) to detect Granger causality in panel datasets. In this branch of econometrics, the empirical literature appears to be lagging, with the latest theoretical developments not always being available in statistical packages. One important contribution of our command is to allow the user to select the number of lags based on the AIC, the BIC, or the HQIC. This choice may impact the conclusion of the test, but some researchers may have overlooked it. Thus, several empirical articles might have reached erroneous conclusions. `xtgcause` also allows the user to calculate bootstrapped critical values, which is a useful option when there is cross-sectional dependence. With this command and this article, we hope to bring some useful clarifications and help practitioners conduct sound research.

6 References

- Berkowitz, J., and L. Kilian. 2000. Recent developments in bootstrapping time series. *Econometric Reviews* 19: 1–48.
- Breitung, J. 2000. The local power of some unit root tests for panel data. In *Advances in Econometrics: Vol. 15—Nonstationary Panels, Panel Cointegration, and Dynamic Panels*, ed. B. H. Baltagi, 161–178. New York: Elsevier.
- Dumitrescu, E.-I., and C. Hurlin. 2012. Testing for Granger non-causality in heterogeneous panels. *Economic Modelling* 29: 1450–1460.
- Granger, C. W. J. 1969. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica* 37: 424–438.
- Hadri, K. 2000. Testing for stationarity in heterogeneous panel data. *Econometrics Journal* 3: 148–161.
- Harris, R. D. F., and E. Tzavalis. 1999. Inference for unit roots in dynamic panels where the time dimension is fixed. *Journal of Econometrics* 91: 201–226.
- Im, K. S., M. H. Pesaran, and Y. Shin. 2003. Testing for unit roots in heterogeneous panels. *Journal of Econometrics* 115: 53–74.

- Levin, A., C.-F. Lin, and C.-S. J. Chu. 2002. Unit root tests in panel data: Asymptotic and finite-sample properties. *Journal of Econometrics* 108: 1–24.
- Paramati, S. R., N. Apergis, and M. Ummalla. 2017. Financing clean energy projects through domestic and foreign capital: The role of political cooperation among the EU, the G20 and OECD countries. *Energy Economics* 61: 62–71.
- Paramati, S. R., M. Ummalla, and N. Apergis. 2016. The effect of foreign direct investment and stock market growth on clean energy use across a panel of emerging market economies. *Energy Economics* 56: 29–41.
- Pesaran, M. H. 2007. A simple panel unit root test in the presence of cross-section dependence. *Journal of Applied Econometrics* 22: 265–312.
- Salahuddin, M., K. Alam, and I. Ozturk. 2016. The effects of Internet usage and economic growth on CO₂ emissions in OECD countries: A panel investigation. *Renewable and Sustainable Energy Reviews* 62: 1226–1235.
- Stine, R. A. 1987. Estimating properties of autoregressive forecasts. *Journal of the American Statistical Association* 82: 1072–1078.

About the authors

Luciano Lopez is a PhD student at the University of Neuchâtel (Switzerland). He is currently writing a PhD thesis in development economics.

Sylvain Weber is a postdoctoral researcher at the Institute of Economic Research of the University of Neuchâtel (Switzerland). His main fields of research are labor and energy economics, and he is interested in microeconometrics in general.

Response surface models for the Elliott, Rothenberg, and Stock unit-root test

Jesús Otero
Universidad del Rosario
Bogotá, Colombia
jesus.otero@urosario.edu.co

Christopher F. Baum
Boston College
Chestnut Hill, MA
baum@bc.edu

Abstract. In this article, we present response surface coefficients for a large range of quantiles of the Elliott, Rothenberg, and Stock (1996, *Econometrica* 64: 813–836) unit-root tests, for different combinations of number of observations, T , and lag order in the test regressions, p , where the latter can either be specified by the user or be endogenously determined. The critical values depend on the method used to select the number of lags. We present the command `ersur` and illustrate its use with an empirical example that tests the validity of the expectations hypothesis of the term structure of interest rates.

Keywords: `st0508`, `ersur`, Elliott, Rothenberg, Stock, unit-root test, Monte Carlo, response surface, critical values, lag length, p -values

1 Introduction

Since Nelson and Plosser (1982), testing for the presence of a unit root has become standard practice in the empirical analysis of economic time series. Among the tests available in the literature, the Said and Dickey (1984) unit-root test, based on extending Dickey and Fuller (1979) and commonly referred to as ADF, continues to be a favorite procedure of applied researchers. This is probably because the regression-based ADF test can be easily computed. However, a common criticism is that the ADF test exhibits disappointing power properties, as shown, for example, in the Monte Carlo simulations performed by DeJong et al. (1992).

During the last three decades, there have been three main research programs in the econometrics literature that aim to overcome the low power problem. First, some authors have continued developing more-powerful modifications of the univariate ADF test, including the generalized least squares (GLS)-ADF test of Elliott, Rothenberg, and Stock (1996), who use conditional GLS, and the ADF-max test of Leybourne (1995), who suggests taking the maximum of two ADF test statistics calculated using both forward and reversed data. Second, testing for unit roots in panel data has also been considered an alternative way to achieve power gains over unit-root tests applied to a single time series. This is because panel data, by combining information from the time-series dimension with that from the cross-section dimension, require fewer time observations for the tests to exhibit power. Among the panel unit-root tests available in the literature, perhaps those put forward by Im, Pesaran, and Shin (2003) and Pesaran (2007b) have proven to be the most popular. Third, authors such as Kapetanios, Shin, and Snell (2003) and

Kapetanios and Shin (2008) have considered tests of the unit-root hypothesis against the alternative of a globally stationary exponential smooth-transition autoregressive process.

Focusing on the first approach, Elliott, Rothenberg, and Stock (1996) propose a modified version of the ADF unit-root test—called the ERS test—that has substantially improved power in the presence of an unknown intercept or trend. Elliott, Rothenberg, and Stock further show that while the t statistic calculated from the GLS-demeaned data has an identical limiting representation to that of the conventional Dickey–Fuller t statistic when there is no intercept, the limiting representation differs in the linear trend case. To apply the test, Elliott, Rothenberg, and Stock tabulate, via stochastic simulation, asymptotic critical values (CVs) based on $T = 50, 100, 200,$ and ∞ time observations.¹ In subsequent work, Cheung and Lai (1995b) examine the sensitivity of CVs to the sample size through response surface regressions that account for the effect of varying the number of observations, T , and the number of lags of the dependent variable, p . However, these CVs do not allow for their possible dependence on the criterion used to select the optimal number of lags.

In this article, we undertake an extensive set of Monte Carlo simulations, summarized through response surface regressions, to calculate finite-sample CVs and approximate p -values of the ERS unit-root test. The simulation experiments not only allow for the presence of stochastic processes with nonzero mean and nonzero trend, but also allow for the lag order to be either fixed or determined endogenously using a data-dependent procedure. We present the command `ersur`, which easily calculates the ERS test statistic, finite-sample CVs, and approximate p -values.

This article is organized as follows: Section 2 provides an overview of the ERS unit-root test. Section 3 presents the design of the Monte Carlo experiments. Section 4 reports the estimated response surfaces and describes the procedure to estimate the associated approximate p -values. Section 5 describes the `ersur` command. Section 6 illustrates the use of `ersur` with an empirical example based on interest-rate spreads. Section 7 concludes the article.

2 The Elliott, Rothenberg, and Stock test

Elliott, Rothenberg, and Stock (1996) propose a test for the null hypothesis of a unit root against the alternative of stationarity, available as the Stata command `dfgls` (see [TS] `dfgls`).² Assuming the presence of a nonzero trend in the underlying data, the ERS test is based on the t statistic that tests the null hypothesis that $a_0 = 0$ against the alternative hypothesis of stationarity $a_0 < 0$, in the following auxiliary regression:

-
1. Empirical applications of the ERS test include Pesaran (2007a) and Le Pen (2011) for output convergence; Pesaran et al. (2009) for purchasing power parity; and Abbott and De Vita (2012) for house price convergence.
 2. The original version of `dfgls` on the Statistical Software Components archive was written by C. F. Baum and Richard Sperling for Stata 6.0.

$$\Delta y_t^d = a_0 y_{t-1}^d + b_1 \Delta y_{t-1}^d + \dots + b_p \Delta y_{t-p}^d + \varepsilon_t \tag{1}$$

where p lags of the dependent variable are included to account for residual serial correlation, and y_t^d is the GLS-detrended version of the original series y_t , that is,

$$y_t^d = y_t - \hat{\beta}_0 - \hat{\beta}_1 t$$

where $\hat{\beta}_0$ and $\hat{\beta}_1$ are obtained through an ordinary least-squares (OLS) regression of \bar{y} against \bar{w} , where

$$\begin{aligned} \bar{y} &= \{y_1, (1 - \bar{\rho}L) y_2, \dots, (1 - \bar{\rho}L) y_T\} \\ \bar{w} &= \{w_1, (1 - \bar{\rho}L) w_2, \dots, (1 - \bar{\rho}L) w_T\} \\ \bar{\rho} &= 1 + \frac{\bar{c}}{T} \end{aligned}$$

and $w_t = (1, t)$ contains the deterministic components.

Elliott, Rothenberg, and Stock (1996) recommend setting $\bar{c} = -13.5$ to obtain the best results in terms of the power of the test. The CVs of the test for trended data were tabulated by Elliott, Rothenberg, and Stock in table 1 for $T = 50, 100, 200,$ and ∞ . Cheung and Lai (1995b) present response surface coefficients that allow for changing T and exogenously determined p .

In the model with no trend, the GLS-demeaned version of the original series y_t is obtained as

$$y_t^d = y_t - \hat{\beta}_0$$

where $\hat{\beta}_0$ is obtained through an OLS regression of \bar{y} against \bar{w} , where

$$\begin{aligned} \bar{y} &= \{y_1, (1 - \bar{\rho}L) y_2, \dots, (1 - \bar{\rho}L) y_T\} \\ \bar{w} &= \{w_1, (1 - \bar{\rho}L) w_2, \dots, (1 - \bar{\rho}L) w_T\} \\ \bar{\rho} &= 1 + \frac{\bar{c}}{T} \end{aligned}$$

and $w_t = (1)$ contains the deterministic component. Elliott, Rothenberg, and Stock (1996) recommend setting $\bar{c} = -7$. The CVs correspond to those originally tabulated by Dickey and Fuller (1979) for the model with no constant; see also MacKinnon (1991) and Cheung and Lai (1995a).

3 Monte Carlo experiment design

The design of the Monte Carlo simulation experiment follows Otero and Smith (2012). Assume that y_t is generated by an autoregressive process of order 1:

$$y_t = y_{t-1} + \varepsilon_t$$

where $\varepsilon_t \sim N(0, 1)$ and $t = 1, \dots, T + 1$. Simulation experiments are carried out for a total of 56 different sample sizes, with $T = 18(2)62, 65(5)100, 110(10)200, 220(20)300, 350(50)500, 600(100)800, 1000, 1400,$ and 2000 , where, for example, $18(2)62$ means that all samples from $T = 18$ to $T = 62$ increasing in steps of 2 are accounted for. The same

notation is used later when listing significance levels. The time series y_t is generated by setting an initial value $y_{-99} = 0$, and then the first 100 observations are discarded. Each experiment consists of 50,000 Monte Carlo replications. The number of lagged differences of the dependent variable, p , is set equal to $p = 0, 1, \dots, 8$. For $T \leq 20$, $p \leq 1$ is used; for $22 \leq T \leq 24$, $p \leq 2$ is used; for $26 \leq T \leq 28$, $p \leq 3$ is used; for $30 \leq T \leq 32$, $p \leq 4$ is used; for $34 \leq T \leq 36$, $p \leq 6$ is used; and for $T > 36$, all values of p are used. Overall, there will be 456 different pairings of T and p .

To account for sampling variability, the setup outlined above is repeated 50 times, implying that there will be 50 CVs of the test for each combination of number of observations, T , and lag truncation, p . Following earlier work by MacKinnon (1991), CVs are calculated at each of 221 significance levels ($l = 0.0001, 0.0002, 0.0005, 0.001(0.001)0.01, 0.015(0.005)0.990, 0.991(0.001)0.999, 0.9995, 0.9998, \text{ and } 0.9999$) of the ERS t statistic for two cases, namely, a nonzero mean process (demeaned data) and a nonzero trend process (detrended data).

Using the simulated CVs, we subsequently fit response surface models at each of the $l = 221$ significance levels. The choice of the response surface functional form follows authors such as MacKinnon (1991), Cheung and Lai (1995a,b), and Harvey and van Dijk (2006), in which the CVs are regressed on an intercept term and on power functions of $1/T$ and p/T . The functional form that is finally selected is

$$CV_{T,p}^l = \theta_\infty^l + \sum_{i=1}^4 \theta_i^l \left(\frac{1}{T}\right)^i + \sum_{i=1}^4 \phi_i^l \left(\frac{p^i}{T}\right) + \epsilon^l \quad (2)$$

where $CV_{T,p}^l$ is the CV estimate at significance level l ; T refers to the number of observations on Δy_t , which is one less than the total number of available observations; and p is the number of lags of the dependent variable that are included to account for residual serial correlation.³ It is worth noticing that the functional form in (2) is such that the larger the number of observations, T , the weaker the CVs' dependence on the lag truncation, p . In addition, as $T \rightarrow \infty$, the intercept term θ_∞^l can be thought of as an estimate of the corresponding asymptotic CV.

4 Main results

Tables 1 and 2 report response surface regression estimates for 3 of the 221 significance levels, namely, $l = 0.01, 0.05, \text{ and } 0.10$ for demeaned and detrended data, respectively. These estimates can be used to obtain CVs for any given T and fixed lag order p . However, in practice, the lag order p is rarely fixed by the user and instead is chosen endogenously using a data-dependent procedure. Thus, we also use information criteria such as Akaike and Schwarz, which we denote as AIC and SIC, respectively. Here the optimal number of lags is determined by varying p in regression (1) between p_{\max}

3. Experimenting with even higher powered terms generally yielded coefficients that were not statistically different from 0 at the 1% significance level nor led to any noticeable increase in the \bar{R}^2 for these models.

and $p_{\min} = 0$ lags, and choosing the best model according to the information criterion being used. We also consider another data-dependent procedure, which is commonly referred to as the general-to-specific (GTS) algorithm, to optimally select p . This algorithm, advocated by [Campbell and Perron \(1991\)](#), [Hall \(1994\)](#), and [Ng and Perron \(1995\)](#), starts by setting some upper bound on p , say, p_{\max} , where $p_{\max} = 0, 1, 2, \dots, 8$, estimating (1) with $p = p_{\max}$, and testing the statistical significance of $b_{p_{\max}}$. If this coefficient is statistically significant, for instance, using a significance level of 5% (denoted GTS_5) or 10% (denoted GTS_{10}), one chooses $p = p_{\max}$. Otherwise, the order of the estimated autoregression in (1) is reduced by 1 until the coefficient on the last included lag is statistically different from 0. Finally, for AIC, SIC, GTS_5 , and GTS_{10} , the same 221 quantiles of the empirical small-sample distribution are recorded as before, but the response surface regressions given in (2) are estimated using p_{\max} instead of p lags.

Table 1. Response surface estimates for demeaned data

Lags	l	Intercept	(Std. Err.)	$1/T$	$1/T^2$	$1/T^3$	$1/T^4$	p/T	p^2/T	p^3/T	p^4/T	R^2
Fixed	0.01	-2.569	(0.0004)	-18.779	151.9	-814.6	-9349.3	-0.306	1.059	-0.214	0.013	0.979
	0.05	-1.942	(0.0002)	-22.761	427.3	-6357.6	33186.4	0.057	0.764	-0.155	0.010	0.990
	0.10	-1.617	(0.0002)	-25.453	555.3	-8898.6	53120.1	0.237	0.618	-0.128	0.008	0.994
AIC	0.01	-2.568	(0.0004)	-19.642	182.4	-939.8	-9367.2	-5.556	1.474	-0.205	0.011	0.994
	0.05	-1.942	(0.0002)	-21.823	369.8	-5570.6	33264.7	-3.440	1.033	-0.144	0.007	0.997
	0.10	-1.617	(0.0001)	-24.485	501.9	-8301.2	54686.0	-2.388	0.804	-0.115	0.006	0.998
SIC	0.01	-2.569	(0.0004)	-15.966	39.0	-1251.2	19847.5	-4.668	1.619	-0.227	0.011	0.991
	0.05	-1.941	(0.0002)	-21.573	414.6	-8169.9	62936.0	-2.453	1.036	-0.155	0.008	0.997
	0.10	-1.617	(0.0001)	-24.784	563.9	-10656.2	77980.7	-1.493	0.748	-0.115	0.006	0.998
GTS ₅	0.01	-2.566	(0.0003)	-21.472	395.0	-7654.8	52031.5	-3.304	0.735	-0.104	0.006	0.993
	0.05	-1.940	(0.0002)	-23.286	469.9	-7592.0	45976.4	-1.219	0.251	-0.037	0.002	0.997
	0.10	-1.616	(0.0001)	-25.483	551.6	-8809.1	54251.6	-0.472	0.108	-0.018	0.001	0.998
GTS ₁₀	0.01	-2.565	(0.0004)	-22.210	419.0	-7449.8	44977.9	-4.669	1.120	-0.152	0.008	0.993
	0.05	-1.940	(0.0002)	-22.503	401.2	-5768.6	30522.2	-2.341	0.545	-0.074	0.004	0.997
	0.10	-1.617	(0.0001)	-24.387	466.4	-6762.8	38261.7	-1.372	0.334	-0.046	0.003	0.998

Table 2. Response surface estimates for detrended data

Lags	l	Intercept	(Std. Err.)	$1/T$	$1/T^2$	$1/T^3$	$1/T^4$	p/T	p^2/T	p^3/T	p^4/T	R^2
Fixed	0.01	-3.405	(0.0004)	-23.650	251.5	-4110.3	9281.0	0.404	1.327	-0.269	0.017	0.984
	0.05	-2.844	(0.0003)	-23.823	477.7	-8786.7	50347.0	0.532	1.060	-0.215	0.014	0.985
	0.10	-2.555	(0.0003)	-24.288	563.5	-10354.1	63554.9	0.584	0.922	-0.187	0.012	0.985
AIC	0.01	-3.399	(0.0005)	-32.582	807.0	-14439.0	72508.5	-6.624	1.676	-0.233	0.012	0.992
	0.05	-2.844	(0.0002)	-23.279	368.0	-4832.1	15025.6	-5.365	1.416	-0.193	0.010	0.997
	0.10	-2.557	(0.0002)	-22.079	377.0	-5816.4	32017.2	-4.522	1.270	-0.171	0.009	0.998
SIC	0.01	-3.412	(0.0004)	-12.420	-627.9	15994.2	-127428.4	-6.018	1.812	-0.233	0.011	0.995
	0.05	-2.847	(0.0002)	-16.716	48.8	-2174.0	30804.8	-4.392	1.694	-0.240	0.012	0.997
	0.10	-2.558	(0.0002)	-18.829	262.8	-6505.4	60506.8	-3.319	1.487	-0.222	0.011	0.998
GTS ₅	0.01	-3.403	(0.0004)	-29.021	739.2	-17781.2	132079.9	-5.268	1.261	-0.173	0.009	0.993
	0.05	-2.842	(0.0002)	-25.286	632.8	-13630.8	97408.7	-2.697	0.583	-0.082	0.005	0.996
	0.10	-2.555	(0.0002)	-24.703	615.9	-12227.4	84332.6	-1.469	0.287	-0.043	0.003	0.996
GTS ₁₀	0.01	-3.403	(0.0004)	-32.813	982.4	-22381.6	160071.8	-6.414	1.631	-0.217	0.011	0.992
	0.05	-2.841	(0.0003)	-26.826	714.7	-14731.7	100113.5	-4.423	1.090	-0.145	0.008	0.995
	0.10	-2.554	(0.0002)	-25.201	633.2	-12149.3	79983.8	-3.213	0.758	-0.099	0.005	0.996

We estimate 2,210 response surface regressions: 2 models multiplied by 5 criteria to select p multiplied by the 221 significance levels. The chosen functional form performs very well, with an average coefficient of determination of 0.994; in only 36 (out of 2,210) cases, it was below 0.95.

Tables 3 and 4 report the CVs estimated from the response surface models for selected values of T and p . For comparison purposes, we also include the CVs of the ERS test. As can be seen from the tables, for $T = 1000$ the implied asymptotic CVs from the response surface models fit in this article are close to those obtained by Elliott, Rothenberg, and Stock (1996). Interestingly, the implied CVs also exhibit dependence on the method used to select the lag length, and in some cases, the differences may be noticeable, especially when T and l are small and p is large. In particular, for a given T , the implied CVs from the response surfaces decrease (in an absolute sense) in p when the augmentation order is fixed by the user, while they increase (in an absolute sense) in p_{\max} when it is optimally determined using any of the data-dependent procedures being considered.

Table 3. Lag order and finite-sample CVs for demeaned data

Criterion to choose lag	p/T	$l = 0.01$						$l = 0.05$						$l = 0.10$						
		100		200		1,000		100		200		1,000		100		200		1,000		
Fixed	0	-2.74	-2.66	-2.59	-2.13	-2.05	-1.96	-1.82	-1.73	-1.64										
	2	-2.72	-2.65	-2.59	-2.11	-2.04	-1.96	-1.80	-1.72	-1.64										
	4	-2.69	-2.63	-2.58	-2.08	-2.02	-1.96	-1.78	-1.71	-1.64										
	6	-2.67	-2.62	-2.58	-2.06	-2.01	-1.96	-1.76	-1.70	-1.64										
AIC	0	-2.75	-2.66	-2.59	-2.13	-2.04	-1.96	-1.82	-1.73	-1.64										
	2	-2.81	-2.70	-2.59	-2.17	-2.06	-1.97	-1.84	-1.74	-1.64										
	4	-2.84	-2.71	-2.60	-2.17	-2.07	-1.97	-1.85	-1.74	-1.64										
	6	-2.86	-2.72	-2.60	-2.18	-2.07	-1.97	-1.84	-1.74	-1.64										
SIC	0	-2.73	-2.65	-2.58	-2.12	-2.04	-1.96	-1.82	-1.73	-1.64										
	2	-2.77	-2.67	-2.59	-2.14	-2.05	-1.96	-1.83	-1.73	-1.64										
	4	-2.77	-2.67	-2.59	-2.13	-2.05	-1.96	-1.82	-1.73	-1.64										
	6	-2.77	-2.67	-2.59	-2.13	-2.04	-1.96	-1.81	-1.72	-1.64										
GTS ₅	0	-2.75	-2.66	-2.59	-2.13	-2.05	-1.96	-1.82	-1.73	-1.64										
	2	-2.79	-2.69	-2.59	-2.15	-2.05	-1.96	-1.83	-1.73	-1.64										
	4	-2.82	-2.70	-2.59	-2.16	-2.06	-1.97	-1.83	-1.74	-1.64										
	6	-2.83	-2.71	-2.60	-2.17	-2.06	-1.97	-1.84	-1.74	-1.64										
GTS ₁₀	0	-2.75	-2.67	-2.59	-2.13	-2.04	-1.96	-1.82	-1.73	-1.64										
	2	-2.81	-2.70	-2.59	-2.16	-2.06	-1.97	-1.84	-1.74	-1.64										
	4	-2.84	-2.71	-2.60	-2.17	-2.07	-1.97	-1.84	-1.74	-1.64										
	6	-2.85	-2.72	-2.60	-2.18	-2.07	-1.97	-1.85	-1.74	-1.64										
ERS	0	-2.59	-2.58	-2.57	-1.94	-1.94	-1.94	-1.61	-1.62											

Note: ERS denotes the MacKinnon (1991) CVs for the no-constant case.

MacKinnon (1994, 1996) points out that the residuals of the estimated response surfaces are expected to exhibit heteroskedasticity. Thus, to evaluate the robustness of the OLS results, we also considered estimation using the generalized method of moments procedure outlined by MacKinnon, which in the context of our simulation exercise averages the CVs across the 50 replications for each combination of T and p , and scaling all the variables in (2) by the standard error in these replications. Then OLS can be used to estimate the resulting equation using the rescaled variables. The results of applying this generalized method of moments procedure produces qualitatively similar results to those obtained with OLS, so they are not reported here.

To obtain approximate p -values of the ERS statistic, we follow MacKinnon (1994, 1996) by estimating the regression

$$\Phi^{-1}(l) = \gamma_0^l + \gamma_1^l \widehat{CV}^l + \gamma_2^l (\widehat{CV}^l)^2 + v^l \tag{3}$$

where Φ^{-1} is the inverse of the cumulative standard normal distribution at each of the 221 quantiles, and \widehat{CV}^l is the fitted value from (2) at the l quantile. Following Harvey and van Dijk (2006), (3) is estimated by OLS using 15 observations, made up of the actual quantile and the 7 quantile observations on either side of the desired quantile.⁴ Approximate p -values of the ERS test statistic can then be obtained as

$$p\text{-value} = \Phi \left[\widehat{\gamma}_0^l + \widehat{\gamma}_1^l \text{ERS}(p) + \widehat{\gamma}_2^l \{\text{ERS}(p)\}^2 \right]$$

where $\widehat{\gamma}_0^l$, $\widehat{\gamma}_1^l$, and $\widehat{\gamma}_2^l$ are the OLS parameter estimates from (3).

Finally, it is worth noting that in all the Monte Carlo simulation experiments, the error term was assumed to be white noise, which to some extent might be regarded as a weakness, because the error process can be quite general. To assess whether the estimated response surfaces continue to be reliable as we diverge from this white noise error specification, we carried out an additional set of simulations in which we computed the 1%, 5%, and 10% estimated CVs of the ERS statistics (for demeaned and detrended data) for two alternative specifications of the error process. The first follows a first-order autoregressive [AR(1)] process with an autoregressive coefficient of 0.9, while the second follows a first-order moving-average [MA(1)] process with a negative coefficient of -0.5 . The sample sizes considered in these additional Monte Carlo simulations were $T = 100, 200,$ and 400 observations, and the number of lags of the dependent variable in the test regression was assumed to be exogenously determined varying between 0 and 8 lags.

Taking the AR and MA error specifications as correct, we then proceeded to calculate, in percentage terms, how the CVs vary from those computed under the assumption of white noise errors. Averaging across T for the two model specifications, our findings (for brevity, not reported here but available upon request) indicate that if errors are AR(1) with an autoregressive coefficient equal to 0.9, the zero-lag CVs for white noise are seriously biased, but for lags 1 to 8, the percentage bias is never more than 1.5%

4. For $l \leq 0.004$ and $l \geq 0.996$, we use the actual quantile and the 14 observations closest to the desired quantile, because there will not be 7 observations on either side.

(and is generally lower). For the MA(1) specification with a coefficient equal to -0.5 , the bias for lags equal to 0, 1, and 2 is sizable but less than 10% for higher lags.

These additional results highlight the importance of capturing serial correlation patterns that may be present in the error term of the series under consideration, while indicating that, for reasonable choices of the lag parameter, the CVs reported by our command are robust to misspecification of the error process.

5 The `ersur` command

The command `ersur` calculates the ERS test statistic; its associated finite-sample CVs for $l = 0.01, 0.05, \text{ and } 0.10$; and its approximate p -values. The estimation of CVs and approximate p -values permits different combinations of number of observations, T , and lag order in the test regression, p , where the latter can be either specified by the user or optimally selected using a data-dependent procedure.

5.1 Syntax

Before using the command `ersur`, and similar to other Stata time-series commands, it is necessary to `tsset` the data. Then,

```
ersur varname [if] [in] [, noprint maxlag(integer) trend]
```

varname may not contain gaps. *varname* may contain time-series operators. The command may be applied to one unit of a panel.

5.2 Options

`noprint` specifies that the results be returned but not printed.

`maxlag(integer)` sets the number of lags to be included in the test regression to account for residual serial correlation. By default, `ersur` sets the number of lags following [Schwert \(1989\)](#), with the formula `maxlag() = int{12(T/100)0.25}`, where T is the total number of observations.

`trend` specifies the modeling of intercepts and trends. By default, `ersur` considers *varname* to be a nonzero mean stochastic process; in this case, [Elliott, Rothenberg, and Stock \(1996\)](#) recommend demeaning the data using GLS. If the `trend` option is specified, `ersur` assumes that *varname* is a nonzero trend stochastic process, in which case [Elliott, Rothenberg, and Stock](#) recommend detrending the data using GLS.

5.3 Stored results

`ersur` stores the following in `r()`:

Scalars			
<code>r(N)</code>	number of observations	<code>r(maxp)</code>	last time period used in the
<code>r(minp)</code>	first time period used in the test regression		test regression
Macros			
<code>r(varname)</code>	variable name	<code>r(tsfmt)</code>	time-series format of the
<code>r(treat)</code>	demeaned or detrended, depending on the <code>trend</code> option		time variable
Matrices			
<code>r(results)</code>	results matrix, 5 x 6		

6 Empirical application

The expectations theory of the term structure of interest rates implies that interest rates of different maturities maintain a long-run equilibrium relationship, so the interest-rate spread does not exhibit a tendency to grow systematically over time; see, for instance, [Campbell and Shiller \(1987\)](#), [Stock and Watson \(1988\)](#), and [Hall, Anderson, and Granger \(1992\)](#) for early applications. This is essentially a question of whether interest rate spreads, defined as the differences between long-term and short-term interest rates, may be characterized as stationary stochastic processes.

In this section, we illustrate the use of the `ersur` command to address this question. We use monthly data on the United States Treasury interest rate series at nine maturities over the sample period 1993m10 to 2013m3, which yields a total of 234 time observations for each series. The specific maturities considered in the analysis correspond to the 3-month, 6-month, 1-year, 3-year, 5-year, 7-year, 10-year, 20-year, and 30-year constant maturity rates, as retrieved from the Federal Reserve Economic Data provided by the Economic Research Division of the Federal Reserve Bank of St. Louis.⁵ The interest rates are denoted r_3 , r_6 , r_{12} , r_{36} , r_{60} , r_{84} , r_{120} , r_{240} , and r_{360} .

We begin by loading the dataset and declaring that it has a time-series format:

```
. use usrates
. tsset date, monthly
      time variable:  date, 1993m10 to 2013m3
      delta:        1 month
```

Next suppose we want to test whether the interest rate spread between r_6 and r_3 (which we shall denote as s_6) contains a unit root against the alternative, that it is a stationary process. Given that s_6 has a nonzero mean, the relevant ERS statistic is based on GLS demeaned data, the default for `ersur`. Setting $p = 3$ lags, the results of applying the command `ersur` are as follows:

5. The task of downloading the time series from the Federal Reserve Economic Data database was greatly simplified using the command `freduse`; see [Drukker \(2006\)](#).

```
. ersur s6, maxlag(3)
Elliott, Rothenberg & Stock (1996) test results for 1994m2 - 2013m3
Variable name: s6
Ho: Unit root
Ha: Stationarity
GLS demeaned data
```

Criteria	Lags	ERS stat.	p-value	1% cv	5% cv	10% cv
FIXED	3	-3.780	0.000	-2.630	-2.016	-1.702
AIC	0	-4.298	0.000	-2.684	-2.048	-1.725
SIC	0	-4.298	0.000	-2.656	-2.033	-1.715
GTS05	0	-4.298	0.000	-2.676	-2.042	-1.720
GTS10	2	-3.562	0.001	-2.685	-2.046	-1.723

Table 5 summarizes the results of applying the ERS test to $s_6 = r_6 - r_3$, $s_{12} = r_{12} - r_3$, and so on, until $s_{360} = r_{360} - r_3$, where the interest rate spreads have been previously demeaned using GLS. We set $p = 3$ when the lag length is fixed and $p_{\max} = 3$ when it is optimally determined. All in all, the ERS test results support the validity of the term structure of interest rates. However, in the case of the longest maturity differential between short- and long-run rates, that is, s_{360} , we fail to reject the presence of a unit root in the corresponding spread at the 5% significance level.

Table 5. Applying the ERS test to interest rate differentials

Variable		Fixed	AIC	SIC	GTS ₅	GTS ₁₀
<i>s</i> ₆	Lags	3	0	0	0	2
	ERS test	-3.780	-4.298	-4.298	-4.298	-3.562
	<i>p</i> -value	[0.000]	[0.000]	[0.000]	[0.000]	[0.001]
<i>s</i> ₁₂	Lags	3	3	1	3	3
	ERS test	-4.013	-4.013	-3.904	-4.013	-4.013
	<i>p</i> -value	[0.000]	[0.000]	[0.000]	[0.000]	[0.000]
<i>s</i> ₂₄	Lags	3	1	1	1	3
	ERS test	-3.574	-3.504	-3.504	-3.504	-3.574
	<i>p</i> -value	[0.000]	[0.001]	[0.001]	[0.001]	[0.001]
<i>s</i> ₃₆	Lags	3	1	1	1	3
	ERS test	-3.262	-3.078	-3.078	-3.078	-3.262
	<i>p</i> -value	[0.001]	[0.003]	[0.003]	[0.003]	[0.002]
<i>s</i> ₆₀	Lags	3	3	1	3	3
	ERS test	-2.739	-2.739	-2.528	-2.739	-2.739
	<i>p</i> -value	[0.007]	[0.009]	[0.014]	[0.008]	[0.009]
<i>s</i> ₈₄	Lags	3	3	1	3	3
	ERS test	-2.485	-2.485	-2.278	-2.485	-2.485
	<i>p</i> -value	[0.015]	[0.017]	[0.028]	[0.017]	[0.017]
<i>s</i> ₁₂₀	Lags	3	3	1	3	3
	ERS test	-2.258	-2.258	-2.065	-2.258	-2.258
	<i>p</i> -value	[0.028]	[0.030]	[0.046]	[0.030]	[0.030]
<i>s</i> ₂₄₀	Lags	3	3	3	3	3
	ERS test	-2.061	-2.061	-2.061	-2.061	-2.061
	<i>p</i> -value	[0.045]	[0.049]	[0.047]	[0.048]	[0.048]
<i>s</i> ₃₆₀	Lags	3	3	3	3	3
	ERS test	-1.983	-1.983	-1.983	-1.983	-1.983
	<i>p</i> -value	[0.054]	[0.058]	[0.056]	[0.057]	[0.058]

7 Conclusions

We fit response surface models for the CVs of the [Elliott, Rothenberg, and Stock \(1996\)](#) unit-root test. The models are fit as a function of the number of observations, T , and lags of the dependent variable in the test regressions, p , for 221 significance levels. The lag length can be determined either exogenously by the user or endogenously using a data-dependent procedure. The results suggest that the method used to select the order of the augmentation affects the finite-sample CVs.

The command `ersur` can easily be used to calculate the ERS test statistic, finite-sample CVs, and approximate p -values. As an empirical application, `ersur` is illustrated by examining whether the theory of the term structure of interest rates holds among a set of U.S. interest rates.

8 Acknowledgments

Jesús Otero would like to acknowledge financial support received from the Universidad del Rosario through Fondo de Investigaciones de la Universidad del Rosario. We are also grateful to an anonymous referee and participants at the 2017 Stata Conferences in Baltimore and London for useful comments and suggestions. The usual disclaimer applies.

9 References

- Abbott, A., and G. De Vita. 2012. Pairwise convergence of district-level house prices in London. *Urban Studies* 49: 721–740.
- Campbell, J. Y., and P. Perron. 1991. Pitfalls and opportunities: What macroeconomists should know about unit roots. *NBER Macroeconomics Annual* 1991 6: 141–201.
- Campbell, J. Y., and R. J. Shiller. 1987. Cointegration and tests of present value models. *Journal of Political Economy* 95: 1062–1088.
- Cheung, Y.-W., and K. S. Lai. 1995a. Lag order and critical values of the augmented Dickey–Fuller test. *Journal of Business and Economic Statistics* 13: 277–280.
- . 1995b. Lag order and critical values of a modified Dickey–Fuller test. *Oxford Bulletin of Economics and Statistics* 57: 411–419.
- DeJong, D. N., J. C. Nankervis, N. E. Savin, and C. H. Whiteman. 1992. The power problems of unit root test in time series with autoregressive errors. *Journal of Econometrics* 53: 323–343.
- Dickey, D. A., and W. A. Fuller. 1979. Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association* 74: 427–431.

- Drukker, D. M. 2006. Importing Federal Reserve economic data. *Stata Journal* 6: 384–386.
- Elliott, G., T. J. Rothenberg, and J. H. Stock. 1996. Efficient tests for an autoregressive unit root. *Econometrica* 64: 813–836.
- Hall, A. 1994. Testing for a unit root in time series with pretest data-based model selection. *Journal of Business and Economic Statistics* 12: 461–470.
- Hall, A. D., H. M. Anderson, and C. W. J. Granger. 1992. A cointegration analysis of treasury bill yields. *Review of Economics and Statistics* 74: 116–126.
- Harvey, D. I., and D. van Dijk. 2006. Sample size, lag order and critical values of seasonal unit root tests. *Computational Statistics and Data Analysis* 50: 2734–2751.
- Im, K. S., M. H. Pesaran, and Y. Shin. 2003. Testing for unit roots in heterogeneous panels. *Journal of Econometrics* 115: 53–74.
- Kapetanios, G., and Y. Shin. 2008. GLS detrending-based unit root tests in nonlinear STAR and SETAR models. *Economics Letters* 100: 377–380.
- Kapetanios, G., Y. Shin, and A. Snell. 2003. Testing for a unit root in the nonlinear STAR framework. *Journal of Econometrics* 112: 359–379.
- Le Pen, Y. 2011. A pair-wise approach to output convergence between European regions. *Economic Modelling* 28: 955–964.
- Leybourne, S. J. 1995. Testing for unit roots using forward and reverse Dickey–Fuller regressions. *Oxford Bulletin of Economics and Statistics* 57: 559–571.
- MacKinnon, J. G. 1991. Critical values for cointegration tests. In *Long-Run Economic Relationships: Readings in Cointegration*, ed. R. F. Engle and C. W. J. Granger, 267–276. Oxford: Oxford University Press.
- . 1994. Approximate asymptotic distribution functions for unit-root and cointegration tests. *Journal of Business and Economic Statistics* 12: 167–176.
- . 1996. Numerical distribution functions for unit root and cointegration tests. *Journal of Applied Econometrics* 11: 601–618.
- Nelson, C. R., and C. I. Plosser. 1982. Trends and random walks in macroeconomic time series: Some evidence and implications. *Journal of Monetary Economics* 10: 139–162.
- Ng, S., and P. Perron. 1995. Unit root tests in ARMA models with data-dependent methods for the selection of the truncation lag. *Journal of the American Statistical Association* 90: 268–281.
- Otero, J., and J. Smith. 2012. Response surface models for the Leybourne unit root tests and lag order dependence. *Computational Statistics* 27: 473–486.

- Pesaran, M. H. 2007a. A pair-wise approach to testing for output and growth convergence. *Journal of Econometrics* 138: 312–355.
- . 2007b. A simple panel unit root test in the presence of cross-section dependence. *Journal of Applied Econometrics* 22: 265–312.
- Pesaran, M. H., R. P. Smith, T. Yamagata, and L. Hvozdnyk. 2009. Pairwise tests of purchasing power parity. *Econometric Reviews* 28: 495–521.
- Said, S. E., and D. A. Dickey. 1984. Testing for unit roots in autoregressive-moving average models of unknown order. *Biometrika* 71: 599–607.
- Schwert, G. W. 1989. Tests for unit roots: A Monte Carlo investigation. *Journal of Business and Economic Statistics* 7: 147–159.
- Stock, J. H., and M. W. Watson. 1988. Testing for common trends. *Journal of the American Statistical Association* 83: 1097–1107.

About the authors

Jesús Otero is a professor of economics at the Universidad del Rosario in Colombia. He received his PhD in economics from Warwick University in the United Kingdom. His field of study is applied time-series econometrics.

Christopher F. Baum is a professor of economics and social work at Boston College. He is an associate editor of the *Stata Journal* and the *Journal of Statistical Software*. Baum founded and manages the Boston College Statistical Software Components (SSC) archive at RePEc (<http://repec.org>). His recent research has addressed issues in social epidemiology and the labor market progress of refugees in developed economies.

Assessing the calibration of dichotomous outcome models with the calibration belt

Giovanni Nattino
Division of Biostatistics
College of Public Health
The Ohio State University
Columbus, OH
nattino.1@osu.edu

Stanley Lemeshow
Division of Biostatistics
College of Public Health
The Ohio State University
Columbus, OH

Gary Phillips
Center for Biostatistics
The Department of Biomedical Informatics
The Ohio State University
Columbus, OH

Stefano Finazzi
GiViTI Coordinating Center
Laboratory of Clinical Epidemiology
IRCCS Istituto di Ricerche Farmacologiche ‘Mario Negri’
Ranica, Italy

Guido Bertolini
GiViTI Coordinating Center
Laboratory of Clinical Epidemiology
IRCCS Istituto di Ricerche Farmacologiche ‘Mario Negri’
Ranica, Italy

Abstract. The calibration belt is a graphical approach designed to evaluate the goodness of fit of binary outcome models such as logistic regression models. The calibration belt examines the relationship between estimated probabilities and observed outcome rates. Significant deviations from the perfect calibration can be spotted on the graph. The graphical approach is paired to a statistical test, synthesizing the calibration assessment in a standard hypothesis testing framework. In this article, we present the `calibrationbelt` command, which implements the calibration belt and its associated test in Stata.

Keywords: gr0071, calibrationbelt, logistic regression, calibration, goodness of fit, binary outcome

1 Introduction

Statistical models that estimate the probability of binary outcomes are extremely common in many research areas. In particular, logistic regression is probably the most widely used method to generate such models. The reliability of binary outcome models requires two properties to be satisfied (Hosmer, Lemeshow, and Sturdivant 2013). First, a model must be able to distinguish between subjects within the two outcome levels. This property is the “discrimination”, which is usually evaluated with the area under the receiver operating characteristic curve. Second, the probabilities estimated by the model must accurately match the true outcome experienced in the data. This second property is the “calibration”, and it is the focus of the procedure described here.

The calibration belt is a graphical method that has been recently proposed to evaluate the calibration of binary outcome models. The methodology and its usefulness for calibration assessment are thoroughly described in previous works (Finazzi et al. 2011; Nattino, Finazzi, and Bertolini 2014b, 2016a). In this article, we describe the `calibrationbelt` command, which implements the calibration belt approach.

The calibration belt is a plot depicting the relationship between the model’s fit probabilities and the observed proportions of the response. Providing information about the statistical significance of the deviations, the calibration belt outperforms the commonly used graphical approaches such as locally weighted smoothers and plots of observed-expected events across deciles (Nattino, Finazzi, and Bertolini 2014a).

The information conveyed by the graphical approach is synthesized in a formal statistical test. Extensive simulations have shown good performances of the test under several scenarios (Nattino, Finazzi, and Bertolini 2014b, 2016a). Taken together, the calibration belt and the test statistic provide useful information when evaluating the performance of predictive models.

Notably, most of the calibration assessment methods have different assumptions in the internal and external validation settings (that is, whether the model has been fit to the same dataset upon which it is evaluated). The calibration belt approach is not an exception; two different procedures are used for the two contexts, and it is important to apply the correct method to identify model performance.

The rest of the article is organized as follows: In section 2, we explain the distinction between internal and external validation settings in assessing a model’s calibration. In section 3, we briefly describe the methodology to generate the calibration belt and the associated statistical test. In section 4, we describe the `calibrationbelt` command and apply it to a dataset of intensive care unit (ICU) patients. Finally, in section 5, we summarize the presented material.

2 Internal and external calibration assessment

The calibration of a model can be evaluated in two different settings. First, the model can be evaluated on the same dataset used to fit the model. This internal assessment

is an important step in the process of model development. Second, performance of an existing model can be evaluated in a new dataset unrelated to the original model development. Assessment of model calibration in a new independent sample is known as external validation.

It is important to distinguish these two frameworks because many calibration assessment procedures use different assumptions in the two cases. For example, consider the Hosmer–Lemeshow \widehat{C} test to evaluate the calibration of logistic regression models. If the data are partitioned into g groups, the statistic is distributed as a χ^2 with $g - 2$ degrees of freedom when models are evaluated on the developmental sample (Hosmer and Lemeshow 1980). However, when one applies models to independent samples, the degrees of freedom are g (Hosmer, Lemeshow, and Sturdivant 2013).

Like the Hosmer–Lemeshow test, the calibration test and belt have different assumptions in the two cases. The most important difference is in the type of models that can be evaluated with the proposed approach. Indeed, the calibration belt and test can be used to evaluate any kind of binary outcome model on external samples. However, only logistic regression models can be evaluated with these methods on the developmental dataset (Nattino, Finazzi, and Bertolini 2016a).

The second difference is in the functional form imposed by the proposed procedure. As will be described in section 3, the calibration belt and test are based on a polynomial regression. The degree of the polynomial link depends on whether the calibration is internally or externally evaluated. Further details are provided in the following section and in Nattino, Finazzi, and Bertolini (2016a).

It is therefore extremely important to recognize the setting where the model is going to be evaluated and to select the options of the `calibrationbelt` command accordingly.

3 The calibration belt and test

In this section, we provide an overview of the methodology behind the calibration belt approach. We consider a sample of size n , where each subject i is characterized by a binary outcome Y_i and by an estimate p_i of $P(Y_i = 1)$, the true probability of the positive outcome. We are interested in assessing the calibration of the model, that is, evaluating whether the estimates p_i are compatible with the true probabilities $P(Y_i = 1)$.

The way the probabilities p_i are generated depends on the setting. In the internal assessment case, the probabilities are the result of a model developed on the same sample. On the other hand, such probabilities are defined according to an independently developed model in the external validation case.

The approach is based on the estimation of the relationship between the predictions p_i and the true probabilities $P(Y_i = 1)$ with a polynomial logistic regression. In particular, the logit transformation of the predictions p_i is considered, and a logistic regression of the form

$$\text{logit}\{P(Y_i = 1)\} = \beta_0 + \beta_1 \text{logit}(p_i) + \cdots + \beta_m \{\text{logit}(p_i)\}^m \quad (1)$$

is fit. The logit function is defined as $\text{logit}(p) = \ln\{p/(1-p)\}$.

Notably, the relationship assumed in (1) requires the specification of the degree m of the polynomial. This is an important choice. The assumed relationship could be too simple to describe the real link between predictions and true probabilities if m were fixed at too small a value. Conversely, fixing m too high would lead to the estimation of several useless parameters whenever the relationship is well described by lower-order polynomials.

To overcome the problems associated with a fixed m , we base the procedure on a data-driven forward selection. A low-order polynomial is initially fit, and a sequence of likelihood-ratio tests is used to forwardly identify the degree m . In particular, the degree of the starting model depends on whether the calibration is evaluated internally or on an independent sample. In both cases, the simplest possible polynomial is considered in this stage. This corresponds to the first-order polynomial in the assessment of external calibration but not in the assessment of the developmental dataset. A first-order polynomial is not informative in the latter scenario, because its parameters β_0 and β_1 would always assume the values 0 and 1, regardless of the calibration of the model (Nattino, Finazzi, and Bertolini 2016a). Therefore, the forward selection starts by fitting a second-order polynomial.

Once m is selected, the fit relationship provides information about the calibration of the predictions p_i . Indeed, by definition, a model is perfectly calibrated if $p_i = P(Y_i = 1)$ for each $i = 1, \dots, n$. Under the link assumed in (1), this identity corresponds to the configuration of the parameters $\beta_0 = \beta_2 = \cdots = \beta_m = 0$ and $\beta_1 = 1$. The idea of the approach is to compare the relationship estimated by fitting the model in (1) with the perfect-calibration link corresponding to the aforementioned choice of the parameters. This comparison can be performed statistically or graphically.

A likelihood-ratio test evaluating the hypothesis $H_0: (\beta_0, \beta_1, \beta_2, \dots, \beta_m) = (0, 1, 0, \dots, 0)$ versus the complementary alternative can be used to formally test the calibration of the model. Notably, the distribution of the likelihood-ratio statistic must account for the forward process used to select m . Nattino, Finazzi, and Bertolini (2014b, 2016a) provide the derivation of the distribution of the statistic in the external and internal calibration assessment, respectively.

To graphically assess the calibration, we can represent the fit relationship between the predictions p_i and the true probabilities $P(Y_i = 1)$ with a curve. This curve can be compared with the line associated with the identity of the two quantities, that is, the bisector of the quadrant (45-degree line). A confidence band around the curve, namely, the calibration belt, reflects the statistical uncertainty about the estimate of the curve

and allows for the evaluation of the significance of the deviations from the line of perfect calibration.

Because the statistical test is based on a likelihood-ratio statistic, the most natural way to define the confidence band is to invert this type of test. Such inversion guarantees the ideal parallelism between the formal statistical test and calibration belt. However, the calculations involved in the inversion of the likelihood-ratio test are computationally intensive for large samples. Fortunately, likelihood-ratio and Wald confidence regions are asymptotically equivalent, and the computations to construct Wald confidence regions are much simpler (Cox and Hinkley 1974). Thus, the construction of the calibration belt plot is implemented using two different algorithms depending on the sample size. If the sample is smaller than 10,000 records, the confidence band is based on the inversion of the likelihood-ratio test, so the same statistical framework is considered for the statistical test and confidence region in small-moderate samples. For samples of 10,000 units or larger, the confidence region is based on the computationally simpler Wald confidence region.

4 The calibrationbelt command

4.1 Syntax

The `calibrationbelt` command is invoked with the following syntax:

```
calibrationbelt [ varlist ] [ if ] [ , devel(string) cLevel1(#) cLevel2(#)
  nPoints(#) maxDeg(#) thres(#) ]
```

The command generates the calibration belt plot and computes the associated statistical test. There are two ways to apply the procedure.

The first way is to pass the variables corresponding to the binary response and to the predicted probabilities of the outcome in the *varlist* (in this order). In this case, the option `devel(string)` must be specified, reporting whether the predictions have been fit on the dataset under evaluation (`devel("internal")`) or if the assessment consists of an external, independent validation (`devel("external")`). For example, if the variables `Y` and `phat` store the dependent variable of the model and the predicted probabilities fit on the same sample where the calibration assessment is performed, the command can be run with

```
calibrationbelt Y phat, devel("internal")
```

The second way to use `calibrationbelt` is after fitting a logistic regression model (using either the `logit` or `logistic` command). In this case, it is possible to simply run `calibrationbelt` without specifying the argument *varlist* or any other option. The command automatically considers the dependent variable and the predictions of the logistic regression fit as the arguments to be used. Moreover, the procedure assumes that the setting is the assessment of internal calibration.

4.2 Options

`devel(string)` specifies whether the calibration is evaluated on the same dataset used to fit the model (`devel("internal")`) or is evaluated on external independent samples (`devel("external")`). Depending on whether the `varlist` argument is passed to the command or not, the program may force the user to specify the setting. For further details about internal and external assessment, see section 2.

`cLevel1(#)` sets one of the confidence levels to be considered in the calibration belt plot. A second confidence level can be set with the argument `cLevel2(#)`. The defaults are `cLevel1(95)` and `cLevel2(80)`. A single calibration belt (that is, a plot with a single confidence level) can be generated by specifying only the first argument. For example, setting `cLevel1(0.99)` produces a single calibration belt with confidence level 99%. A double calibration belt with customized pairs of confidence levels can be produced by providing both optional arguments, `cLevel1(#)` and `cLevel2(#)`.

`cLevel2(#)` is described above; see `cLevel1(#)`.

`nPoints(#)` specifies the number of points defining the edges of the belt. The default is `nPoints(100)`. Reducing the number of points can substantially speed up the production of the plot in large datasets. However, this number also affects the estimate of the probabilities where the belt crosses the bisector (that is, the limits of the intervals reported in the table on the plot). Indeed, the greater the value of `nPoints()`, the higher the precision in the estimate of these values. If the production of the belt is too slow, but the analysis requires an iterative construction of many belts, for example, in exploratory analyses, a possible strategy is to decrease the number of points to values much smaller than the default (say, 20 or 50), accounting for the larger uncertainty in the interpretation of the plots. Finally, when the analysis is set up, the number of points can be increased to the default value to achieve more accurate estimates of the potential deviations from the bisector.

`maxDeg(#)` fixes the maximum degree of the polynomial function used to evaluate the model. The default is `maxDeg(4)`. For further information, see section 3.

`thres(#)` sets the threshold parameter involved in the construction of the polynomial function used to evaluate the model. This parameter corresponds to one minus the significance level used when testing the increase of the polynomial order in the forward selection (see section 3 for further details). The default is `thres(0.95)`, specifying a forward selection ruled by a sequence of classic 0.05-level tests. Greater values of `thres()` correspond to more conservative scenarios, where low-order polynomials are preferred to high-order ones. Calibration belts based on lower `thres()` values are more likely to be based on high-order polynomials.

4.3 An example: ICU data from the GiViTI network

To provide an example of applying the `calibrationbelt` command, we use a dataset of 1,000 ICU patients admitted to Italian ICUs. This dataset is a subsample of the cohort of patients enrolled in the Margherita-PROSAFE project, an international observational study established to monitor the quality of care in ICU. The ongoing project is based on the continuous data collection of clinical information in about 250 units. This collaborative effort was promoted by the GiViTI network (*Gruppo Italiano per la valutazione degli interventi in Terapia Intensiva*, Italian Group for the Evaluation of the Interventions in Intensive Care Units).

The variables of the dataset include hospital mortality and the 15 covariates that compose the simplified acute physiology score, a widely used prognostic model for hospital mortality in ICU patients (Le Gall, Lemeshow, and Saulnier 1993). The 15 variables include patient demographic information, comorbidities, and clinical information. The actual values of the variables have been modified to protect subject confidentiality.

In the example described in the following sections, we use the available sample to fit a logistic regression model using these predictors. The dataset is randomly split in 750 records for model development and 250 patients for external validation. First, we fit the model on the developmental sample, and we evaluate the internal calibration with the calibration belt approach (section 4.4). Then, that model is applied to the validation sample, and the external calibration is assessed (section 4.5).

4.4 calibrationbelt for internal validation

The dataset was randomly split into developmental and validation subsets of 750 and 250 patients, respectively. The logistic regression model with the available predictors is fit to the developmental sample.

```
. use icudata
(ICU patients from the international GiViTI network)
. set seed 101
. generate random = runiform()
. sort random, stable
. generate extValidation = (_n>750)
. quietly logit outcome ib3.adm ib1.chronic ib1.age ib5.gcs
>                 ib3.BP ib3.HR ib1.temp ib3.urine ib1.urea
>                 ib2.WBC ib2.potassium ib2.sodium ib3.HCO3
>                 ib1.bili ib1.paFiIfVent if extValidation == 0
```

Considering the demonstrative purposes of this example, we evaluate only the calibration of the fit model, without considering the other fundamental model-building steps. The calibration belt and the related test for the internal assessment can be produced using the second method described in section 4.1 by simply typing `calibrationbelt` after fitting the model.

1010 *Assessing calibration of dichotomous outcome models: calibration belt*

```
. calibrationbelt
```

GiViTI Calibration Belt

Calibration belt and test for internal validation:
the calibration is evaluated on the training sample.

Sample size: 750
Polynomial degree: 2
Test statistic: 1.54
p-value: 0.2142

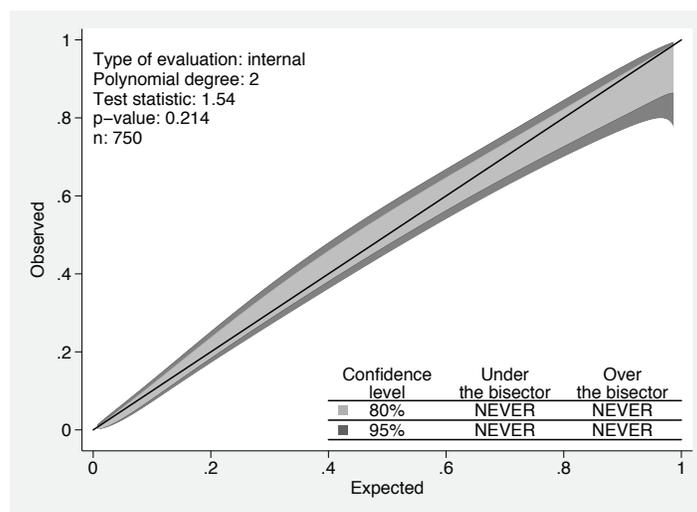


Figure 1. Calibration belt plot on the developmental sample

The output of the program reports the value of the statistic (1.54) and the p -value (0.21) of the test. These results suggest that the hypothesis of good calibration is not rejected (at the classically adopted 0.05 level). Similar conclusions can be drawn from the interpretation of the produced plot, reported in figure 1. We note that both the 80% and 95% calibration belts encompass the bisector over the whole range of the predicted probabilities. This suggests that the predictions of the model do not significantly deviate from the observed rate in the developmental sample (that is, that the model's internal calibration is acceptable).

4.5 calibrationbelt for external validation

The calibration of the fitted model is then evaluated on the records set aside for the external validation of the model. We generate the predicted probabilities resulting from the model fit in section 4.4, and we run the `calibrationbelt` command on the external sample using the `if` qualifier. Here we use the first method described in section 4.1 to invoke the command, that is, passing the variables containing the binary response (`outcome`) and the predicted probabilities (`phat`) to the program. The option `devel("external")` specifies the setting of external calibration assessment.

```

. predict phat, pr
. calibrationbelt outcome phat if extValidation==1, devel("external")
-----
                GiViTI Calibration Belt
Calibration belt and test for external validation:
the calibration is evaluated on an external, independent sample.
Selection: extValidation==1
Sample size: 250
Polynomial degree: 1
Test statistic: 26.30
p-value: 0.0000
-----

```

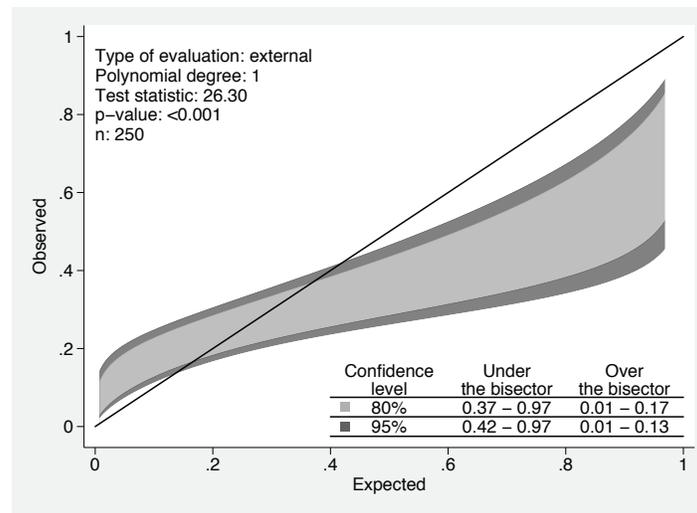


Figure 2. Calibration belt plot on the external sample

The output suggests that the fitted model is not well calibrated in the validation sample. The p -value is extremely small (less than 0.0001), which rejects the hypothesis of satisfactory fit even with very conservative significance levels. The produced calibration belts (reported in figure 2) provide interesting information. Because they lie above and do not include the bisector for small predictions, the predictions of the model significantly underestimate the actual risk in the low range of probabilities. With 95% and 80% confidence, the mortality rates are underestimated for estimated probabilities smaller than 0.13 and 0.17, respectively (see the table in the bottom-right corner of the plot). However, the model also overestimates the mortality rates for high predicted probabilities. Indeed, the calibration belts are below the bisector for probabilities higher than 0.42 and 0.37 with 95% and 80% confidence, respectively.

These results suggest that this model performed poorly in a sample different from the one on which the model was developed. A careful application of the recommended model-building steps would be necessary to achieve a better-fitting model.

5 Discussion

We presented the `calibrationbelt` command, which implements the calibration belt approach to provide important information about the calibration of binary outcome models. The calibration belt plot spots any deviation from the perfect fit, pointing out the direction of possible miscalibrations. Conveying information about the statistical significance of the deviations, the method outperforms the existing approaches to graphically evaluate binary outcome models (Nattino, Finazzi, and Bertolini 2014a).

The graphical method is paired to a statistical test, resulting in a p -value reflecting the model under consideration. Tests and belts often return concordant outputs: nonsignificant tests are often associated with the belt encompassing the 45-degree lines and significant tests with the belt deviating from the bisector. However, there are cases where the output of belt and test can disagree. Whenever the $(1 - \alpha)$ 100%-calibration belt deviates from the bisector, the statistical test is always significant at the α -level (that is, the p -value is smaller than α). On the other hand, a significant test at the α -level might correspond to a belt that does not deviate significantly from the 45-degree line at any point. The reason for the potential disagreement is the difference between the two approaches in terms of power. The calibration belt is a two-dimensional projection of the multidimensional polynomial relationship used to test the fit of the model. Being assessed in the multidimensional parameter space, the statistical test takes advantage of the full information available. On the other hand, the calibration belt, as a graphical projection, is associated with a loss of information and lower power. A simulation study investigating the agreement between test and belt is described by Nattino, Finazzi, and Bertolini (2014b). The results confirmed this theoretical explanation, showing that the calibration belt behaves slightly more conservatively than the test. In particular, the chances of having a discordant output between belt and test increase with the increase of the polynomial order.

Despite the usefulness of the statistical method, the procedure has been implemented only in an R package so far (Nattino et al. 2016b). The `calibrationbelt` command provides a user-friendly way to generate the calibration belt in Stata. The informativeness of the plot generated with the easy-to-use command can be an invaluable tool in developing better models.

6 References

- Cox, D. R., and D. V. Hinkley. 1974. *Theoretical Statistics*. London: Chapman & Hall.
- Finazzi, S., D. Poole, D. Luciani, P. E. Cogo, and G. Bertolini. 2011. Calibration belt for quality-of-care assessment based on dichotomous outcomes. *PLOS ONE* 6: e16110.
- Hosmer, D. W., Jr., and S. Lemeshow. 1980. Goodness of fit tests for the multiple logistic regression model. *Communications in Statistics—Theory and Methods* 9: 1043–1069.
- Hosmer, D. W., Jr., S. Lemeshow, and R. X. Sturdivant. 2013. *Applied Logistic Regression*. 3rd ed. Hoboken, NJ: Wiley.

- Le Gall, J.-R., S. Lemeshow, and F. Saulnier. 1993. A new simplified acute physiology score (SAPS II) based on a European/North American multicenter study. *JAMA* 270: 2957–2963.
- Nattino, G., S. Finazzi, and G. Bertolini. 2014a. Comments on ‘Graphical assessment of internal and external calibration of logistic regression models by using loess smoothers’ by Peter C. Austin and Ewout W. Steyerberg. *Statistics in Medicine* 33: 2696–2698.
- . 2014b. A new calibration test and a reappraisal of the calibration belt for the assessment of prediction models based on dichotomous outcomes. *Statistics in Medicine* 33: 2390–2407.
- . 2016a. A new test and graphical tool to assess the goodness of fit of logistic regression models. *Statistics in Medicine* 35: 709–720.
- Nattino, G., S. Finazzi, G. Bertolini, C. Rossi, and G. Carrara. 2016b. *givitiR: The GiViTI Calibration Test and Belt*. R package version 1.3. <http://CRAN.R-project.org/package=givitiR>.

About the authors

Giovanni Nattino holds a master’s of science degree in applied mathematics and is currently pursuing a PhD in biostatistics from the Ohio State University. He received a postgraduate certificate in Biomedical Research from the Mario Negri Institute for Pharmacological Research, where he has worked as a statistician for four years. He has experience in statistical applications in the areas of critical care, infant mortality, and emergency department admission. His research interests include modeling of clinical data and causal inference in observational studies.

Stanley Lemeshow, PhD, has been with the Ohio State University since 1999 as a biostatistics professor in the School of Public Health and the Department of Statistics, director of the biostatistics core of the Comprehensive Cancer Center, and director of the University’s Center for Biostatistics. He served as founding Dean of the Ohio State University College of Public Health from 2003–2013. His biostatistics research includes statistical modeling of medical data, sampling, health disparities, and cancer prevention.

Gary Phillips is a consulting biostatistician who retired from the Center for Biostatistics in the Department of Biomedical Informatics at The Ohio State University College of Medicine. He has 14 years of consulting experience and has worked on both large and small databases in the areas of critical care medicine, oncology, surgery, pharmacy, veterinary medicine, and social work. He specializes in statistical techniques involving logistic regression, linear regression, time to event analysis, and longitudinal analysis.

Stefano Finazzi is a theoretical physicist with a PhD degree in astrophysics. He has worked in the areas of quantum field theory in curved spacetime and quantum optics. He is currently a researcher at the Laboratory of Clinical Epidemiology, Mario Negri Institute for Pharmacological Research (Bergamo, Italy), and he is pursuing a PhD in life sciences from the Open University, UK. His current research interests include mathematical modeling of physiological systems and statistical applications in the area of critical care.

Guido Bertolini, MD, is the head of the Laboratory of Clinical Epidemiology at the Mario Negri Institute for Pharmacological Research (Bergamo, Italy) and of the GiViTI (Italian Group for the Evaluation of Interventions in Intensive Care Medicine) Coordinating Center. He has

1014 *Assessing calibration of dichotomous outcome models: calibration belt*

served as expert for different institutions, such as the EMA (European Medicines Agency), the National Bioethics Committee, and the AIFA (Italian Medicines Agency). He has been a contract professor in “Research methods and statistical analysis” at the postgraduate school in Anesthesia and Intensive Care at the Universities of Brescia and Milan and an adjunct professor in “Methodology for research and training in health services” at the University of Bergamo. He reviews grant applications for several organizations, including the European Commission. He has served as principal investigator of five research projects funded by the Italian Ministry of Health and the European Union. His major fields of expertise are quality-of-care assessment, comparative effectiveness research, infection control, and traumatic brain injury. He has spoken at more than 300 national and international conferences and has authored over 90 peer-reviewed articles.

Estimating receiver operative characteristic curves for time-dependent outcomes: The *stroccurve* package

Mattia Cattaneo
Department of Management
Information and Production Engineering
University of Bergamo
Bergamo, Italy
mattia.cattaneo@unibg.it

Paolo Malighetti
Department of Management
Information and Production Engineering
University of Bergamo
Bergamo, Italy
paolo.malighetti@unibg.it

Daniele Spinelli
Department of Management
Information and Production Engineering
University of Bergamo
Bergamo, Italy
daniele.spinelli@unibg.it

Abstract. Receiver operating characteristic (ROC) curves are an established method for assessing the predictive capacity of a continuous biomarker for a binary outcome. However, in some cases, outcomes are time dependent. Although the literature has proposed packages for performing ROC analysis of time-independent outcomes, a package is not yet available for analyzing the predictive capacity of continuous biomarkers when the binary outcome is time dependent. In this article, we present *stroccurve*, a new command for performing ROC analysis within a survival framework.

Keywords: st0509, *stroccurve*, receiver operating characteristic curves, time-dependent outcome, survival ROC

1 Introduction

Receiver operative characteristic (ROC) curves are a well-known method for assessing the predictive accuracy of a continuous biomarker. They provide estimates of sensitivity and one minus specificity of every possible cutoff in the biomarker distribution for a binary outcome. For an introduction to ROC curves, see [Pepe, Longton, and Janes \(2009\)](#).

While packages estimating the ROC curve for time-independent outcomes have already been developed for Stata users—that is, the *roccurve* package by [Pepe, Longton, and Janes \(2009\)](#)—a command that allows users to obtain sensitivity and specificity measure within a failure-time (survival) data framework is not yet available.

To this end, we present a new command for Stata users, *stroccurve*, that can assess the classification accuracy of a continuous biomarker when failures occur at different

points in time and when observations are subject to censoring. `stroccurve` is mainly based on the contribution of [Heagerty, Lumley, and Pepe \(2000\)](#).

This article is organized as follows: Section 2 illustrates the methods for assessing the predictive accuracy of a continuous biomarker within a survival framework, section 3 describes the syntax of the new command, and section 4 provides examples to outline the use of `stroccurve`.

2 Estimation of time-dependent ROC curves

[Heagerty, Lumley, and Pepe \(2000\)](#) defined two approaches to estimate ROC curves for failure-time data. The first applies Bayes' theorem and the Kaplan–Meier (KM) survival estimator ([Kaplan and Meier 1958](#)), while the second smooths the conditional survival function through the method provided by [Akritas \(1994\)](#).

In this section, the notation is the following:

- n : number of individuals
- T_i : failure time for individual i
- X_i : marker value for individual i
- C_i : censoring time for individual i
- $Z_i = \min(T_i, C_i)$: follow-up time for individual i
- d_i : censoring indicator for individual i . $d_i = 1$ if $T_i \leq C_i$ and $d_i = 0$ if $T_i > C_i$
- $D_i(t)$: failure status prior to time t . $D_i(t) = 1$ if $T_i \leq t$ and $D_i(t) = 0$ if $T_i > t$
- $T_n(t)$: unique levels of Z_i for observed events $D_i = 1$ in $Z_i \leq t$
- $S(t) = P(T > t)$: the survival function

2.1 KM method

The estimates of sensitivity (true positive rate, TP) and specificity (true negative rate, TN) for each possible value c of the biomarker X can be derived through Bayes' theorem,

$$\text{TP}(c, t) = P\{X > c | D(t) = 1\} = \frac{P(X > c)\{1 - S(t|X > c)\}}{1 - S(t)}$$

$$\text{TN}(c, t) = P\{X \leq c | D(t) = 0\} = \frac{P(X \leq c)S(t|X \leq c)}{S(t)}$$

where $P(X \leq c)$ is estimated by the biomarker empirical cumulative distribution $F_X(X_i) = 1/n \sum_i \mathbf{1}(X_i \leq c)$, while the estimates of the survival function $S(t)$ and of

the conditional survival function $S(t|X > c)$ are produced relying on the KM estimator $S_{KM(t)}$. The KM estimator is defined as

$$\widehat{S}_{KM}(t) = \sum_{s \in T_n(t)} \left\{ 1 - \frac{\mathbf{1}(Z_j = s)\delta_j}{\mathbf{1}(Z_j \geq s)} \right\}$$

Therefore, the estimates of sensitivity and specificity are calculated as follows:

$$\widehat{TP}_{KM}(c, t) = \frac{\{1 - F_X(c)\} \{1 - \widehat{S}_{KM}(t|X > c)\}}{1 - \widehat{S}_{KM}(t)}$$

$$\widehat{TN}_{KM}(c, t) = \frac{F_X(c)\widehat{S}_{KM}(t|X \leq c)}{\widehat{S}_{KM}(t)}$$

However, this approach produces sensitivity and specificity functions that may not be monotone with respect to the marker values, because the estimator $P(X > c, T > t) = \{1 - F_X(c)\}\{1 - \widehat{S}_{KM}(t|X > c)\}$ does not provide a valid bivariate distribution. Moreover, the KM-based estimator has a potential problem arising from the assumption that the censoring process is independent from X , which might be violated in practice.

2.2 Nearest-neighbor estimation

To overcome the violation of the monotonicity of the ROC curve with respect to the biomarker and accommodate the possibility that the censoring process is not independent from the marker values, [Heagerty, Lumley, and Pepe \(2000\)](#) provided an alternative method for estimating valid sensitivity and specificity functions using the estimator provided by [Akritas \(1994\)](#),

$$\widehat{S}_{\lambda_n}(c, t) = \frac{1}{n} \sum_i \widehat{S}_{\lambda_n}(t|X = X_i)\mathbf{1}(X_i > c)$$

where \widehat{S}_{λ_n} is a smoothed estimator of the conditional survival function depending on parameter λ_n :

$$\widehat{S}_{\lambda_n}(t|X = X_i) = \sum_{s \in T_n(t)} \left\{ 1 - \frac{\sum_j K_{\lambda_n}(X_i, X_j)\mathbf{1}(Z_j = s)d_j}{\sum_j K_{\lambda_n}(X_i, X_j)\mathbf{1}(Z_j \geq s)} \right\}$$

This parameter defines the binary nearest-neighbor kernel $K_{\lambda_n}(X_i, X_j)$, representing the percentage observations included in each neighborhood,

$$K_{\lambda_n}(X_i, X_j) = \mathbf{1}\{|F_X(X_i) - F_X(X_j)| < \lambda_n\}$$

$$2\lambda_n \in (0, 1)$$

where $\lambda_n = O(n^{-1/3})$ is sufficient to provide a weakly consistent estimator of the bivariate function in a practical situation (Heagerty, Lumley, and Pepe 2000). Hence, estimates of sensitivity and specificity can be obtained using

$$\widehat{\text{TP}}_{\lambda_n}(c, t) = \frac{1 - F_X(c) - \widehat{S}_{\lambda_n}(c, t)}{1 - \widehat{S}_{\lambda_n}(-\infty, t)}$$

$$\widehat{\text{TN}}_{\lambda_n}(c, t) = \frac{\widehat{S}_{\lambda_n}(c, t)}{\widehat{S}_{\lambda_n}(-\infty, t)}$$

where $\widehat{S}_{\lambda_n}(-\infty, t) = 1/n \sum_i \{1(X_i > c)(1 - S_{\lambda_n}(t|X = X_i))\}$.

2.3 Estimation of the optimal cutpoint

Along with ROC curves, it is often necessary to establish a single cutpoint for stratifying individuals into risk categories. For this purpose, the literature provides three decision criteria for defining an optimal cutpoint when the outcome is binary. Such criteria are based on the accuracy measures provided by ROC curves. There are three main strategies for cutpoint estimation based on selecting a biomarker value c^* from its distribution, such that

- i) the Youden function (Youden 1950), namely, the sum of sensitivity and specificity minus one, is maximized, which is equivalent to optimizing the biomarker's classification accuracy if sensitivity and specificity have the same weight in the decision maker's perspective;
- ii) the distance between the selected point and the point representing perfect classification (FP = 0, TP = 1) is minimized (Perkins and Schisterman 2006); and
- iii) the concordance probability function (defined as the product of sensitivity and specificity [Liu and Jin 2015]) is maximized.

3 The stroccurve command

3.1 Syntax

The user is required to use `stset` before using `stroccurve`. The syntax of the command is

```
stroccurve markervar [if] [in], timepoint(#) [nne lambda(#) km
  genrocvars replace nograph liu youden nearest]
```

where *markervar* is the continuous biomarker variable for which the time-dependent ROC curve is to be calculated.

3.2 Options

`timepoint(#)` specifies the time point for which the ROC curve is to be calculated. `timepoint()` is required.

`nne` calculates the time-dependent ROC curve with 0/1 nearest-neighbor kernel smoothing of the conditional survival function, the default.

`lambda(#)` specifies the percentage of observations to be included in each neighborhood if the nearest-neighbor estimator of the survival function method is used. It has to be included in the (0, 0.5) interval. The default is `lambda(0.25*n1/3)`, where n is the number of observations.

`km` calculates the time-dependent ROC curve with the KM method.

`genrocvars` generates new specificity and sensitivity variables for `markervar`, `_FP_R`, and `_TP_R` corresponding to their marker values.

`replace` requests `genrocvars` to overwrite the existing `_FP_R` and `_TP_R` variables.

`nograph` suppresses the plot.

`liu` estimates the cutoff by maximizing the concordance probability.

`youden` estimates the cutoff by maximizing the Youden function.

`nearest` estimates the closest cutpoint to (0, 1).

3.3 Stored results

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

Scalars

<code>e(AUC)</code>	returns the area under the ROC curve
<code>e(youden)</code>	returns the cutpoint maximizing the Youden criterion if the <code>youden</code> option is specified
<code>e(liu)</code>	returns the cutpoint maximizing the concordance probability if the <code>liu</code> option is specified
<code>e(nearest)</code>	returns the nearest cutpoint to (0, 1) if the <code>nearest</code> option is specified

Matrices

<code>e(rocmat)</code>	returns an $m \times 3$ matrix, where m is the number of unique marker values; the first column includes marker values; the second and third columns report the estimates of sensitivity and one minus specificity for such marker values
------------------------	---

Functions

<code>e(sample)</code>	marks the estimation sample
------------------------	-----------------------------

4 Examples

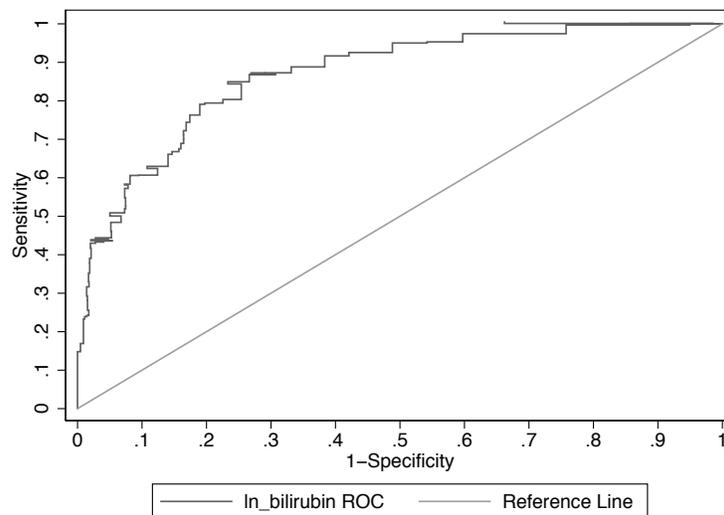
We use primary biliary cirrhosis data from [Fleming and Harrington \(1991\)](#).

```
. use pbc.dta
(PBC data, 3 sources)
. quietly stset survtime, failure(censdead==1)
```

► Example 1

We calculate the survival ROC curve for survival at 2,000 days, with the Kaplan–Meier estimator of the survival function.

```
. stroccurve ln_bilirubin, timepoint(2000) km
Time dependent ROC curve at time:      2000
Survival Function Estimation Method:   Kaplan Meier
N=                                     312
Area under the ROC Curve:              0.868
```

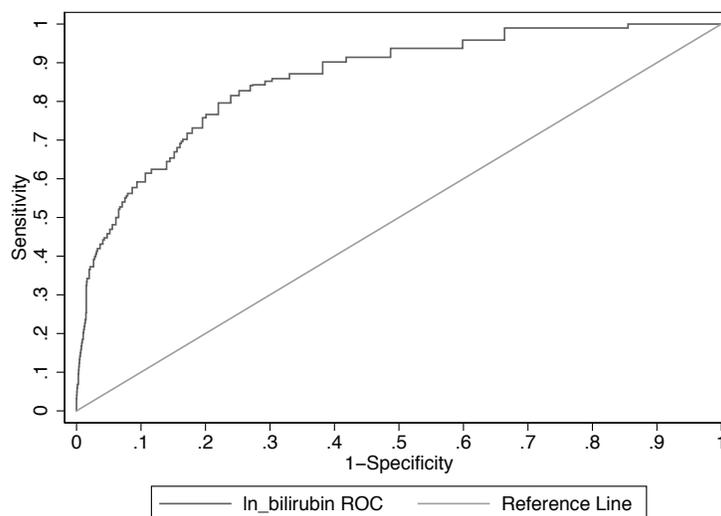


◀

► **Example 2**

We calculate the nearest-neighbor ROC curve for survival at 2,000 days, with the smoothing parameter equal to $0.25 \times N^{1/3}$.

```
. stroccurve ln_bilirubin, timepoint(2000)
Smoothing parameter automatically set at 0.25*N^(-1/3)
Time dependent ROC curve at time:      2000
Survival Function Estimation Method:   Nearest Neighbor
Smoothing parameter:                   0.037
N=                                      312
Area under the ROC Curve:               0.858
```



◀

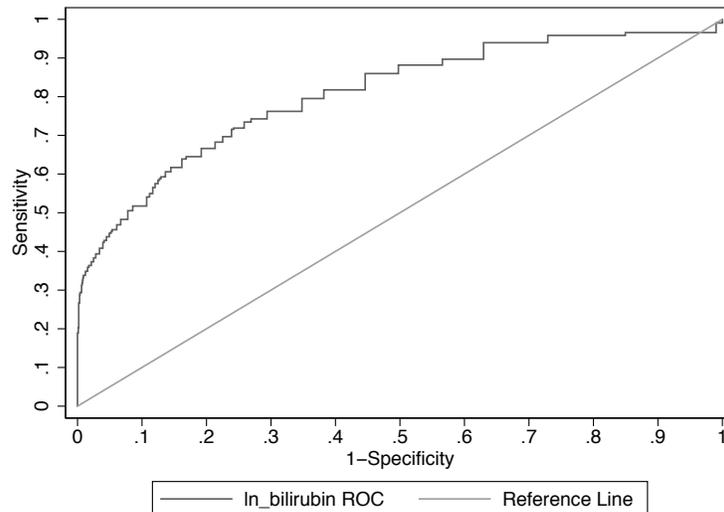
► Example 3

We calculate the nearest-neighbor ROC curve for survival at 3,000 days, with the smoothing parameter equal to $0.25 \times N^{1/3}$, and request the optimal cutpoint according to Perkins and Schisterman (2006).

```
. stroccurve ln_bilirubin, timepoint(3000) nearest
Smoothing parameter automatically set at 0.25*N^(-1/3)
Time dependent ROC curve at time:      3000
Survival Function Estimation Method:   Nearest Neighbor
Smoothing parameter:                   0.037
N=                                      312
Area under the ROC Curve:              0.816
```

OPTIMAL CUTPOINTS:

```
Nearest point to (0,1)
Optimal cutpoint:          3.245
Sensitivity at optimal cutpoint: 0.734
Specificity at optimal cutpoint: 0.742
```



◀

5 Acknowledgments

We thank Anna Falanga, MD, Cinzia Giaccherini, and Cristina Verzeroli (ASST Papa Giovanni XXIII) for their helpful comments.

6 References

- Akritis, M. G. 1994. Nearest neighbor estimation of a bivariate distribution under random censoring. *Annals of Statistics* 22: 1299–1327.
- Fleming, T. R., and D. P. Harrington. 1991. *Counting Processes and Survival Analysis*. New York: Wiley.
- Heagerty, P. J., T. Lumley, and M. S. Pepe. 2000. Time-dependent ROC curves for censored survival data and a diagnostic marker. *Biometrics* 56: 337–344.
- Kaplan, E. L., and P. Meier. 1958. Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association* 53: 457–481.
- Liu, X., and Z. Jin. 2015. Optimal survival time-related cut-point with censored data. *Statistics in Medicine* 34: 515–524.
- Pepe, M. S., G. Longton, and H. Janes. 2009. Estimation and comparison of receiver operating characteristic curves. *Stata Journal* 9: 1–16.
- Perkins, N. J., and E. F. Schisterman. 2006. The inconsistency of “optimal” cutpoints obtained using two criteria based on the receiver operating characteristic curve. *American Journal of Epidemiology* 163: 670–675.
- Youden, W. J. 1950. Index for rating diagnostic tests. *Cancer* 3: 32–35.

About the authors

Mattia Cattaneo, PhD, is a researcher at the Department of Management, Information and Production Engineering of the University of Bergamo, a scientific coordinator of the International Center for Competitiveness Studies in the Aviation Industry (ICCSAI), and a member of the Higher Education Research Group (HERG). His research interests include healthcare management, transport economics, regional development, and higher education. He is often a reviewer for several journals in the field of economics and higher education.

Paolo Malighetti, PhD, is associate professor at the Department of Management, Information and Production Engineering of the University of Bergamo, where he teaches health technology assessment, business economics, and transport management. His main research interests are related to the pricing and evolution of air transport networks and to healthcare management and economics, specifically, aspects related to the adoption of new technology within clinical practice. Since 2013, he has been director of the Human Factors and Technology in Healthcare (HTH) Center at the University of Bergamo.

Daniele Spinelli is a PhD student in economics and management of technology (DREAMT) at the Department of Management, Information and Production Engineering of the University of Bergamo and at the Department of Economics and Management of the University of Pavia. His main research interests are health economics, econometrics, and biostatistics.

Software Updates

st0245_1: Age-period-cohort models in Stata. P. D. Sasieni *Stata Journal* 12: 45–60.

In the previous version of the program, the linear term from the cohort and period effect (that is, drift) was only entered in the model when either an age-period or the default age-period-cohort model was specified. This has been fixed. The program now allows the user to specify whether the drift is to be included (that is, age versus age-drift model). To include this linear term, the user should specify the option `drift`; for example, `apcspline cases age year if sex==1, drift exposure(population)`. Having this option offers greater flexibility to the user, who can choose to model an age-only model and compare it with more complex models such as the age-drift model. Moreover, the command now allows users to run projections with no damping on the drift by specifying a damping factor of 1 in the option, `damping(1)`. Damping must be set to 1 because the drift is multiplied by the damping factor. We are not advocating that this leads to realistic estimates but that this offers the opportunity to run sensitivity analyses to ascertain a scenario where trends carry on into the future. Finally, the command has now been modified so that the user can include covariates in the model as well as interaction terms using factor variables and time-series operators.

st0389_5: Conducting interrupted time-series analysis for single- and multiple-group comparisons. A. Linden. *Stata Journal* 17: 515–516; 17: 73–88; 16: 813–814; 16: 521–522; 15: 480–500.

Fixed the figures so that values for the “actual” observations will be weighted when the user specifies a weight.

Rearranged the order of variables presented in the regression output table so that variables generated by `itsa` come first.

st0393_2: Estimating almost-ideal demand systems with endogenous regressors.

S. Lecocq and J.-M. Robin. *Stata Journal* 16: 244; 15: 554–573.

The following changes have been made in the `aidsills` command:

1. An error has been corrected in the calculation of the Stone price index.
2. Attempting to run the command with more or fewer variables in prices than in shares now results in an error message.
3. Finding missing values in parameter estimates b and their variance–covariance matrix \mathbf{V} now results in an error message suggesting that the user check the model specification.