The Stata Journal (2019)
19, Number 4, pp. 950–988 DOI: 10.1177/1536867X19893640

Simar and Wilson two-stage efficiency analysis for Stata

Oleg Badunenko
University of Portsmouth
Portsmouth, UK
oleg.badunenko@port.ac.uk

Harald Tauchmann
Friedrich-Alexander-Universität Erlangen-Nürnberg
Rheinisch-Westfälisches Institut für Wirtschaftsforschung (RWI)
and CINCH (Centre of Health Economics Research)
Nürnberg, Germany
harald.tauchmann@fau.de

Abstract. When one analyzes the determinants of production efficiency, regressing efficiency scores estimated by data envelopment analysis on explanatory variables has much intuitive appeal. Simar and Wilson (2007, *Journal of Econometrics* 136: 31–64) show that this conventional two-stage estimation procedure suffers from severe flaws that render its results, and particularly statistical inference based on them, questionable. They additionally propose a statistically grounded bootstrap-based two-stage estimator that eliminates the above-mentioned weaknesses of its conventional predecessors and comes in two variants. In this article, we introduce the new command `simarwilson`, which implements either variant of the suggested estimator in Stata. The command allows for various options and extends the original procedure in some respects. For instance, it allows for analyzing both output- and input-oriented efficiency. To demonstrate the capabilities of `simarwilson`, we use data from the Penn World Tables and the Global Competitiveness Report by the World Economic Forum to perform a cross-country empirical study about the importance of quality of governance in a country for its efficiency of output production.

Keywords: st0585, simarwilson, simarwilson postestimation, gciget, Global Competitiveness Index, DEA, two-stage estimation, truncated regression, bootstrap, efficiency, bias correction, environmental variables

1 Introduction

Analyzing the technical efficiency of production or decision-making units (DMUs) has developed into a major field in empirical economics and management science.[1] From a methodological perspective, the two most popular strands of the literature can be distinguished: i) analyses that rest on parametric, regression-based methods, namely,

[1] In a supplement to their recent survey article, Emrouznejad and Yang (2018) list more than 10,000 publications, considering only the nonparametric strand of this literature.

stochastic frontier analysis (Aigner, Lovell, and Schmidt 1977), and ii) analyses that use nonparametric methods, namely, data envelopment analysis (DEA) (Charnes, Cooper, and Rhodes 1978). The pros and cons of either approach have been discussed extensively (for example, Hjalmarsson, Kumbhakar, and Heshmati [1996]; Murillo-Zamorano [2004]).

One of the advantages of the parametric approaches, namely, the truncated normal stochastic frontier model, is that it not only allows for measuring inefficiency but also incorporates a model of the determinants of inefficiency.[2] In contrast, nonparametric approaches are primarily concerned with estimating a production-possibility frontier (or an input-requirement frontier) and with measuring the distance of observed input-output combinations to this frontier. However, shedding light on what determines the magnitude of this distance is out of the narrow[3] scope of nonparametric approaches such as DEA.

For many research questions, however, identifying determinants of inefficiency is more relevant than determining its magnitude for specific DMUs. For this reason, in the domain of nonparametric efficiency analysis, semiparametric two-stage approaches that combine efficiency measurement by DEA with a regression analysis that uses DEA estimated efficiency as dependent variables have become popular. Simar and Wilson (2007) list almost 50 published articles and mention hundreds of unpublished articles that use such two-stage procedures. In these (early) applications, the second stage is typically a censored (tobitlike) regression to account for the bounded nature of DEA efficiency scores or just simply ordinary least squares (Simar and Wilson 2007).

Despite their popularity and their intuitive appeal, such conventional two-stage estimators are criticized by Simar and Wilson (2007) mainly for two reasons. First, they stress the absence of a clear theory of the underlying data-generating process that would justify the conventional two-stage approach.[4] Second, they criticize the conventional inference that is pursued in most two-stage applications for ignoring that estimated DEA efficiency scores are calculated from a common sample of data. Treating them as if

[2] The Stata command `frontier` includes the option `cm()`, which allows for specifying the conditional mean of the truncated normal distribution, from which the distance to the frontier is assumed to be drawn as a linear function of exogenous variables. One may alternatively use the option `uhet()` to specify the variance of the inefficiency as a function of exogenous variables.

[3] Some DEA models were proposed that directly include environmental variables in the DEA linear programming problem (compare Coelli et al. [2005]). However, these models suffer from several shortcomings. For instance, they allow only for continuous environmental variables. More recently, smoothing-based, fully nonparametric methods for estimating conditional frontiers, which substantially extend the familiar DEA framework, have been suggested (Daraio and Simar 2005, 2007). Unlike two-stage approaches, these models allow for environmental variables affecting the shape of the frontier. However, this is different than specifying the distance from the frontier as a function of exogenous variables.

[4] In a closely related article, Simar and Wilson (2011) discuss further contributions (Hoff 2007; McDonald 2009; Ramalho, Ramalho, and Henriques 2010; Banker and Natarajan 2008) to the literature on two-stage estimators. Unlike many applications of the conventional approach, these articles provide rationales or statistical justification for the proposed two-stage procedures. Simar and Wilson (2011, 206), however, deny the first three a decent basis in statistical theory. With respect to the last, they argue that the claimed desirable statistical properties rely on restrictive assumptions.

they were independent observations is not appropriate, because the problems related to invalid inference due to serial correlation arise. Simar and Wilson (2007) develop a two-stage procedure that accounts for the above-mentioned issues. They describe an underlying data-generating process that is consistent with a two-stage estimation procedure, which—as the most obvious difference to the earlier conventional approach—implies a truncated rather than censored regression model. This reflects that the substantial share of fully efficient DMUs typically found in DEA is an artifact of the finite-sample bias inherent in DEA but does not represent a feature of the true underlying data-generating process. Moreover, they propose two parametric bootstrap procedures that are consistent with the assumed data-generating process and address the second issue. They yield estimated standard errors and confidence intervals that do not suffer from bias due to estimated efficiency scores being correlated.

The Simar and Wilson (2007) procedure has become a workhorse of empirical efficiency analysis with hundreds of applications from various fields of economics.[5] This popular, but technically involved estimator, has not yet been available to Stata users unless they developed their own code. In this article, we introduce the new command `simarwilson`, which allows for applying this estimator in Stata.[6] In doing this, it greatly benefits from the recently published community-contributed command `teradial` (Badunenko and Mozharovskyi 2016), which is required for running `simarwilson`. For the first time, `teradial` enables fast estimation of DEA in Stata even for large samples. This is essential for practical applications of the Simar and Wilson (2007) estimator because it involves bootstrapping the DEA estimator.[7]

The remainder of this article is organized as follows. Section 2 gives a brief summary of the Simar and Wilson (2007) two-stage estimator. Section 3 describes the syntax of `simarwilson`. Section 4 presents an application to cross-country data. Section 5 concludes.

2 The Simar and Wilson (2007) estimator in brief

2.1 Some essential ideas

Simar and Wilson (2007) consider a setting in which a researcher observes three types of variables $\mathbf{x}_i$, $\mathbf{y}_i$, and $\mathbf{z}_i$ for a sample of $i = 1, \ldots, N$ DMUs. $\mathbf{x}_i$ denotes a vector of P inputs to production. $\mathbf{y}_i$ is a vector of Q outputs from production. $\mathbf{z}_i$ denotes a row vector of K environmental variables that may affect the ability of DMU i to efficiently combine the consumed inputs to the produced outputs. The effect of $\mathbf{z}_i$ on efficiency

[5] Google Scholar (accessed September 16, 2019) lists more than 2,600 citations (for instance, Fragkiadakis et al. [2016]; Pérez Urdiales, Lansink, and Wall [2016]; Glass, Kenjegalieva, and Taylor [2015]; Chortareas, Girardone, and Ventouri [2013]). Interestingly, the inventors of this popular method dissociate themselves from advocating two-stage approaches (Simar and Wilson 2011, 216).

[6] An earlier, less powerful version of `simarwilson` was made available through the Statistical Software Components archive in 2016.

[7] This applies to algorithm 2 but not to algorithm 1; see section 2. Prior to `teradial` being available, algorithm 2 was hence effectively out of reach for Stata users.

is in the focus of the empirical analysis. The production technology is assumed to be homogeneous across DMUs. That is, a common production-possibility frontier—the boundary of the convex production-possibility set—represents all combinations $(\mathbf{y}_j^*, \mathbf{x}_j^*)$ that are fully efficient in that no output can be increased without decreasing at least one other output or increasing at least one input (Koopmans 1951). A crucial assumption is that the shape of the production-possibility frontier does not depend on $\mathbf{z}_i$, which is referred to as separability in Simar and Wilson (2007).

The output-input set $(\mathbf{y}_i, \mathbf{x}_i)$ observed for DMU i will regularly fail in realizing a point at the frontier. This deviation is necessarily directional; that is, i produces less output than technically feasible, or it consumes more input than technically feasible. The widely used output-oriented Farrell (1957) distance measure quantifies the deviation from the frontier as the relative radial distance in output direction θ_i. That is, θ_i denotes the factor by which output generation $\mathbf{y}_i$ of DMU i has to be proportionally increased to project $(\mathbf{y}_i, \mathbf{x}_i)$ onto the frontier. θ_i is hence a measure of inefficiency that is bounded to the $[1, \infty)$ interval. Alternatively, one may measure the Farrell distance in input direction as ϑ_i, that is, the factor by which input consumption $\mathbf{x}_i$ of DMU i has to be proportionally reduced to project $(\mathbf{y}_i, \mathbf{x}_i)$ onto the frontier. Hence, ϑ_i is a measure of efficiency that is bounded to the $(0, 1]$ interval.[8] Yet, in Simar and Wilson (2007), the focus is on θ_i.[9]

The key idea in Simar and Wilson (2007) about the data-generating process is that efficiency θ_i linearly depends on $\mathbf{z}_i$,

$$\theta_i = \mathbf{z}_i\boldsymbol{\beta} + \varepsilon_i \tag{1}$$

where $\boldsymbol{\beta}$ denotes a column vector of coefficients, the estimation of which is the ultimate objective of the empirical analysis. The disturbances ε_i are assumed to be statistically independent across DMUs[10] and to follow a truncated normal distribution with parameters $\mu = 0$ and σ and left-truncation at $1 - \mathbf{z}_i\boldsymbol{\beta}$.[11] This assumption guarantees that θ_i cannot be smaller than unity irrespective of the values the variables in $\mathbf{z}_i$ may take. Though full efficiency ($\theta_i = 1$) is in principle possible, it occurs with zero probability. Conditional on θ_i, DMU i chooses a set of outputs and inputs $(\mathbf{y}_i, \mathbf{x}_i)$ as $(\mathbf{y}_i^*/\theta_i, \mathbf{x}_i^*)$, with $(\mathbf{y}_i^*, \mathbf{x}_i^*)$ denoting some point on the production-possibility frontier.[12] That is, rather than the technically feasible amount of output $\mathbf{y}_i^*$, only $(100/\theta_i)$ percent of $\mathbf{y}_i^*$ is actually produced.

[8] Shephard (1970) introduced an alternative to the Farrell (1957) distance measure that is just the reciprocal of θ_i and ϑ_i, respectively. `simarwilson` accommodates the Shephard measure by specifying the option `invert`; see sections 2.3 and 3.

[9] Simar and Wilson (2007) do not explicitly consider input-oriented efficiency, that is, ϑ_i. `simarwilson` straightforwardly extends the original Simar and Wilson (2007) to accommodating input-oriented efficiency too; see sections 2.3 and 3.

[10] This implies that `simarwilson` is meant for being used with standard cross-sectional data but not with panel data or other types of clustered data.

[11] The first and second moments of the conditional error distribution are $E(\varepsilon_i|\mathbf{z}_i) = \sigma\lambda_i$ and $\mathrm{Var}(\varepsilon_i|\mathbf{z}_i) = \sigma^2[1 + \{(1 - \mathbf{z}_i\boldsymbol{\beta})/\sigma\}(\lambda_i - \lambda_i^2)]$, with λ_i denoting the inverse Mills ratio $(\phi[\{(1 - \mathbf{z}_i\boldsymbol{\beta})/\sigma\}])/[1 - \Phi\{(1 - \mathbf{z}_i\boldsymbol{\beta})/\sigma\}]$.

[12] The data-generating process can hence be described as sampling from the joint distribution $f(\mathbf{x}, \mathbf{y}, \mathbf{z})$, which can be written as $f(\mathbf{x}, \mathbf{y}|\theta, \mathbf{z}) \times f(\theta|\mathbf{z}) \times f(\mathbf{z})$ (Simar and Wilson 2007, 35). The focus of the empirical analysis is on $f(\theta|\mathbf{z})$.

It is key for understanding the shortcomings of conventional two-stage approaches that θ_i is genuinely unobservable. Consequently, the estimated efficiency score $\widehat{\theta}_i$, which is obtained by running a DEA,[13] is not θ_i. In other words, $\widehat{\theta}_i$ is not the distance of $(\mathbf{y}_i, \mathbf{x}_i)$ to the true production-possibility frontier, but the distance to an estimate of the latter. Because of the boundary estimation framework of DEA, this estimate suffers from finite-sample bias, and in turn $\widehat{\theta}_i$ is biased toward the value of 1. That means that (1) cannot be estimated straightforwardly, and θ_i has to be replaced in (1) by the biased estimate $\widehat{\theta}_i$ to formulate an operational regression equation. As pointed out in Simar and Wilson (2007), this generates two major problems for conventional two-step approaches. First, although the errors ε_i are assumed to be statistically independent across DMUs, the operational errors in a regression of $\widehat{\theta}_i$ on $\mathbf{z}_i$ are not, because the $\widehat{\theta}_i$ are estimated from a common sample of data. Second, in any application of DEA some—usually numerous—$\widehat{\theta}_i$ take the value of 1, though according to (1), θ_i takes this value with zero probability.

In the procedure[14] suggested in Simar and Wilson (2007), the former issue is addressed by estimating standard errors and confidence intervals for $\widehat{\boldsymbol{\beta}}$ with a parametric bootstrap procedure, in which artificial pseudoerrors are independently drawn from the truncated normal distribution with left-truncation at $1 - \mathbf{z}_i\widehat{\boldsymbol{\beta}}$. The latter issue is addressed in Simar and Wilson (2007) in two ways, which leads to two different suggested estimation procedures (algorithm 1 and algorithm 2). Algorithm 1 simply excludes those DMUs from the regression analysis, for which DEA yields scores $\widehat{\theta}_i$ that equal 1. These are obviously artifacts of finite-sample bias. The remaining M (with $M < N$) DEA scores enter a truncated regression model (left-truncation at 1) as the left-hand-side variable. Fitting this model yields $\widehat{\boldsymbol{\beta}}$, which, together with the estimate for the variance parameter $\widehat{\sigma}$, enters the bootstrap procedure mentioned above. The second suggested approach (algorithm 2) is more involved and rests on bias-corrected DEA scores $\widehat{\theta_i^{\text{bc}}}$ as the left-hand-side variable. Because $\widehat{\theta_i^{\text{bc}}} > 1$ holds for $i = 1, \dots, N$, unlike algorithm 1, all DMUs are considered in the truncated regression analysis and the subsequent bootstrap procedure. The bias correction itself rests on a bootstrap procedure that incorporates the assumptions regarding the data-generating process of θ_i, that is, (1). For this reason, it is computationally simpler and more parametric than alternative bias-correction procedures that have been suggested in the literature (Simar and Wilson 2000; Kneip, Simar, and Wilson 2008) and have recently been made available to Stata users by the community-contributed command `teradialbc` (Badunenko and Mozharovskyi 2016).

[13] Discussing the linear programming problem that has to be solved to obtain $\widehat{\theta}_i$ is out of the scope of this article. Readers not familiar with DEA are referred to the seminal article of Charnes, Cooper, and Rhodes (1978) and standard textbooks such as Coelli et al. (2005) and Cooper, Seiford, and Tone (2007). Badunenko and Mozharovskyi (2016) also provide a Stata-oriented, brief, and intuitive introduction.

[14] This article focuses on the intuition behind the Simar and Wilson (2007) estimator and its practical implementation in Stata, so we do not discuss its statistical properties in depth. Readers who are interested in more theory-oriented discussion of the estimator are referred to the original article (Simar and Wilson 2007).

Figure 1 graphically illustrates (and the notes below describe) the concepts of true, DEA estimated, and bias-corrected estimated inefficiency, using randomly generated data and considering a simple single input–single output production technology.

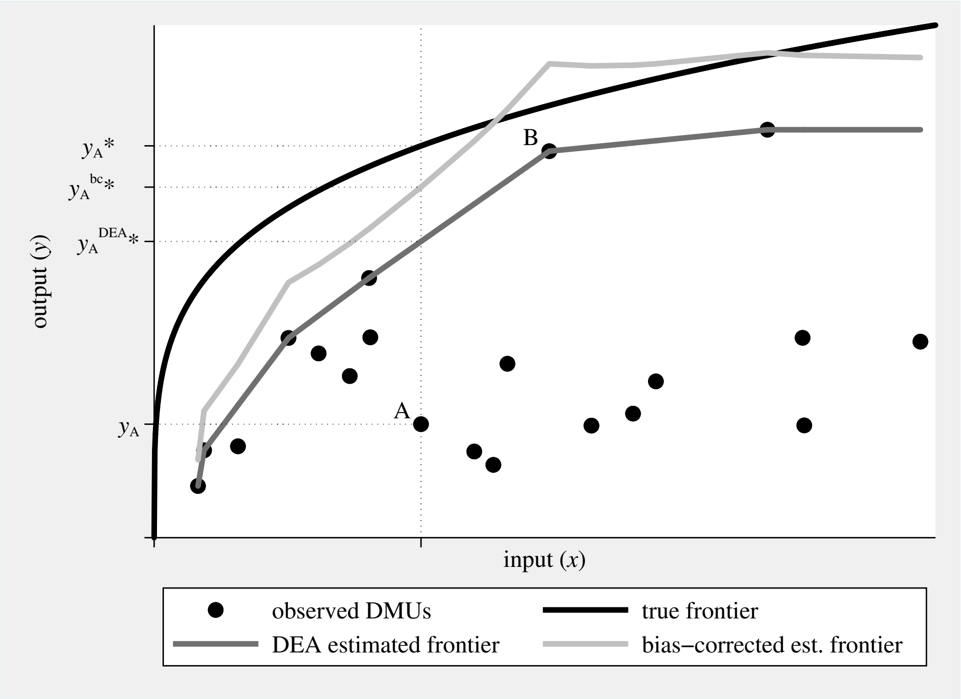

Figure 1. Graphical illustration of true and estimated inefficiency. Considering DMU A, true inefficiency θ_{A} is $y^*_{\mathrm{A}}/y_{\mathrm{A}}$, (uncorrected) DEA estimated inefficiency $\widehat{\theta_{\mathrm{A}}}$ is $y^{\mathrm{DEA}*}_{\mathrm{A}}/y_{\mathrm{A}}$, and bias-corrected estimated inefficiency $\widehat{\theta^{\mathrm{bc}}_{\mathrm{A}}}$ is $y^{\mathrm{bc}*}_{\mathrm{A}}/y_{\mathrm{A}}$. In this finite and small ($N = 20$) artificial sample, DEA systematically underestimates true inefficiency. Bias correction adjusts estimated inefficiency upward. DMU B, for instance, which is seemingly fully efficient according to conventional DEA, is inefficient according to the bias-corrected estimated frontier. Indeed, the inefficiency of B is even overestimated by $\widehat{\theta^{\mathrm{bc}}_{\mathrm{B}}}$. Unlike for conventional DEA, with bias correction the estimated production-possibility set is not convex; the estimated frontier is not even monotone. NOTE: Input quantities x randomly drawn from continuous uniform $U(0, 2)$ distribution; true frontier (production function) $y = x^{1/4}$; inefficiency generated according to (1), with $\boldsymbol{\beta} = \mathbf{0}$ and $\sigma = 3$; variable returns to scale assumed in the DEA; bias correction follows steps 1–4, algorithm 2 (Simar and Wilson [2007]; see below). SOURCE: Calculations are our own.

2.2 The procedures suggested in Simar and Wilson (2007)

This subsection describes in detail the suggested procedures algorithm 1 and algorithm 2. In doing this, it almost exactly reproduces what is found on pages 41–43 in Simar and Wilson (2007). This particularly applies to the subsequent numbered paragraphs that describe the steps of the estimation procedures almost exactly as they are described in the key reference.

Algorithm 1 consists of the following steps:

1. Compute $\widehat{\theta}_i$ for all DMUs $i = 1, \ldots, N$ using DEA.

2. Use those M (with $M < N$) DMUs for which $\widehat{\theta}_i > 1$ holds in a truncated regression (left-truncation at 1) of $\widehat{\theta}_i$ on $\mathbf{z}_i$ to obtain coefficient estimates $\widehat{\boldsymbol{\beta}}$ and an estimate for variance parameter $\widehat{\sigma}$ by maximum likelihood.

3. Loop over the following steps 3.1–3.3 B times to obtain a set of B bootstrap estimates $(\widehat{\boldsymbol{\beta}}^b, \widehat{\sigma}^b)$, with $b = 1, \ldots, B$.

    3.1. For each DMU $i = 1, \ldots, M$, draw an artificial error $\widetilde{\varepsilon}_i$ from the truncated $N(0, \widehat{\sigma})$ distribution with left-truncation at $1 - \mathbf{z}_i\widehat{\boldsymbol{\beta}}$.

    3.2. Calculate artificial efficiency scores $\widetilde{\theta}_i$ as $\mathbf{z}_i\widehat{\boldsymbol{\beta}} + \widetilde{\varepsilon}_i$ for each DMU $i = 1, \ldots, M$.

    3.3. Run a truncated regression (left-truncation at 1) of $\widetilde{\theta}_i$ on $\mathbf{z}_i$ to obtain maximum-likelihood bootstrap estimates $\widehat{\boldsymbol{\beta}}^b$ and $\widehat{\sigma}^b$.

4. Calculate confidence intervals and standard errors for $\widehat{\boldsymbol{\beta}}$ and $\widehat{\sigma}$ from the bootstrap distributions of $\widehat{\boldsymbol{\beta}}^b$ and $\widehat{\sigma}^b$.

The more involved **algorithm 2** consists of the following steps:

1. Compute $\widehat{\theta}_i$ for all DMUs $i = 1, \ldots, N$ using DEA.

2. Use those M ($M < N$) DMUs for which $\widehat{\theta}_i > 1$ holds in a truncated regression (left-truncation at 1) of $\widehat{\theta}_i$ on $\mathbf{z}_i$ to obtain coefficient estimates $\widehat{\boldsymbol{\beta}}$ and an estimate for variance parameter $\widehat{\sigma}$ by maximum likelihood.

3. Loop over the following steps 3.1–3.4 B_1 times to obtain a set of B_1 bootstrap estimates $\widehat{\theta}_i^b$ for each DMU $i = 1, \ldots, N$, with $b = 1, \ldots, B_1$.

    3.1. For each DMU $i = 1, \ldots, N$, draw an artificial error $\widetilde{\varepsilon}_i$ from the truncated $N(0, \widehat{\sigma})$ distribution with left-truncation at $1 - \mathbf{z}_i\widehat{\boldsymbol{\beta}}$.

    3.2. Calculate artificial efficiency scores $\widetilde{\theta}_i$ as $\mathbf{z}_i\widehat{\boldsymbol{\beta}} + \widetilde{\varepsilon}_i$ for each DMU $i = 1, \ldots, N$.

    3.3. Generate $i = 1, \ldots, N$ artificial DMUs with input quantities $\widetilde{\mathbf{x}}_i = \mathbf{x}_i$ and output quantities $\widetilde{\mathbf{y}}_i = (\widehat{\theta}_i / \widetilde{\theta}_i)\mathbf{y}_i$.

    3.4. Use the N artificial DMUs, generated in step 3.3, as reference set in a DEA that yields $\widehat{\theta}_i^b$ for each original DMU $i = 1, \ldots, N$.

4. For each DMU $i = 1, \ldots, N$, calculate a bias-corrected efficiency score $\widehat{\theta}_i^{\mathrm{bc}}$ as $\widehat{\theta}_i - \{(1/B_1)\sum_{b=1}^{B_1} \widehat{\theta}_i^b - \widehat{\theta}_i\}$.

5. Run a truncated regression (left-truncation at 1) of $\widehat{\theta}_i^{\mathrm{bc}}$ on $\mathbf{z}_i$ to obtain coefficient estimates $\widehat{\widehat{\boldsymbol{\beta}}}$ and an estimate for variance parameter $\widehat{\widehat{\sigma}}$ by maximum likelihood.

6. Loop over the following steps 6.1–6.3 B_2 times to obtain a set of B_2 bootstrap estimates $(\widehat{\widehat{\boldsymbol{\beta}}}^b, \widehat{\widehat{\sigma}}^b)$, with $b = 1, \ldots, B_2$.

   6.1. For each DMU $i = 1, \ldots, N$, draw an artificial error $\widetilde{\widetilde{\varepsilon}}_i$ from the truncated $N(0, \widehat{\widehat{\sigma}})$ distribution with left-truncation at $1 - \mathbf{z}_i\widehat{\widehat{\boldsymbol{\beta}}}$.

   6.2. Calculate artificial efficiency scores $\widetilde{\widetilde{\theta}}_i$ as $\mathbf{z}_i\widehat{\widehat{\boldsymbol{\beta}}} + \widetilde{\widetilde{\varepsilon}}_i$ for each DMU $i = 1, \ldots, N$.

   6.3. Run a truncated regression (left-truncation at 1) of $\widetilde{\widetilde{\theta}}_i$ on $\mathbf{z}_i$ to obtain bootstrap estimates $\widehat{\widehat{\boldsymbol{\beta}}}^b$ and $\widehat{\widehat{\sigma}}^b$ by maximum likelihood.

7. Calculate confidence intervals and standard errors for $\widehat{\widehat{\boldsymbol{\beta}}}$ and $\widehat{\widehat{\sigma}}$ from the bootstrap distribution of $\widehat{\widehat{\boldsymbol{\beta}}}^b$ and $\widehat{\widehat{\sigma}}^b$.

`simarwilson` uses the inverse-transform method for generating pseudotruncated normal random variates.[15] Choosing sufficiently large values for B_1 and B_2—the latter corresponds to B in algorithm 1—is crucial for the bias correction and estimation of percentile-based confidence intervals yielding meaningful results. For B_1 and B_2, `simarwilson` uses the default of 100 and 1,000 bootstrap repetitions, respectively. The former default value is suggested in Simar and Wilson (2007), yet depending on the data used, choosing a substantially larger number for B_1 may be advisable. If normal-approximated confidence intervals (option `cinormal`) are preferred, one may choose a much smaller number than the default for B and B_2, respectively. Running `simarwilson`, particularly algorithm 2, requires a substantial amount of computing time, which rapidly increases in the number of observations and the number of inputs and outputs in DEA. For small samples, looping over truncated regression takes the lion's share of computing time. If the sample is large, looping over DEA consumes relatively more time.[16]

[15] In rare cases for which the linear prediction $\mathbf{z}_i\widehat{\boldsymbol{\beta}}$ takes extreme values, generating pseudotruncated normal random variates may fail (Chopin 2011). Therefore, `simarwilson` stops and issues an error message if the initial truncated regression (step 2 or step 5) yields abs$\{(1 - \mathbf{z}_i\widehat{\boldsymbol{\beta}})/\widehat{\sigma}\} > 37.5$ for at least one observation. This provides strong indication for the model being ill specified or the data suffering from a severe outlier problem.

[16] In the application ($N = 131$, $M = 113$, $Q = 1$, $P = 3$, $K = 7$), presented in section 4, with $B = 2000$, runtime for algorithm 1 is 88 seconds, while it is 96 seconds for algorithm 2, with $B_1 = 1000$ and $B_2 = 2000$ (Stata/MP 15.1, Microsoft Windows 10™, Intel® Core™ i7-8550U 1.80 GHz, 24 GB RAM). Using the default values for B, B_1, and B_2 reduces runtime to 44 and 46 seconds, respectively. If the sample is expanded by the factors 2, 5, 10, 25, and 50, runtime for algorithm 1 is increased by the factors 1.1, 1.3, 1.9, 3.0, and 4.9, respectively. The corresponding factors for algorithm 2 are 1.3, 1.8, 2.8, 6.0, and 12.2. If the application is run on a weaker machine (Intel® Core™ i7-3520M 2.90 GHz, 8 GB RAM), runtime increases substantially. It is 148 seconds for algorithm 1 and 162 seconds for algorithm 2. Expanding the sample as described inflates computing times by almost the same factors as for the faster machine. (Algorithm 1: 1.1, 1.3, 1.7, 3.0, 5.2; algorithm 2: 1.2, 1.8, 2.7, 5.6, 11.4.)

2.3 Some minor extensions

The new command `simarwilson` is meant to implement the above procedures one to one in Stata. It deviates from what is suggested in Simar and Wilson (2007) only by allowing for some settings and features that are not explicitly considered there.

- `simarwilson` allows for analyzing input-oriented efficiency, while Simar and Wilson (2007) consider only the output-oriented counterpart. This requires estimating an input-oriented efficiency measure $\widehat{\vartheta}_i$ in step 1 (algorithm 1) and steps 1 and 3.4 (algorithm 2) and interchanging the treatment of inputs and outputs in step 3.3 (algorithm 2). Beyond this, only two minor changes are required: i) all truncated regressions, by default, consider two-sided truncation (at 0 from the left and at 1 from the right) rather than one-sided truncation; ii) rather than sampling from a one-sided truncated normal distribution, the artificial errors are drawn from a two-sided truncated normal distribution with left-truncation at $-\mathbf{z}_i\widehat{\boldsymbol{\beta}}$ and right-truncation at $1-\mathbf{z}_i\widehat{\boldsymbol{\beta}}$ (algorithm 2, step 6.1 $-\mathbf{z}_i\widehat{\widehat{\boldsymbol{\beta}}}$ and $1-\mathbf{z}_i\widehat{\widehat{\boldsymbol{\beta}}}$, respectively).[17] By this, one takes into account that the Farrell input-oriented efficiency measure is bounded to the unit interval. Specifying the option `base(input)` invokes these deviations from the default procedure. One may optionally (option `notwosided`) stick to one-sided truncation and consider truncation only from the right when analyzing input-oriented efficiency. Using option `notwosided` seems questionable because it rests on simulating a data-generating process that is inconsistent with the nonnegative nature of θ_i. In particular, `notwosided` is not recommended with algorithm 2.[18]

- One may opt for the Shephard rather than the Farrell distance measure (option `invert`). This simply means that all (internally; see below) estimated scores are inverted through all steps of the estimation procedure. If constant returns to scale are assumed for the production technology, this is equivalent to switching from output- to input-oriented efficiency. For variable and nonincreasing returns, this one-to-one correspondence does not hold. If option `invert` is specified with output-oriented efficiency, the same changes to the estimation procedure apply as described above with respect to option `base(input)` (without option `invert`). Considering the input-oriented Farrell or the output-oriented Shephard efficiency measure, which are both bounded to the unit interval, may lead to counterintuitive results when performing the bias correction in algorithm 2. More precisely, it may happen that the bias-corrected scores are negative for some DMUs. Negative scores do not enter the truncated regression analysis, unless option `notwosided` is specified. If negative efficiency measures occur, `simarwilson` issues a warning and recommends switching to Farrell output-oriented or Shephard input-oriented

[17] We would like to thank Ramon Christen for suggesting implementing two-sided truncation in `simarwilson`.

[18] Sampling from the right-truncated normal distribution may result in $\widetilde{\vartheta}_i < 0$ (step 3.2) and, in consequence, may make the bias correction fail [$\widetilde{\mathbf{x}}_i = (\widehat{\vartheta}_i/\widetilde{\vartheta}_i)\mathbf{x}_i < 0$; step 3.3, algorithm 2]. For this reason, `notwosided` is ignored in step 3.1 of algorithm 2.

efficiency, for which bias correction cannot result in negative scores. Yet ultimately the decision of how to respond to this problem is up to the user.

- Related to the discussion in Simar and Wilson (2007, 45), one may assume a data-generating process that deviates from (1) by considering log (in)efficiency as the left-hand-side variable (option `logscore`); that is,

$$\ln(\theta_i) = \mathbf{z}_i\boldsymbol{\beta} + \epsilon_i \tag{2}$$

  Here ϵ_i is assumed to be truncated normally distributed, with left-truncation at $-\mathbf{z}_i\boldsymbol{\beta}$.[19] If $\ln(\vartheta_i)$ is considered as the left-hand-side variable, truncation at $-\mathbf{z}_i\boldsymbol{\beta}$ is from the right. If all $\widehat{\theta}_i$ are close to unity, specifying the option `logscore` will make little difference. Yet, if the data include DMUs that according to the DEA are very inefficient, specifying `logscore` may result in a model specification that is more easily estimated in the truncated regressions.

- `simarwilson` allows for restricting the reference set for the DEA to a subset of the considered DMUs (option `reference()`); compare figure 4. Unlike `teradial`, it does not allow for considering DMUs as elements of the reference set for which no efficiency scores are estimated.[20] Restricting the reference set to a subsample of the considered DMUs will regularly result in some irregular (superefficient) estimated scores. Such DMUs are ignored in the truncated regressions. In general, restricting the reference set makes the DEA model substantially deviate from what is considered in Simar and Wilson (2007). Hence, users should carefully think about whether using the option `reference()` makes sense in their application.

- `simarwilson` allows for using efficiency scores that were estimated beforehand by some estimation procedure using Stata[21] or any other software. This effectively means that step 1 in algorithm 1 is skipped. If externally estimated, bias-corrected scores are available, one may in principle also skip steps 1–4 in algorithm 2. However, the bias-correction procedure suggested above is specific and incorporates the assumptions on which the subsequent steps are based. Appropriate bias-corrected scores will hence rarely be available. The scores calculated by `teradialbc`, though similar in some respects (compare Simar and Wilson [2007]), deviate from what is computed in steps 1–4 of algorithm 2. Because using any kind of numeric, nonnegative variable as an externally estimated score is technically feasible, it is the user's responsibility to make sure that this variable is a radial measure of technical efficiency.

[19] Note that (2) is not equivalent to $\ln(\theta_i - 1) = \mathbf{z}_i\boldsymbol{\beta} + \zeta_i$, with $\zeta_i \sim N\left(0, \sigma^2\right)$, which might—erroneously—be regarded as an obvious choice. Unlike (2), this process not only assumes $\theta_i = 1$ to occur with zero probability but also regards full efficiency as genuinely impossible. This conflicts with the production-possibility set, including its boundary. Moreover, unlike (2), the above process assumes that for any DMU, reaching some neighborhood of $\theta_i = 1$ is relatively unlikely; that is, $\Pr\left(1 \leq \theta_i < 1 + \tau\right) < \Pr\left(1 + \tau \leq \theta_i < 1 + 2\tau\right) \forall i$ if $\tau \to 0$.

[20] This is technically infeasible because in steps 3.2 and 3.3 (algorithm 2), an estimated score is required for any DMU that contributes to the artificial reference set.

[21] Besides `teradial`, the community-contributed `dea` command (Ji and Lee 2010) allows for this.

- `simarwilson` allows for weighted estimation (only `pweights` and `iweights` are allowed). Note that weights are immaterial for the DEA steps within `simarwilson` but only affect truncated regression estimation. Zero weights can hence be used for excluding some DMUs from the truncated regression analysis that are considered in the DEA.

3 The simarwilson command

`simarwilson` requires Stata 12 or higher. Unless externally estimated efficiency scores are used, `simarwilson` requires the community-contributed command `teradial`, including the associated plugin (Badunenko and Mozharovskyi 2016). With internal DEA, the number of observations is limited by the value of `matsize` (see [R] **matsize**), which is 11,000 at the maximum. The prefix commands `by` and `svy` are not allowed. The prefix command `bootstrap` is technically allowed with externally estimated scores. However, using it is entirely counterproductive. `pweights` (default) and `iweights` are allowed; see [U] **11.1.6 weight**. Weights affect only the truncated regression steps within `simarwilson` but not the DEA steps. If `iweights` are used, (regression) numbers of observations are expressed in terms of rounded sums of weights.

3.1 Syntax

The syntax for `simarwilson` is

```
simarwilson [ (outputs = inputs) ] [ depvar ] indepvars [ if ] [ in ] [ weight ]
   [ , algorithm(1|2) notwosided logscore nounit rts(crs|nirs|vrs)
   base(output|input) reference(varname) invert tename(newvar)
   tebc(newvar) biaste(newvar) reps(#) bcreps(#) saveall(name)
   bcsaveall(name) dots cinormal bbootstrap level(#) noomitted
   baselevels noprint nodeaprint trnoisily maximize_options ]
```

outputs is the list of outputs from the production process, and *inputs* is the corresponding list of inputs. Either *varlist* may include only numeric, nonnegative variables. Factor variables and time-series operators are not allowed. The number of output and input variables must not exceed the number of considered DMUs.

depvar specifies an existing variable that contains an externally estimated efficiency measure (score) meant to enter the regression model as a dependent variable. Specifying *depvar* is possible only if `(`*outputs* `=` *inputs*`)` is not specified. That means, with `(`*outputs* `=` *inputs*`)` specified, any variable in the following *varlist* is interpreted as an element of *indepvars*. `simarwilson` expects *depvar* to be a radial efficiency measure that is bounded either to the $(0,1]$ interval or to the $[1,\infty)$ interval. This implies that *depvar* must not be measured in percent. If some values of *depvar* are smaller than 1 while others exceed 1, `simarwilson` issues a warning and ignores observations depending on how the option `nounit` is specified. This may happen if the preceding efficiency analysis is carried out using a reference set that does not include all observations for which efficiency scores are estimated. Note that Simar and Wilson (2007) do not consider this case. Only numeric and strictly positive values are allowed for *depvar*.

indepvars denotes the list of explanatory variables. Unlike *outputs* and *inputs*, factor variables are allowed in *indepvars*; see [U] **11.4.3 Factor variables**. Time-series operators such as `L.` and `F.` are not allowed.

3.2 Options

`algorithm(1|2)` specifies whether algorithm 1 or 2 is applied. To calculate bias-corrected efficiency scores, algorithm 2 involves another bootstrap procedure that loops over DEA. `algorithm(2)` requires `(`*outputs* `=` *inputs*`)` to be specified. If one uses external DEA scores as *depvar*, one has to opt for `algorithm(1)` even if the externally estimated scores are bias corrected.[22] The default is `algorithm(1)`.

`notwosided` makes `simarwilson` apply a one-sided truncated regression model, irrespective of whether (regular) efficiency scores are bounded to the $(0,1]$ interval or to the $[1,\infty)$ interval. For (regular) scores within $(0,1]$, the default (`twosided`) is to use a two-sided truncated regression model and to sample from the two-sided truncated normal distribution. With `twosided`, the procedure hence considers that input-oriented (Farrell) efficiency scores are not only less than or equal to 1 but also strictly positive. The latter is ignored with `notwosided`. Hence, with `notwosided`, `simarwilson` in a mirror-inverted way applies the procedure suggested in Simar and Wilson (2007), who only consider scores within $[1,\infty)$, to efficiency scores within $(0,1]$. That is, with `notwosided` specified, the regression model at the second stage of `simarwilson` does not differ between output- and input-oriented efficiency, except for the truncation being either from the right or from the left. For (regular) efficiency scores ≥ 1, specifying `notwosided` has no effect. `notwosided` is not recommended with `algorithm(2)`; compare footnote 18.

`logscore` makes `simarwilson` use the natural logarithm of the efficiency score as the left-hand-side variable in the truncated regressions. With `logscore` specified, truncation is at 0 rather than at 1 and is always one sided. If externally estimated scores are used, one must not take the logarithm beforehand but let the original score enter the estimation procedure as *depvar*.

`nounit` specifies whether inefficiency is indicated by efficiency score < 1 (`unit`) or by efficiency score > 1 (`nounit`). Specifying this option will rarely be necessary. If the DEA is carried out internally, `simarwilson` internally sets `nounit` depending on how the options `base()` and `invert` are specified. If externally estimated scores are used and all observations of *depvar* are either in the $(0,1]$ or in the $[1,\infty)$ interval, specifying the `nounit` option is also not required, because `simarwilson` recognizes which DMUs are inefficient and which are efficient. Only if external scores are used

[22] With bias-corrected (externally estimated) scores in hand, steps 2–4 of algorithm 1 are fully equivalent to steps 5–7 of algorithm 2; compare section 2.2.

that are bounded neither to the $(0,1]$ interval nor to the $[1,\infty)$ interval, `nounit` is required to specify which observations of *depvar* are regular (inefficient) and which are irregular (superefficient). Note that Simar and Wilson (2007) do not consider irregular (superefficient) DMUs.

`rts(crs|nirs|vrs)` specifies under which assumption regarding the returns to scale of the considered production process the measure of technical efficiency is estimated. `crs` requests constant returns to scale, `nirs` requests nonincreasing returns to scale, and `vrs` requests variable returns to scale. The default is `rts(vrs)`.[23] `rts()` is passed through to `teradial`. If externally estimated scores are used, specifying `rts()` has no effect.

`base(output|input)` specifies the orientation or base of the radial measure of technical efficiency. `output` requests output orientation, while `input` requests input orientation. The default is `base(output)`. `base()` is passed through to `teradial` and has no effect if externally estimated scores are used.

`reference(`*varname*`)` specifies the indicator variable that defines which data points of *outputs* and *inputs* (DMUs) form the technology reference set. *varname* needs to be binary (numeric or string), with the (alphanumerically) larger value indicating being part of the reference set. Because for each reference DMU an efficiency score is required when running `simarwilson`, the full set of DMUs or a subset of DMUs may serve as the reference set. Yet the reference set may not include any observations for which technical efficiency is not estimated. This precludes the specification `(`*ref_outputs* `=` *ref_inputs*`)`, which is allowed in `teradial`. Specifying a subset of observations as a reference set will frequently result in irregular efficiency estimates (superefficient DMUs). Note that Simar and Wilson (2007) consider the full set of observations as a reference set. Hence, specifying a subset as reference results in a DEA model that substantially deviates from what is assumed in Simar and Wilson (2007).

`invert` makes `simarwilson` calculate and use the Shephard instead of the Farrell (default) efficiency measure. That is, all estimated efficiency scores are inverted, unless they were externally estimated. With option `invert`, scores smaller than one indicate inefficiency for the output-oriented efficiency measures (that is, the factor by which output generation proportionally falls short of what is technically feasible), and scores larger than one indicate inefficiency for the input-oriented efficiency measure (that is, the factor by which input utilization proportionally exceeds what is technically feasible). `invert` is redundant for `rts(crs)` because for constant returns to scale, input-oriented efficiency is just the reciprocal of output-oriented efficiency. Hence, rather than specifying `invert`, one can just switch the base. Yet this does not hold for `rts(nirs)` and `rts(vrs)`. With externally estimated scores, specifying `invert` has no effect. One rather has to manually invert the externally estimated scores prior to running `simarwilson`, if one wants to switch between the Farrell and the Shephard measures.

[23] Simar and Wilson (2007, 37) proposed their procedures for the DEA estimator under the variable returns assumption.

`tename(`*newvar*`)` creates a new variable *newvar* that contains estimates of radial technical efficiency (DEA scores).

`tebc(`*newvar*`)` creates a new variable *newvar* that contains bias-corrected estimates of radial technical efficiency (bias-corrected DEA scores). `tebc(`*newvar*`)` requires `algorithm(2)`.

`biaste(`*newvar*`)` creates a new variable *newvar* that contains a bootstrap bias estimate for original radial measures of technical efficiency. `biaste(`*newvar*`)` requires `algorithm(2)`.

`reps(`#`)` specifies the number of bootstrap replications for estimating confidence intervals and standard errors for the regression coefficients. The default is `reps(1000)`.

`bcreps(`#`)` specifies the number of bootstrap replications for the bias correction of DEA scores. The default is `bcreps(100)` as suggested in Simar and Wilson (2007).

`saveall(`*name*`)` makes `simarwilson` save all bootstrap estimates of the regression coefficients to the (`reps()` $\times K + 1$) Mata matrix *name*. Any existing Mata matrix *name* is replaced. This option is useful for reporting confidence intervals for different levels of confidence.

`bcsaveall(`*name*`)` makes `simarwilson` save all bootstrap efficiency scores that are estimated in the bias-correction procedure to the (`bcreps()` × `e(N_dea)`) Mata matrix *name*. Any existing Mata matrix *name* is replaced. Depending on `bcreps(`#`)` and the number of considered DMUs, the saved Mata matrix may be huge.

`dots` makes `simarwilson` display one dot character for each bootstrap replication.

`cinormal` makes `simarwilson` display normal-approximated confidence intervals rather than percentile-based bootstrap confidence intervals for the regression coefficients. One may change the reported type of confidence intervals by retyping `simarwilson` without arguments and specifying only the option `cinormal`.

`bbootstrap` makes `simarwilson` display mean bootstrap coefficients rather than the original coefficients from fitting the truncated regression model. One may change the type of the reported coefficient vector by retyping `simarwilson` without arguments and specifying only the option `bbootstrap`.

`level(`#`)`; see [R] **level** estimation options. One may change the reported confidence level by retyping `simarwilson` without arguments and specifying only the option `level(`#`)`. For percentile-based confidence intervals, this requires the option `saveall(`*name*`)`.

`noomitted` specifies that variables that were omitted because of collinearity not be displayed. The default is to include in the results table any variables omitted because of collinearity and to label them as omitted with the `o.` prefix.

`baselevels` makes `simarwilson` display base categories of factor variables in the results table and label them as base by with the `#b.` prefix.

`noprint` prevents `simarwilson` from displaying warnings. Error messages are displayed irrespective of whether `noprint` is specified.

`nodeaprint` prevents `simarwilson` from displaying DEA output.

`trnoisily` makes `simarwilson` display genuine output of `truncreg` for the initial truncated regression or regressions (not for truncated regressions within bootstrap procedures). Specifying this option might be useful if `simarwilson` issues the error message `truncated regression failed` or `convergence not achieved in truncated regression` and the accompanying return code is inconclusive about what makes `truncreg` fail.

maximize_options allows for all maximization options that are allowed with `truncreg`, which are simply passed through; see [R] **maximize**. Moreover, one may specify the `truncreg` options `noconstant`, `offset(`*varname*`)`, and `constraints(`*constraints*`)`, which are also passed through; see [R] **truncreg**.

3.3 Stored results

`simarwilson` stores the following results to `e()`:

Scalars

`e(N)`	number of observations (inefficient DMUs)
`e(k_eq)`	number of equations in `e(b)`
`e(df_m)`	model degrees of freedom
`e(ll)`	pseudo-log likelihood (initial truncated regression)
`e(chi2)`	χ^2
`e(p)`	model significance, p-value
`e(ic)`	number of iterations (initial truncated regression)
`e(rc)`	return code
`e(converged)`	`1` converged, `0` otherwise (initial truncated regression)
`e(N_lim)`	number of limit observations (efficient DMUs)
`e(N_irreg)`	number of irregular observations (superefficient DMUs)
`e(N_all)`	overall number of observations (DMUs)
`e(N_reps)`	number of completed bootstrap replications
`e(N_misreps)`	number of failed bootstrap replications
`e(N_dea)`	number of DMUs for which efficiency scores are estimated
`e(N_dearef)`	number of DMUs in reference set
`e(N_deaneg)`	number of negative bias-corrected scores
`e(N_bc)`	number of completed bootstrap replications (bias correction)
`e(level)`	confidence level
`e(sigma)`	estimate of sigma
`e(wgtsum)`	sum of weights (if weights are specified)
`e(algorithm)`	algorithm used (`1` or `2`)
`e(k_aux)`	number of auxiliary parameters
`e(noutps)`	number of output variables
`e(ninps)`	number of input variables

Macros

`e(cmd)`	`simarwilson`
`e(cmdline)`	command as typed
`e(depvar)`	either `efficiency` or `inefficiency`
`e(depvarname)`	name of left-hand-side variable (if provided as *depvar* or saved as *newvar*)
`e(wtype)`	weight type
`e(wexp)`	weight expression
`e(title)`	`Simar & Wilson (2007) two-stage efficiency analysis`
`e(shorttitle)`	`Simar & Wilson (2007) eff. analysis`
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`
`e(marginsok)`	predictions allowed by `margins`
`e(marginsdefault)`	default `predict()` specification for `margins`
`e(unit)`	either `unit` or `nounit`
`e(truncation)`	either `twosided` or `onesided`
`e(logscore)`	`logscore` if the option `logscore` is specified
`e(saveall)`	*name* if `saveall(`*name*`)` is specified
`e(bcsaveall)`	*name* if `bcsaveall(`*name*`)` is specified
`e(cinormal)`	`cinormal` (if `cinormal` is specified)
`e(bbootstrap)`	`bbootstrap` (if `bbootstrap` is specified)
`e(scoretype)`	either `score` or `bcscore`
`e(deatype)`	either `internal` or `external`
`e(invert)`	either `Farrell` or `Shephard`
`e(biaste)`	*varname* of estimated bias (if `biaste()` is specified)
`e(tebc)`	*varname* of estimated bias-corrected efficiency (if `tebc()` is specified)
`e(tename)`	*varname* of estimated uncorrected efficiency (if `tename()` is specified)
`e(rts)`	returns to scale (`CRS` or `NIRS` or `VRS`) (if DEA is internal)
`e(base)`	base and orientation (`output` or `input`) (if DEA is internal)
`e(outputs)`	*varlist* of outputs (if DEA is internal)
`e(inputs)`	*varlist* of inputs (if DEA is internal)

Matrices

`e(b)`	coefficient vector
`e(Cns)`	constraints matrix (if `constraints()` is specified)
`e(V)`	variance–covariance matrix of the estimators
`e(b_bstr)`	bootstrap estimates of coefficients
`e(bias_bstr)`	bootstrap estimated biases
`e(ci_percentile)`	bootstrap percentile confidence intervals

Functions

`e(sample)`	marks estimation sample (truncated regression)

Note that `e(sample)` and `e(N)` refer to those observations that enter the truncated regression analysis.

3.4 simarwilson postestimation

The postestimation commands available after `simarwilson` are almost the same as for `truncreg`; see [R] **truncreg postestimation**. Among others, these are `test`, `testnl`, `lincom`, `nlcom`, `predict`, `predictnl`, and [R] **margins**. `margins, dydx(`*indepvars*`)` appears to be particularly valuable. After `simarwilson`, `margins` behaves slightly differently than it behaves after `truncreg`. The default is to estimate marginal effects on expected (in)efficiency that is on $E\left(\theta_i|\theta_i > 1, \mathbf{z}_i\right)$ (Farrell output oriented) and $E(\vartheta_i|0 < \vartheta_i < 1, \mathbf{z}_i)$ (Farrell input oriented), respectively.[24] That is, `margins`, by default, internally sets the `predict(e(1,.))` and `predict(e(0,1))` options, respectively.[25] If one wants to estimate marginal effects on the linear index, specifying the option `predict(xb)` is required. The options `predict(ystar(`*a*`,`*b*`))` and `predict(pr(`*a*`,`*b*`))` are not allowed with `margins` after `simarwilson`. They make `margins` consider a censored outcome, which makes little sense with `simarwilson`. Note that some postestimation commands may behave differently than described in Stata versions earlier than 15. For instance, in Stata 13 and earlier, the default for `margins` is `predict(xb)`.

One should generally be careful in interpreting the results from postestimation commands, such as `predict`, used after `simarwilson`. The postestimation commands treat the results of `simarwilson` as if they were generated by `truncreg`. However, one should be aware that in terms of the underlying model, both are not the same. Besides the estimated variance–covariance matrix, the key difference is that `truncreg` usually assumes that the left-hand-side variable of the data-generating process is observed for not-truncated observation and may in principle also be observable for truncated observations. In contrast, `simarwilson` rests on the assumption that the true outcome variable is genuinely unobservable. Moreover, while in many applications of `truncreg`, truncation originates from missing information, for `simarwilson`, truncation is a genuine feature of the data-generating process; see section 2.

4 An application of simarwilson

4.1 Comparison of estimation methods

To illustrate how `simarwilson` can be used in applied work, in this section we use the command for empirically addressing the question of whether the quality of governance, including quality of the judicial system, at the national level matters for the efficiency of gross domestic product (GDP) generation. The analysis is based on cross-country data provided through the Penn World Table database, version 9 (Feenstra, Inklaar, and Timmer (2015) and the World Economic Forum, Global Competitiveness Report, version 2018-02-26 (World Economic Forum 2018; Schwab 2017). Though both databases are publicly available on the Internet, only the Penn World Table is provided in a format that can directly be used with Stata. For this reason, this article is accompanied by the community-contributed command `gciget`, which facilitates the retrieval of the Global Competitiveness Index (GCI) data using Stata. See section 4.3 for a more detailed de-

[24] Because the data-generating process (1) assumes a truncated distribution for ε_i, $E(\theta_i|\theta_i > 1, \mathbf{z}_i)$ coincides with $E(\theta_i|\mathbf{z}_i)$. That is, $\theta_i < 1$ is not only unobserved but also impossible according to (1). The same argument applies to $E(\vartheta_i|0 < \vartheta_i < 1, \mathbf{z}_i)$ as well as to the Shephard measures. This argument would not hold for standard applications of `truncreg`, which illustrates that the `truncreg` postestimation commands should be used with some caution after `simarwilson`. With the options `nounit` and `notwosided` simultaneously specified, `margins` by default considers $E(\vartheta_i|\vartheta_i < 1, \mathbf{z}_i)$; that is, the option `predict(e(.,1))` is internally set.

[25] With option `logscore`, the default is `predict(e(0,.))` and `predict(e(.,0))`, respectively. That is, `margins` yields semielasticities by default.

scription of gciget. The output below is from using gciget to load three selected variables (EOSQ048, EOSQ051, EOSQ144) of the GCI into Stata and merging them to the Penn World Table data.

```
. gciget EOSQ048 EOSQ051 EOSQ144
DISCLAIMER: The World Economic Forum is the provider of the Global
  Competitiveness Index 2017-2018, a framework and a corresponding set of
  indicators for 137 economies. The software gciget.ado provides a practical
  way to read the indicators into Stata (R). The responsibility of complying
  with the terms and conditions of use under which the owner of the data
  grants access to the indicators is entirely with the user but not with the
  authors of the software gciget.ado. Any user of gciget.ado is responsible
  for making him or herself familiar with the terms of use under which she or
  he is allowed to work with the data of the Global Competitiveness Index. For
  more information and methodology, please see http://wef.ch/gcr17. In no
  event will the authors, owners, and creators of gciget.ado, or their
  employers or any other party who may modify and/or redistribute this
  software, accept liability for any loss or damage suffered as a result of
  using the gciget.ado software.
Downloading the GCI_Dataset_2007-2017.xlsx file
Importing the GCI_Dataset_2007-2017.xlsx file
Processing EOSQ048:  1.09 Burden of government regulation, 1-7 (best)
Processing EOSQ051:  1.01 Property rights, 1-7 (best)
Processing EOSQ144:  1.06 Judicial independence, 1-7 (best)
. quietly merge 1:1 countrycode year using
> "https://www.rug.nl/ggdc/docs/pwt90.dta"
```

We consider a national-level production process that generates the single output real GDP (rgdpo) by using three inputs: capital stock (ck), number of persons engaged (emp), and human capital (hc). We assume variable returns to scale and consider the output-oriented Farrell efficiency measure. We consider the burden of government regulation (EOSQ048), property rights protection (EOSQ051), and judicial independence (EOSQ144) as key explanatory variables. While the rest of the data used are from the Penn World Table, the latter three variables are provided through the Global Competitiveness Report. These indices are measured on a continuous scale ranging from 1 to 7 and originate from answers to the following questions in the World Economic Forum, Executive Opinion Survey (see Schwab [2017] appendix C for details): “In your country, how burdensome is it for companies to comply with public administration’s requirements (for example, permits, regulations, reporting)? [1 = extremely burdensome; 7 = not burdensome at all]”; “In your country, to what extent are property rights, including financial assets, protected? [1 = not at all; 7 = to a great extent]”; “In your country, how independent is the judicial system from influences of the government, individuals, or companies? [1 = not independent at all; 7 = entirely independent]” (World Economic Forum 2018). To address possible endogeneity concerns regarding these regressors, we let them enter the model as lagged values. In addition to the three explanatory variables of primary interest, we include lagged log-population (lpop) as control. To account for possible country-size-related heterogeneity in the link between governance quality and national efficiency, we interact the governance-quality indices with lpop in the regression models.

After loading the working data into Stata's memory, we generate the explanatory variables that we actually need in the empirical analysis and give them more telling names. Because `simarwilson` does not allow for time-series operators, we generate lagged values "by hand". To make the code easier to read, we place the governance-quality variables in the global macro `g_list` and define the global macro `z_list`, which contains the comprehensive list of explanatory variables. Because the sample size is relatively small, we opt for a rather generous level of significance by setting the confidence level to 90%. Moreover, we set a new seed for Stata's random-number generator.[26] To facilitate the replication of results, the random-number generator is reset to this state every time `simarwilson` runs in the application. To preserve the spirit of randomness, you should avoid this in your own applications.

```
. quietly generate regu = EOSQ048[_n-1] if countrycode == countrycode[_n-1]
. quietly generate prop = EOSQ051[_n-1] if countrycode == countrycode[_n-1]
. quietly generate judi = EOSQ144[_n-1] if countrycode == countrycode[_n-1]
. quietly generate lpop = ln(pop[_n-1]) if countrycode == countrycode[_n-1]
. global g_list "regu prop judi"
. global z_list "regu prop judi lpop c.regu#c.lpop c.prop#c.lpop c.judi#c.lpop"
. set level 90
. set seed 341566575
```

Second, we use `teradial` to generate externally estimated DEA efficiency scores (`te_vrs_o`) using the most recent year that is available in the data, that is, 2014. We restrict the DEA to countries for which information on all right-hand-side variables is available.[27] Because we do not define a reference set that deviates from the sample for which efficiency measures are estimated, the option `base(output)` makes `te_vrs_o` take values equal to or greater than 1.[28] Then, we let Stata report descriptive statistics for the variables used in the subsequent regressions. Because of missing information in some variables, only 131 countries out of 182 covered by the Penn World Table can be used for estimation.

```
. teradial rgdpo = ck emp hc if year == 2014 & regu <. & prop <. & judi <.
> & lpop <., tename(te_vrs_o) rts(vrs) base(output) noprint
. summarize te_vrs_o regu prop judi lpop if e(sample)
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
    te_vrs_o |        131    1.699949    .6236905          1   5.513838
        regu |        131    3.435143    .6711715   1.846199    5.42263
        prop |        131    4.304648    1.030568   1.610298   6.378975
        judi |        131    3.897085    1.315987   1.113236   6.678279
        lpop |        131    2.566502    1.586448  -1.264066   7.217087
```

[26] The default random-number generator (`mt64`) of Stata 15 is used.

[27] This makes the DEA steps in algorithms 1 and 2 using the same sample. Only for the latter are the right-hand-side variables required for estimating (bias-corrected) efficiency scores; compare step 3, algorithm 2, page 956.

[28] Unless the option `invert` is used, a positive coefficient of a variable in `simarwilson` would imply it has a negative effect on efficiency. Conversely, a negative coefficient would mean that the variable has a positive effect on efficiency.

In the next step of the analysis, we use four empirical models to explain (in)efficiency in the GDP generation. Besides `simarwilson, algorithm(1)`, and `algorithm(2)`, we also consider `tobit` and `truncreg` for comparison. Because the model coefficients themselves cannot straightforwardly be interpreted in quantitative terms, we use `margins, dydx()` to estimate average marginal effects of the governance-quality indices on national GDP efficiency.

We start with tobit estimation, which—according to Simar and Wilson (2007)—erroneously regards full efficiency (`te_vrs_o = 1`) as an outcome of the underlying data-generating process rather than an artifact of finite-sample bias.[29] Consistent with this misinterpretation, we use the option `predict(ystar(1,.))` with `margins`. Estimated marginal effects are not displayed but stored with `estimates store` for later comparison. The output from `tobit` reveals that, according to DEA, 18 countries are fully efficient while 113 are found to be inefficient. With judicial independence being the only exception, the governance variables are individually significant at the 10% level and bear the expected negative signs. However, because the model includes several interactions with log population, making any statement about the link between governance quality and GDP efficiency is hardly possible without examining marginal effects. At least, the signs of coefficients attached to the interaction variables seem to indicate that possible efficiency gains through less business regulation and better protection of property rights are first a matter of small countries.

[29] The options `nolstretch` and `vsquish` are just for making the displayed output fit on a printed page.

```
. tobit te_vrs_o $z_list, ll(1) nolstretch vsquish
Refining starting values:
Grid node 0:   log likelihood = -130.85914
Fitting full model:
Iteration 0:   log likelihood = -130.85914
Iteration 1:   log likelihood = -128.73992
Iteration 2:   log likelihood = -128.71027
Iteration 3:   log likelihood =  -128.7102
Iteration 4:   log likelihood =  -128.7102
Tobit regression                                Number of obs     =        131
                                                   Uncensored     =        113
Limits: lower = 1                                  Left-censored  =         18
        upper = +inf                               Right-censored =          0
                                                LR chi2(7)        =      20.43
                                                Prob > chi2       =     0.0047
Log likelihood =  -128.7102                     Pseudo R2         =     0.0735

------------------------------------------------------------------------------
    te_vrs_o |      Coef.   Std. Err.      t    P>|t|     [90% Conf. Interval]
-------------+----------------------------------------------------------------
        regu |  -.3925008   .2014823    -1.95   0.054    -.7264042   -.0585973
        prop |  -.5199721   .2574393    -2.02   0.046    -.9466096   -.0933347
        judi |   .2488415   .1888903     1.32   0.190     -.064194    .5618771
        lpop |  -.8211409   .2667289    -3.08   0.003    -1.263173   -.3791084
      c.regu#|
      c.lpop |   .1484147   .0687327     2.16   0.033     .0345084    .2623209
      c.prop#|
      c.lpop |   .1251518   .0871451     1.44   0.153    -.0192682    .2695717
      c.judi#|
      c.lpop |  -.0858924   .0693701    -1.24   0.218    -.2008549    .0290701
       _cons |   4.589835   .7747277     5.92   0.000     3.305929    5.873741
-------------+----------------------------------------------------------------
var(e.te_v~o)|   .4098449   .0562185                      .3265083     .514452
------------------------------------------------------------------------------
. quietly margins, dydx($g_list) predict(ystar(1,.)) post
. estimates store tobit
```

Then, we turn to the truncated regression by using `truncreg`. Unlike `tobit`, this approach drops observations for which `te_vrs_o = 1` holds, so we use the option `predict(e(1,.))` when estimating marginal effects. The estimated coefficients look quite different compared with their counterparts from `tobit`, yet in terms of the signs, the results are similar to their counterparts from `tobit`. According to the results from `truncreg`, judicial independence seems to matter for efficiency because both `judi` and its interaction with `lpop` are statistically significant at the 10% level. This points to judicial independence being negatively associated with efficiency, at least in small countries. However, following the argument of Simar and Wilson (2007), this result might be an artifact of incorrectly estimated standard errors.

```
. truncreg te_vrs_o $z_list, ll(1) nolstretch vsquish
(note: 18 obs. truncated)

Fitting full model:

Iteration 0:   log likelihood = -76.432745
Iteration 1:   log likelihood = -68.515376
Iteration 2:   log likelihood = -67.617067
Iteration 3:   log likelihood = -67.606346
Iteration 4:   log likelihood = -67.606307
Iteration 5:   log likelihood = -67.606307

Truncated regression
Limit:   lower =          1                     Number of obs     =        113
         upper =       +inf                     Wald chi2(7)      =      18.90
Log likelihood = -67.606307                     Prob > chi2       =     0.0085

------------------------------------------------------------------------------
    te_vrs_o |      Coef.   Std. Err.      z    P>|z|     [90% Conf. Interval]
-------------+----------------------------------------------------------------
        regu |  -.9258069   .4299483    -2.15   0.031    -1.633009   -.2186048
        prop |  -1.243902   .4991533    -2.49   0.013    -2.064936   -.4228676
        judi |   .7784162   .3780368     2.06   0.039      .156601    1.400231
        lpop |  -1.739993   .5952224    -2.92   0.003    -2.719046   -.7609389
      c.regu#|
      c.lpop |   .4253728   .1720618     2.47   0.013     .1423563    .7083893
      c.prop#|
      c.lpop |   .2581352   .1794841     1.44   0.150    -.0370899    .5533604
      c.judi#|
      c.lpop |  -.2592945   .1497392    -1.73   0.083    -.5055935   -.0129955
       _cons |   7.447817   1.629842     4.57   0.000     4.766965    10.12867
-------------+----------------------------------------------------------------
      /sigma |   .7222912   .0925133     7.81   0.000     .5701204    .8744621
------------------------------------------------------------------------------

. quietly margins, dydx($g_list) predict(e(1,.)) post

. estimates store truncreg
```

Hence, in the next step, we turn to `simarwilson, algorithm(1)`. Because externally estimated efficiency scores are already available, we do not rerun the DEA within `simarwilson` but use `te_vrs_o` as the dependent variable. Using the `(rgdpo = ck emp hc)` syntax instead and specifying the options `rts(vrs)` and `base(output)` would have generated identical results. Because we report percentile confidence intervals for the coefficients, we request a large number (2,000) for the bootstrap replications. This choice results in a substantial computing time of 88 seconds (Stata/SE 15.1).[30] Specifying the option `predict()` is not required for `margins`, because the appropriate specification is set internally. As a practical matter, we advise you use one processor in Stata/MP by typing `set processors 1` before executing `simarwilson`. The estimated coefficients necessarily coincide with what we got from `truncreg` because `simarwilson, algorithm(1)` affects only the estimated standard errors and confidence intervals. Yet, even with respect to them, the deviation from their conventional counterparts from `truncreg` is rather moderate. This is in line with what is frequently found in applications of algorithm 1.

[30] Carrying out the DEA internally affects computing time just marginally.

```
. simarwilson te_vrs_o $z_list, reps(2000)
Simar & Wilson (2007) eff. analysis      Number of obs               =       113
(algorithm #1)                           Number of efficient DMUs    =        18
                                         Number of bootstr. reps     =      2000
                                         Wald chi2(7)                =     21.63
inefficient if te_vrs_o > 1              Prob > chi2(7)              =    0.0029
------------------------------------------------------------------------------
Data Envelopment Analysis:                         externally estimated scores
------------------------------------------------------------------------------
             |   Observed   Bootstrap                         Percentile
inefficiency |      Coef.   Std. Err.      z    P>|z|     [90% Conf. Interval]
-------------+----------------------------------------------------------------
te_vrs_o     |
        regu |  -.9258069    .4021808    -2.30   0.021    -1.617009   -.2838173
        prop |  -1.243902    .4715584    -2.64   0.008    -2.042034   -.5066595
        judi |   .7784162     .356048     2.19   0.029     .1985666    1.371619
        lpop |  -1.739993    .5688841    -3.06   0.002    -2.670876   -.8296476
c.regu#c.l~p |   .4253728    .1611459     2.64   0.008     .1649802    .6971241
c.prop#c.l~p |   .2581352    .1692766     1.52   0.127     -.013744    .5414581
c.judi#c.l~p |  -.2592945    .1400455    -1.85   0.064    -.4829779   -.0310317
       _cons |   7.447817    1.557957     4.78   0.000     4.988534    9.974978
-------------+----------------------------------------------------------------
      /sigma |   .7222912    .0877174     8.23   0.000     .5537709    .8368938
------------------------------------------------------------------------------
. quietly margins, dydx($g_list) post
. estimates store alg_1
```

Then, we turn to `algorithm(2)`. In this procedure, tailored, bias-corrected efficiency scores enter the regression model at the left-hand side. Hence, we cannot use externally estimated scores but instead let `simarwilson` carry out the bias correction internally. This requires the `(rgdpo = ck emp hc)` syntax and the options `rts(vrs)` and `base(output)`. The last two determine the DEA model used. By specifying the option `tebc(tebc_vrs_o)`, we save the estimated, bias-corrected efficiency scores for possible later use. We opt for 1,000 replications in the bias-correction bootstrap, which is well above the default suggested in Simar and Wilson (2007). Fitting this model takes 96 seconds. Because of the relatively small sample size, using `algorithm(2)` increases computing time by only 10%; compare footnote 16. Because we do not use externally estimated scores as the left-hand-side variable but let `simarwilson` run the DEA internally, the reported output also involves comprehensive information about the DEA model used.[31] In this application, using bias-corrected instead of uncorrected scores has only a moderate impact on the estimated coefficients and the associated estimated confidence intervals.

[31] The option `nodeaprint` suppresses displaying the DEA-related information.

```
. simarwilson (rgdpo = ck emp hc) $z_list if year == 2014, algorithm(2)
> rts(vrs) base(output) reps(2000) bcreps(1000) tebc(tebc_vrs_o)

Simar & Wilson (2007) eff. analysis     Number of obs             =       131
(algorithm #2)                          Number of efficient DMUs  =         0
                                        Number of bootstr. reps   =      2000
                                        Wald chi2(7)              =     21.10
inefficient if tebc_vrs_o > 1           Prob > chi2(7)            =    0.0036
─────────────────────────────────────────────────────────────────────────────
Data Envelopment Analysis:              Number of DMUs            =       131
                                        Number of ref. DMUs       =       131
output oriented (Farrell)               Number of outputs         =         1
variable returns to scale               Number of inputs          =         3
bias corrected efficiency measure       Number of reps (bc)       =      1000

─────────────────────────────────────────────────────────────────────────────
             │  Observed   Bootstrap                         Percentile
inefficiency │     Coef.   Std. Err.      z    P>|z|     [90% Conf. Interval]
─────────────┼───────────────────────────────────────────────────────────────
tebc_vrs_o   │
        regu │ -.9161716    .3876275    -2.36   0.018    -1.547635   -.2845122
        prop │ -1.209787    .5058764    -2.39   0.017    -2.062226   -.4182389
        judi │  .6717764    .3638639     1.85   0.065     .0861624    1.271846
        lpop │ -1.796833    .5587313    -3.22   0.001    -2.725979   -.8933189
c.regu#c.l~p │  .4237473    .1520378     2.79   0.005     .1775733    .6736762
c.prop#c.l~p │  .2335061    .1755503     1.33   0.183    -.0346361      .53319
c.judi#c.l~p │ -.2302095    .1372156    -1.68   0.093      -.45712   -.0115268
       _cons │  7.887194    1.604777     4.91   0.000     5.281076    10.45693
─────────────┼───────────────────────────────────────────────────────────────
      /sigma │  .8735555    .1007678     8.67   0.000     .6807801    1.010888
─────────────────────────────────────────────────────────────────────────────

. estimates store alg_2_raw

. quietly margins, dydx($g_list) post

. estimates store alg_2
```

summarize gives us descriptive statistics for estimated, bias-corrected inefficiency. Comparing them with the descriptives for te_vrs_o shows that the bias correction adjusts the estimated scores away from unity, ruling out (seemingly) fully efficient countries.

```
. summarize tebc_vrs_o

    Variable │        Obs        Mean    Std. Dev.       Min        Max
─────────────┼─────────────────────────────────────────────────────────
  tebc_vrs_o │        131    1.956169    .7083735   1.068883   6.482759
```

To interpret the results qualitatively, we examine the estimated mean marginal effects. This yields a rather clear picture. While, on average, the regulatory burden and judicial independence appear to be immaterial for the efficiency of GDP generation, the protection of property rights matters. Except for tobit, the estimated marginal effect is clearly significant and amounts to roughly 1/3 (Farrell, output-oriented) units, by which inefficiency is reduced in response to a one-unit increase in property rights protection. This appears to be a strong effect that corresponds to a shift from the median to the 27th percentile of the sample distribution of tebc_vrs_o.

```
. estimates table tobit truncreg alg_1 alg_2,
> title(Estimated mean marginal effects) p

Estimated mean marginal effects

--------------------------------------------------------------------
    Variable |    tobit       truncreg       alg_1         alg_2
-------------+------------------------------------------------------
        regu | -.02001409    .04003719     .04003719     .04118827
             |     0.8110       0.6720        0.6607        0.6814
        prop | -.17286701   -.33398801    -.33398801    -.33925269
             |     0.1211       0.0049        0.0042        0.0108
        judi |  .02948397    .08804266     .08804266     .06698891
             |     0.7278       0.3449        0.3325        0.5203
--------------------------------------------------------------------
                                                         legend: b/p
```

Measuring effects in terms of Farrell (output-oriented) efficiency units appears not to be particularly telling. Hence, one may prefer a scaled efficiency measure that allows for interpreting marginal effects in terms of percentage points. Thus, one may switch to the Shephard efficiency measure. Switching from output- to input-oriented efficiency, which would also yield efficiency scores within the unit interval, does not have much appeal for this application. It would imply the thought experiment of reducing input consumption, which appears rather odd given that the national capital stock and human capital are among the input variables.

While switching to the Shephard measure was straightforward for `algorithm(1)`—one just has to use the reciprocal of `te_vrs_o` as the dependent variable—in this application, it causes difficulties with `algorithm(2)`. As indicated by a warning issued by `simarwilson` (see below), the bias correction yields some negative scores; compare section 2.3. These are not used in the truncated regressions. Thus, only 127, not 131, countries enter the regression analysis. In qualitative terms, using the Shephard measure as the left-hand-side variable does not change the general pattern of results. As expected (see footnote 28), the signs of all coefficients are just reversed, and all coefficients remain statistically significant.

```
. simarwilson (rgdpo = ck emp hc) $z_list if year == 2014, algorithm(2)
> rts(vrs) base(output) reps(2000) bcreps(1000) invert
warning: bias-correction yields at least one negative score; consider
  dropping opt. invert or switching to base(input)

Simar & Wilson (2007) eff. analysis      Number of obs             =       127
(algorithm #2)                           Number of efficient DMUs  =         0
                                         Number of bootstr. reps   =      2000
inefficient if bcscore < 1               Wald chi2(7)              =     93.44
twosided truncation                      Prob > chi2(7)            =    0.0000
──────────────────────────────────────────────────────────────────────────────
Data Envelopment Analysis:               Number of DMUs            =       131
                                         Number of ref. DMUs       =       131
output oriented (Shephard)               Number of outputs         =         1
variable returns to scale                Number of inputs          =         3
bias corrected efficiency measure        Number of reps (bc)       =      1000
──────────────────────────────────────────────────────────────────────────────
             │   Observed   Bootstrap                         Percentile
  efficiency │      Coef.   Std. Err.      z    P>|z|     [90% Conf. Interval]
─────────────┼────────────────────────────────────────────────────────────────
bcscore      │
        regu │   .0691237   .0376769     1.83   0.067      .006694    .1316281
        prop │   .2030272    .049161     4.13   0.000      .123413    .2827933
        judi │  -.1056821   .0366765    -2.88   0.004    -.1656055      -.0459
        lpop │   .2506157   .0484025     5.18   0.000     .1726308    .3321083
c.regu#c.l~p │  -.0408009    .013037    -3.13   0.002     -.062014   -.0190109
c.prop#c.l~p │  -.0632933   .0165594    -3.82   0.000    -.0897817   -.0356286
c.judi#c.l~p │   .0520734    .013219     3.94   0.000     .0304517     .074041
       _cons │  -.3667276   .1422998    -2.58   0.010     -.605271   -.1369738
─────────────┼────────────────────────────────────────────────────────────────
      /sigma │   .1190544   .0076651    15.53   0.000     .1025166    .1276049
──────────────────────────────────────────────────────────────────────────────

. quietly margins, dydx($g_list) post

. estimates store alg_2_inv
```

One may force simarwilson to use negative bias-corrected scores in the regression analysis by combining invert with the option notwosided. However, one accepts two inconsistencies by doing this. Besides allowing for negative efficiency scores, which arguably makes little sense, one makes simarwilson apply different truncation rules in different steps of the estimation procedure; see footnote 18. As can be seen from the output below, simarwilson points the user to this issue. Indeed, forcing simarwilson to consider the few observations with negative scores has a noticeable impact on the estimated coefficients.

```
. simarwilson (rgdpo = ck emp hc) $z_list if year == 2014, algorithm(2)
> rts(vrs) base(output) reps(2000) bcreps(1000) invert notwosided
warning: opt. notwosided not recommendable with alg. #2; in step 3.1
  (alg. #2) sampling is from the twosided-truncated normal distribution
warning: bias-correction yields at least one negative score; consider
  dropping opt. invert or switching to base(input)

Simar & Wilson (2007) eff. analysis      Number of obs             =       131
(algorithm #2)                           Number of efficient DMUs  =         0
                                         Number of bootstr. reps   =      2000
inefficient if bcscore < 1               Wald chi2(7)              =     98.54
onesided truncation                      Prob > chi2(7)            =    0.0000
------------------------------------------------------------------------------

Data Envelopment Analysis:               Number of DMUs            =       131
                                         Number of ref. DMUs       =       131
output oriented (Shephard)               Number of outputs         =         1
variable returns to scale                Number of inputs          =         3
bias corrected efficiency measure        Number of reps (bc)       =      1000

------------------------------------------------------------------------------
             |   Observed   Bootstrap                         Percentile
  efficiency |      Coef.   Std. Err.      z    P>|z|     [90% Conf. Interval]
-------------+----------------------------------------------------------------
bcscore      |
        regu |   .0400735   .0429141     0.93   0.350     -.031693    .1113316
        prop |   .2704966   .0544725     4.97   0.000     .1828421    .3623146
        judi |  -.1504577   .0401147    -3.75   0.000     -.216313   -.0863225
        lpop |   .2538996    .056574     4.49   0.000     .1615896    .3499556
c.regu#c.l~p |  -.0301825   .0149123    -2.02   0.043    -.0546047   -.0048266
c.prop#c.l~p |  -.0856331   .0186131    -4.60   0.000    -.1159246   -.0555597
c.judi#c.l~p |   .0696979   .0147094     4.74   0.000     .0452582    .0940363
       _cons |  -.4241697   .1654795    -2.56   0.010    -.7034528   -.1535988
-------------+----------------------------------------------------------------
      /sigma |   .1404817   .0088438    15.88   0.000     .1210069    .1504123
------------------------------------------------------------------------------

. quietly margins, dydx($g_list) post

. estimates store alg_2_notwo
```

To specify a model that renders interpreting estimation results in quantitative terms more convenient, one can use the option `logscore` as a possible alternative to `invert`. By considering log inefficiency as the left-hand-side variable, one can interpret marginal effects as percentage reductions in inefficiency. Hence, we rerun our preferred model (algorithm 2, Farrell output-oriented efficiency) using the option `logscore`. The statistical significance and the signs of the estimated coefficients are equivalent to those from the specification of reference.

```
. simarwilson (rgdpo = ck emp hc) $z_list if year == 2014, algorithm(2)
> rts(vrs) base(output) reps(2000) bcreps(1000) logscore
Simar & Wilson (2007) eff. analysis      Number of obs            =       131
(algorithm #2)                           Number of efficient DMUs =         0
                                         Number of bootstr. reps  =      2000
                                         Wald chi2(7)             =     37.47
inefficient if ln(bcscore) > 0           Prob > chi2(7)           =    0.0000
─────────────────────────────────────────────────────────────────────────────
Data Envelopment Analysis:               Number of DMUs           =       131
                                         Number of ref. DMUs      =       131
output oriented (Farrell)                Number of outputs        =         1
variable returns to scale                Number of inputs         =         3
bias corrected efficiency measure        Number of reps (bc)      =      1000
─────────────────────────────────────────────────────────────────────────────
             │  Observed   Bootstrap                         Percentile
inefficiency │     Coef.   Std. Err.      z    P>|z|     [90% Conf. Interval]
─────────────┼───────────────────────────────────────────────────────────────
ln(bcscore)  │
        regu │ -.2212588    .0942843    -2.35   0.019    -.3813767   -.0688785
        prop │ -.3127128    .1199976    -2.61   0.009    -.5092462   -.1177127
        judi │  .1617935      .08771     1.84   0.065     .0229253    .3082684
        lpop │ -.4737086    .1281714    -3.70   0.000    -.6940146   -.2679593
c.regu#c.l~p │  .0927766    .0340622     2.72   0.006      .038239    .1526358
c.prop#c.l~p │  .0777964    .0414699     1.88   0.061     .0125089    .1486909
c.judi#c.l~p │ -.0644932    .0323854    -1.99   0.046    -.1192701   -.0134126
       _cons │  2.312936    .3710666     6.23   0.000     1.700748     2.91863
─────────────┼───────────────────────────────────────────────────────────────
      /sigma │  .2882745    .0207225    13.91   0.000     .2448331    .3138336
─────────────────────────────────────────────────────────────────────────────
. quietly margins, dydx($g_list) post
. estimates store alg_2_log
```

One may not feel comfortable with using a (bias-corrected) efficiency measure that conflicts with convexity of the production-possibility set; compare figure 1. One way of addressing this issue is to once again envelop the nonconvex bias-corrected estimated frontier with a convex hull and to use the distance to this convexified bias-corrected frontier as a dependent variable in the regression analysis (compare Badunenko, Henderson, and Russell [2013] and figure 5). The (*ref_outputs* = *ref_inputs*) specification of `teradial` allows one to straightforwardly implement this procedure; see the output below and Badunenko and Mozharovskyi (2016). Compared with its direct counterpart (`simarwilson, algorithm(2)` without `invert` and `logscore`), this once more adjusted efficiency measure changes the estimated coefficients markedly. Yet, qualitatively, the pattern of estimates remains the same.

```
. quietly generate rgdpo_front = tebc_vrs_o*rgdpo

. teradial rgdpo = ck emp hc (rgdpo_front = ck emp hc) if year == 2014 &
> regu <. & prop <. & judi <. & lpop <., tename(tebc_vrs_o_convex) rts(vrs)
> base(output) noprint

. simarwilson tebc_vrs_o_convex $z_list if year == 2014, reps(2000)

Simar & Wilson (2007) eff. analysis      Number of obs             =       131
(algorithm #1)                           Number of efficient DMUs  =         0
                                         Number of bootstr. reps   =      2000
                                         Wald chi2(7)              =     24.89
inefficient if tebc_vrs_o_convex > 1     Prob > chi2(7)            =    0.0008
------------------------------------------------------------------------------
Data Envelopment Analysis:                          externally estimated scores
------------------------------------------------------------------------------
             |   Observed   Bootstrap                         Percentile
inefficiency |      Coef.   Std. Err.      z    P>|z|     [90% Conf. Interval]
-------------+----------------------------------------------------------------
tebc_vrs_o~x |
        regu |  -.8525049     .351338    -2.43   0.015    -1.436492   -.2731832
        prop |  -1.104847    .4583591    -2.41   0.016    -1.884513   -.3711119
        judi |   .5759623    .3329715     1.73   0.084     .0401372    1.122519
        lpop |  -1.748744    .5217818    -3.35   0.001    -2.625194    -.919087
c.regu#c.l~p |   .3752556    .1349932     2.78   0.005     .1569001    .5928115
c.prop#c.l~p |   .2194959    .1578936     1.39   0.164    -.0314325    .4908592
c.judi#c.l~p |  -.1856799    .1233833    -1.50   0.132    -.3866211    .0144541
       _cons |   7.956469    1.475931     5.39   0.000     5.576847      10.362
-------------+----------------------------------------------------------------
      /sigma |   .8986468    .0911453     9.86   0.000     .7240245    1.017877
------------------------------------------------------------------------------

. quietly margins, dydx($g_list) post

. estimates store bc_convex
```

Finally, we compare the marginal effects for all specifications of `simarwilson` that we have estimated. Somewhat surprisingly, unlike the specification of reference, the specifications using the Shephard measure argue for more regulatory interference for improving efficiency (p-values 0.039 and 0.096, respectively). One may hence speculate that `regu` captures both detrimental and beneficial facets of business regulation. In terms of the point estimates, all model specifications yield a positive association of property right protection and GDP efficiency. Only for the Shephard measure as the left-hand-side variable (without option `notwosided`) does the average marginal effect of `prop` turn statistically insignificant at the 10% level.

Using the Shephard measure (option `invert`) or the option `logscore` makes interpreting the estimated marginal effect easier. According to the specification using the Shephard measure (without `notwosided`), a one-unit increase in `prop` on average improves efficiency by 3.6 percentage points. According to the specification using log inefficiency at the left-hand side, the mean effect is a 10.7% reduction in inefficiency. With respect to `judi`, all estimated marginal effects are statistically insignificant. In terms of estimated average marginal effects, basing the analysis on a convex estimated hull has almost no effect compared with using the nonconvex, bias-corrected estimated frontier.

```
. estimates table alg_1 alg_2 alg_2_inv alg_2_notwo alg_2_log bc_convex,
> title(Estimated mean marginal effects) p b(%5.4f) p(%4.3f)
Estimated mean marginal effects
──────────────────────────────────────────────────────────────────────────────
    Variable │   alg_1      alg_2     alg_2~v    alg_2~o    alg_2~g    bc_co~x
─────────────┼────────────────────────────────────────────────────────────────
        regu │   0.0400     0.0412    -0.0378    -0.0358     0.0074     0.0210
             │    0.661      0.681      0.039      0.096      0.861      0.848
        prop │  -0.3340    -0.3393     0.0359     0.0532    -0.1067    -0.3541
             │    0.004      0.011      0.139      0.063      0.056      0.015
        judi │   0.0880     0.0670     0.0310     0.0256     0.0020     0.0828
             │    0.332      0.520      0.104      0.253      0.964      0.455
──────────────────────────────────────────────────────────────────────────────
                                                                  legend: b/p
```

4.2 Effect heterogeneity

We complete our application by analyzing possible heterogeneity in the efficiency effects of burden of government regulation, property rights protection, and judicial independence. In doing this, we focus on `simarwilson, algorithm(2)` without `invert` and `logscore` as our preferred estimation method. The estimated mean marginal effects from this model suggest that only the protection of property rights matters for efficiency. However, this result might just be an artifact of averaging heterogeneous effects. We graphically examine possible effect heterogeneity using the `marginsplot` command; see [R] **marginsplot** and the output below. We consider two dimensions of heterogeneity: heterogeneity with respect to country size measured by `lpop` (figure 2, right panel) and heterogeneity with respect to the respective considered dimension of governance quality (figure 2, left panel).

```
. estimates restore alg_2_raw
(results alg_2_raw are active now)
. local h_list "$g_list lpop"
. foreach h of varlist `h_list´ {
  2.         quietly sum `h´ if e(sample)
  3.         local mymin = r(min)*0.98
  4.         local myxmin = ceil(`mymin´)
  5.         local mymax = r(max)*1.02
  6.         local myxmax = floor(`mymax´)
  7.         local mystep = (`mymax´-`mymin´)/25
  8.         foreach g of varlist `h_list´ {
  9.                 local r_list : list h_list - h
 10.                 quietly margins if e(sample), dydx(`g´)
> at(`h´=(`mymin´(`mystep´)`mymax´) (asobserved) `r_list´)
 11.                 quietly marginsplot, xlabel(`myxmin´(1)`myxmax´)
> recast(line) recastci(rarea) scheme(s2manual)
 12.                 quietly graph export "simarwilson2`g´_`h´.eps", as(eps)
> preview(off) replace fontface(Times)
 13.         }
 14. }
```

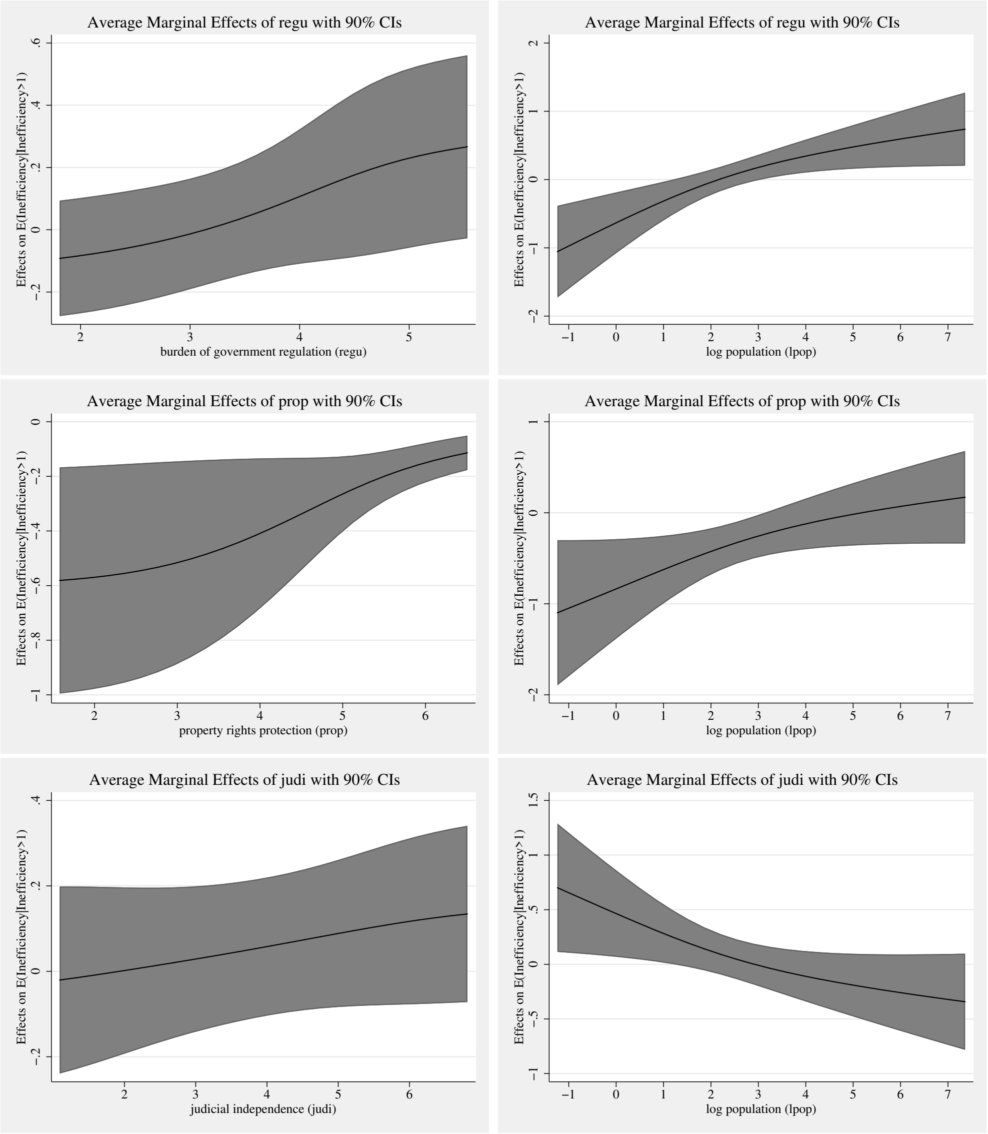

Figure 2. Estimated marginal effects of governance-quality indices on inefficiency by country size (right panel) and its respective own value (left panel). NOTE: Farrell output-oriented efficiency as dependent variable; Simar and Wilson (2007), algorithm 2 used for estimation; 90% confidence bands indicated by shaded areas. SOURCE: Calculations are our own, based on Penn World Table and World Economic Forum Global Competitiveness Report data.

The left panel of figure 2 does not suggest that the effect heterogeneity with respect to the respective category of governance quality is a big issue, at least qualitatively.

The effects of both the burden of government regulation and judicial independence on inefficiency are statistically insignificant at any level of `regu` and `judi`. This is perfectly in line with the small and statistically insignificant estimated mean effects. Yet, if one focuses on the point estimates of the marginal effect of `regu` and for a moment ignores statistical significance, then figure 2 points to relaxing government regulation being beneficial if the regulator burden is high but exerting a negative effect on efficiency if it is already small. This pattern arguably makes sense. The pattern for property rights protection does also not conflict with what we found for the mean effect. Here we find a significant inefficiency-reducing effect of better property rights protection over the entire range of `prop`. Yet the effect seems to be much stronger for low levels of property rights protection, though the estimated marginal effect gets increasingly noisy for small values of `prop`.

The overall picture is somewhat different for heterogeneity with respect to country size (figure 2, right panel). There the marginal effect of all three governance indicators exhibits substantial heterogeneity. While focusing on mean marginal effects suggested that the level of regulation was immaterial for national efficiency, considering effect heterogeneity challenges this finding. More specifically, figure 2 suggests that relaxing government regulation reduces inefficiency in small countries. Yet in big countries, it seems to exert a negative effect on national efficiency. This pattern corroborates our earlier hypothesis of regulatory burden being an ambiguous concept because in certain circumstances some regulation may be well required for efficient production. A similar pattern of heterogeneity is found for the effect of `prop`. In small countries, improving the protection of property rights is clearly beneficial for efficiency of GDP production, while for big countries, such an effect is not found in the data. The reverse pattern of heterogeneity is found with respect to judicial independence. While the effect of `judi` on efficiency is statistically insignificant for many values of `lpop`, figure 2 suggests a statistically significant, efficiency-reducing effect for very small countries. However, this somewhat surprising finding has to be interpreted with caution. Near collinearity might be a technical explanation for the mirror-inverted patterns found for `prop` and `judi`. Both variables are strongly correlated (0.903) in the estimation sample, while their respective correlations with `regu` (0.506 and 0.448) are much weaker.

4.3 The gciget command

As mentioned in section 4.1, importing the indices from the Global Competitiveness Report that we used in our empirical study is not straightforward. We have developed the new command `gciget` to get the indices directly into the memory of Stata from the World Economic Forum's Global Competitiveness Report.

`gciget` has three steps. First, it downloads the file `GCI_Dataset_2007-2017.xlsx` from the The Global Competitiveness Report section (http://reports.weforum.org/global-competitiveness-index-2017-2018/) of the World Economic Forum website. The user can optionally indicate the path to the Excel file `GCI_Dataset_2007-2017.xlsx` stored locally. Second, `gciget` imports the Excel file; see [D] **import excel** regarding the requirement for the version of Stata. Third, `gciget` processes the variables the user has specified after `gciget`. The resulting data are in a long format and are by default declared to be panel data; see [XT] **xtset**.

The syntax for `gciget` reads as follows:

`gciget` [*varlist*] [`, clear noxtset noquery panelvar(`*newvar*`) url(`*filename*`)`
`sheet("`*sheetname*`") cellrange(`[*start*] [`:` *end*]`) nowarnings`]

The user can optionally specify the *varlist* from the list of indices in the Global Competitiveness Report (see the Excel file `GCI_Dataset_2007-2017.xlsx` for the possible names). If no valid name of the index is specified, all indices will be processed.

The following options are available:

`clear` clears data in memory before loading the GCI data.

`noxtset` prevents loaded data from being declared panel data (see [XT] **xtset**). Without `noxtset`, `gciget` generates a new numeric *panelvar*, which is a numeric equivalent to the string variable `countrycode`. `year` serves as *timevar*.

`noquery` suppresses summary calculations by `xtset`. With `noxtset`, `noquery` has no effect.

`panelvar(`*newvar*`)` specifies the name of the numeric *panelvar* that is generated by `gciget`. The default is `panelvar(cnNumber)`. With `noxtset`, `panelvar()` has no effect.

`url(`*filename*`)` specifies the URL of the GCI data. The default is `url(http://www3.weforum.org/docs/GCR2017-2018/GCI_Dataset_2007-2017.xlsx)`. Specifying `url(`*filename*`)` might become necessary if a new version of the GCI is published and the name of the data file changes or if the web address changes for any other reason. `url(`*filename*`)` may also be used to specify the path to the GCI data, if the Excel file `GCI_Dataset_2007-2017.xlsx` has already been downloaded and locally stored.

`sheet("`*sheetname*`")` specifies the Excel worksheet to load. The default is `sheet("Data")`. Specifying `sheet("`*sheetname*`")` might become necessary if a new version of the GCI is published and the name of the relevant Excel worksheet changes. *sheetname* is passed through to `import excel`.

`cellrange(`[*start*] [`:` *end*]`)` specifies a range of cells within the Excel sheet to load. The default is `cellrange(A3:FM6527)`. Specifying `cellrange(`[*start*] [`:` *end*]`)` might become necessary if a new version of the GCI is published and the structure of the Excel file changes. *start*`:`*end* is passed through to `import excel`.

`nowarnings` prevents `gciget` from displaying warnings that are routinely issued if the options `url(`*filename*`)`, `sheet("`*sheetname*`")`, and `cellrange(`[*start*] [`:` *end*]`)` deviate from the respective default.

gciget only helps the user get the data from the World Economic Forum into Stata. Thus, any liability for the data or their usage is disclaimed. That the data come from the World Economic Forum also puts restriction on the data availability and the terms and conditions under which the data can be used. As of this writing, the data are available for 2007–2017. The following code illustrates a simple import of four indices and plotting GCI for four countries.

```
. gciget EOSQ048 EOSQ051 GCI GCI.A.02.01, clear
DISCLAIMER: The World Economic Forum is the provider of the Global
  Competitiveness Index 2017-2018, a framework and a corresponding set of
  indicators for 137 economies. The software gciget.ado provides a practical
  way to read the indicators into Stata (R). The responsibility of complying
  with the terms and conditions of use under which the owner of the data
  grants access to the indicators is entirely with the user but not with the
  authors of the software gciget.ado. Any user of gciget.ado is responsible
  for making him or herself familiar with the terms of use under which she or
  he is allowed to work with the data of the Global Competitiveness Index. For
  more information and methodology, please see http://wef.ch/gcr17. In no
  event will the authors, owners, and creators of gciget.ado, or their
  employers or any other party who may modify and/or redistribute this
  software, accept liability for any loss or damage suffered as a result of
  using the gciget.ado software.
Downloading the GCI_Dataset_2007-2017.xlsx file
Importing the GCI_Dataset_2007-2017.xlsx file
Processing EOSQ048:  1.09 Burden of government regulation, 1-7 (best)
Processing EOSQ051:  1.01 Property rights, 1-7 (best)
Processing GCI:  Global Competitiveness Index
Processing GCI_A_02_01:  A. Transport infrastructure
. xtline GCI if countrycode == "USA" | countrycode == "DEU" |
> countrycode == "FRA" | countrycode == "GBR", overlay i(country)
> t(year) scheme(sj) xlabel(2007(2)2017)
. quietly graph export "GCI_four_cns.eps", as(eps) preview(off) replace
> fontface(Times)
```

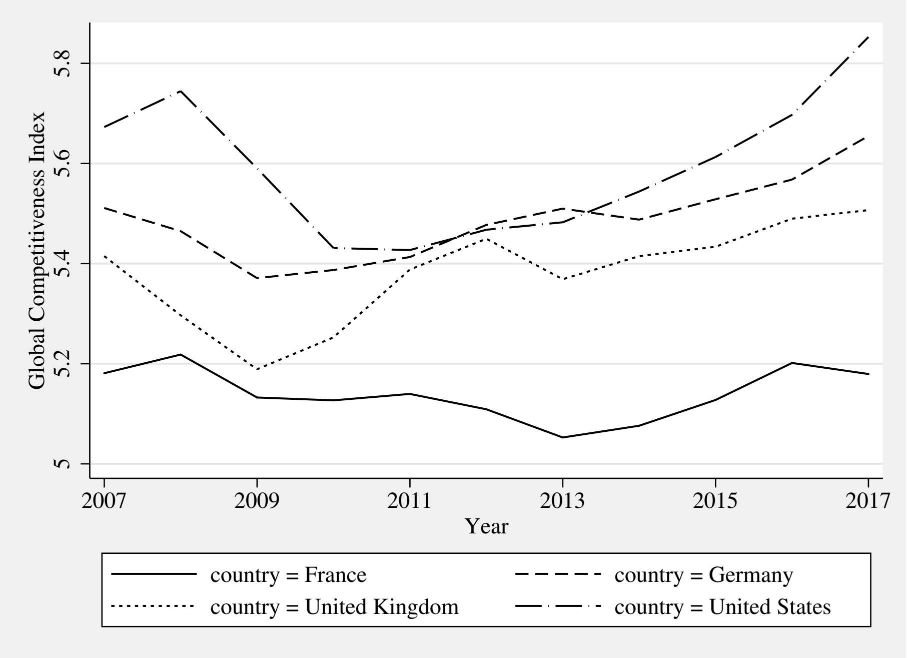

Figure 3. GCI for France, Germany, the UK, and the United States. *Source*: World Economic Forum's Global Competitiveness Report.

5 Summary and conclusions

In this article, we introduced the new community-contributed command `simarwilson`, which implements Simar and Wilson (2007) two-stage efficiency analysis. This estimator has substantial value for applied efficiency analysis because it puts regression analysis of DEA scores on firm statistical ground. The new command extends the originally proposed procedure in some respects, which increases its applicability in applied empirical work. `simarwilson` complements the contributions of Ji and Lee (2010), Tauchmann (2012), and, particularly, Badunenko and Mozharovskyi (2016), who have already made related methods of nonparametric efficiency analysis available to Stata users.

6 Acknowledgments

This work has been supported in part by the Collaborative Research Center "Statistical Modelling of Nonlinear Dynamic Processes" (SFB 823) of the German Research Foundation. The authors are grateful to Ramon Christen, Rita Maria Ribeiro Bastiao, Akash Issar, Ana Claudia Sant'Anna, Jarmila Curtiss, Meir José Behar Mayerstain, Erik Alda, Annika Herr, Hendrik Schmitz, Franziska Valder, Franz Josef Zorzi, Irina Simankova, Christian Merkl, Howard J. Newton, the participants of the 2015 German Stata Users Group meeting, and one anonymous reviewer for many valuable comments.

7 Supplementary figures

The figures below graphically illustrate the concepts of a restricted reference set (figure 4) and a convexified frontier (figure 5) that were referred to in this article, using the same artificial data that were used to illustrate DEA in figure 1.

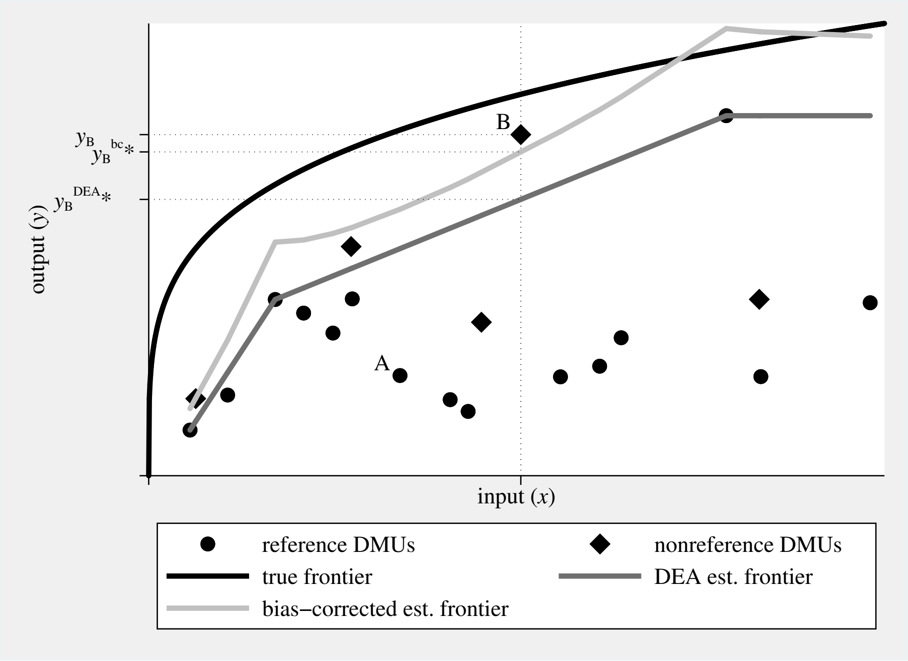

Figure 4. Estimated inefficiency for subsample of DMUs used as reference. Considering only a subsample of DMUs as reference set renders DMU B seemingly superefficient, both according to conventional DEA ($\widehat{\theta_{\mathrm{B}}} = (y_{\mathrm{B}}^{\mathrm{DEA}*}/y_{\mathrm{B}}) < 1$) and according to bias-corrected DEA ($\widehat{\theta_{\mathrm{B}}^{\mathrm{bc}}} = (y_{\mathrm{B}}^{\mathrm{bc}*}/y_{\mathrm{B}}) < 1$). DMU A is still estimated to be inefficient; compare figure 1. Yet the magnitude of estimated inefficiency is somewhat smaller. NOTE: Artificial data generated in the same way as for figure 1. SOURCE: Calculations are our own.

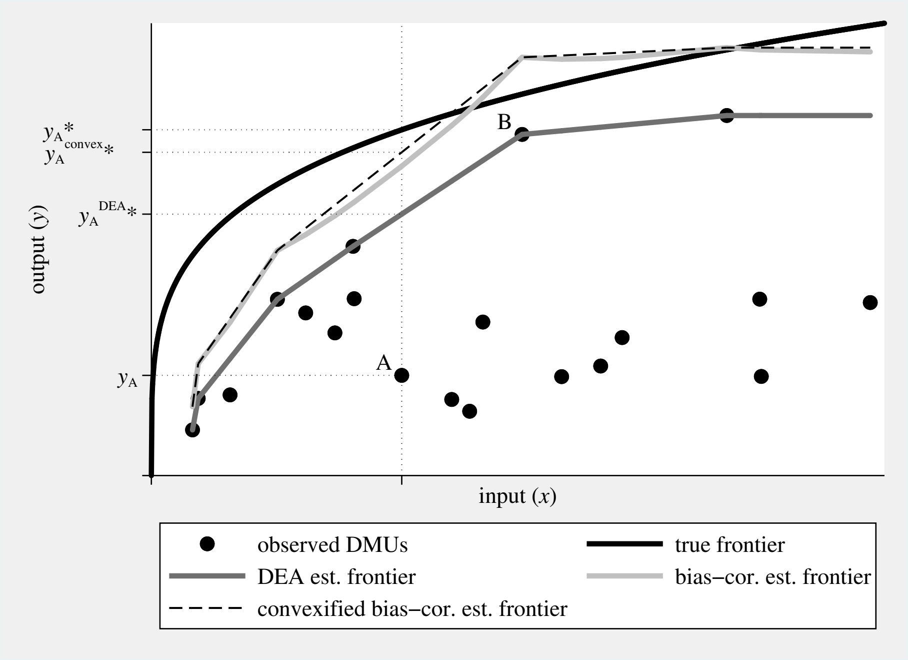

Figure 5. Convexified bias-corrected estimated frontier. Measuring inefficiency relative to the convexified bias-corrected frontier either does not affect estimated bias-corrected inefficiency (for example, DMU B) or increases estimated bias-corrected inefficiency (for example, DMU A). NOTE: Artificial data generated the same as figure 1. SOURCE: Calculations are our own.

8 Programs and supplemental materials

To install a snapshot of the corresponding software files as they existed at the time of publication of this article, type

```
. net sj 19-4
. net install st0585      (to install program files, if available)
. net get st0585          (to install ancillary files, if available)
```

9 References

Aigner, D. J., C. A. K. Lovell, and P. Schmidt. 1977. Formulation and estimation of stochastic frontier production function models. *Journal of Econometrics* 6: 21–37.

Badunenko, O., D. J. Henderson, and R. R. Russell. 2013. Polarization of the worldwide distribution of productivity. *Journal of Productivity Analysis* 40: 153–171.

Badunenko, O., and P. Mozharovskyi. 2016. Nonparametric frontier analysis using Stata. *Stata Journal* 16: 550–589.

Banker, R. D., and R. Natarajan. 2008. Evaluating contextual variables affecting productivity using data envelopment analysis. *Operations Research* 56: 48–58.

Charnes, A., W. W. Cooper, and E. Rhodes. 1978. Measuring the efficiency of decision making units. *European Journal of Operational Research* 2: 429–444.

Chopin, N. 2011. Fast simulation of truncated Gaussian distributions. *Statistics and Computing* 21: 275–288.

Chortareas, G. E., C. Girardone, and A. Ventouri. 2013. Financial freedom and bank efficiency: Evidence from the European Union. *Journal of Banking & Finance* 37: 1223–1231.

Coelli, T. J., D. S. P. Rao, C. J. O'Donnell, and G. E. Battese. 2005. *An Introduction to Efficiency and Productivity Analysis*. 2nd ed. New York: Springer.

Cooper, W. W., L. M. Seiford, and K. Tone. 2007. *Data Envelopment Analysis: A Comprehensive Text with Models, Applications, References and DEA-Solver Software*. 2nd ed. New York: Springer.

Daraio, C., and L. Simar. 2005. Introducing environmental variables in nonparametric frontier models: A probabilistic approach. *Journal of Productivity Analysis* 24: 93–121.

———. 2007. *Advanced Robust and Nonparametric Methods in Efficiency Analysis: Methodology and Applications*. New York: Springer.

Emrouznejad, A., and G. Yang. 2018. A survey and analysis of the first 40 years of scholarly literature in DEA: 1978–2016. *Socio-Economic Planning Sciences* 61: 4–8.

Farrell, M. J. 1957. The measurement of productive efficiency. *Journal of the Royal Statistical Society, Series A* 120: 253–290.

Feenstra, R. C., R. Inklaar, and M. P. Timmer. 2015. The next generation of the Penn World Table. *American Economic Review* 105: 3150–3182.

Fragkiadakis, G., M. Doumpos, C. Zopounidis, and C. Germain. 2016. Operational and economic efficiency analysis of public hospitals in Greece. *Annals of Operations Research* 247: 787–806.

Glass, A. J., K. Kenjegalieva, and J. Taylor. 2015. Game, set and match: Evaluating the efficiency of male professional tennis players. *Journal of Productivity Analysis* 43: 119–131.

Hjalmarsson, L., S. C. Kumbhakar, and A. Heshmati. 1996. DEA, DFA and SFA: A comparison. *Journal of Productivity Analysis* 7: 303–327.

Hoff, A. 2007. Second stage DEA: Comparison of approaches for modelling the DEA score. *European Journal of Operational Research* 181: 425–435.

Ji, Y.-B., and C. Lee. 2010. Data envelopment analysis. *Stata Journal* 10: 267–280.

Kneip, A., L. Simar, and P. W. Wilson. 2008. Asymptotics and consistent bootstraps for DEA estimators in nonparametric frontier models. *Econometric Theory* 24: 1663–1697.

Koopmans, T. C. 1951. An analysis of production as an efficient combination of activities. In *Activity Analysis of Production and Allocation*, ed. T. C. Koopmans, 33–97. New York: Wiley.

McDonald, J. 2009. Using least squares and tobit in second stage DEA efficiency analyses. *European Journal of Operational Research* 197: 792–798.

Murillo-Zamorano, L. R. 2004. Economic efficiency and frontier techniques. *Journal of Economic Surveys* 18: 33–77.

Pérez Urdiales, M., A. O. Lansink, and A. Wall. 2016. Eco-efficiency among dairy farmers: The importance of socio-economic characteristics and farmer attitudes. *Environmental & Resource Economics* 64: 559–574.

Ramalho, E. A., J. J. S. Ramalho, and P. D. Henriques. 2010. Fractional regression models for second stage DEA efficiency analyses. *Journal of Productivity Analysis* 34: 239–255.

Schwab, K., ed. 2017. *The Global Competitiveness Report 2017–2018.* Geneva: World Economic Forum.

Shephard, R. W. 1970. *Theory of Cost and Production Functions.* Princeton, NJ: Princeton University Press.

Simar, L., and P. W. Wilson. 2000. A general methodology for bootstrapping in nonparametric frontier models. *Journal of Applied Statistics* 27: 779–802.

———. 2007. Estimation and inference in two-stage, semi-parametric models of production processes. *Journal of Econometrics* 136: 31–64.

———. 2011. Two-stage DEA: *Caveat emptor. Journal of Productivity Analysis* 36: 205–218.

Tauchmann, H. 2012. Partial frontier efficiency analysis. *Stata Journal* 12: 461–478.

World Economic Forum. 2018. The global competitiveness index dataset 2007–2018. http://www3.weforum.org/docs/GCR2017-2018/GCI_Dataset_2007-2017.xlsx.

About the authors

Oleg Badunenko is a senior lecturer in economics at the University of Portsmouth. His major research interests are efficiency and productivity analysis, as well as microeconometrics and nonparametric econometrics.

Harald Tauchmann is a professor of health economics at the Friedrich-Alexander-Universität Erlangen-Nürnberg; guest researcher at Rheinisch-Westfälisches Institut für Wirtschaftsforschung (RWI), Essen, Germany; and research fellow at Centre of Health Economics Research (CINCH), Essen, Germany. His research interests include health economics, applied econometrics, and statistical methods.