Badunenko, Oleg; Tauchmann, Harald

Working Paper

Simar and Wilson two-stage efficiency analysis for Stata

FAU Discussion Papers in Economics, No. 08/2018

Provided in Cooperation with:
Friedrich-Alexander University Erlangen-Nuremberg, Institute for Economics

Suggested Citation: Badunenko, Oleg; Tauchmann, Harald (2018) : Simar and Wilson two-stage efficiency analysis for Stata, FAU Discussion Papers in Economics, No. 08/2018, Friedrich-Alexander-Universität Erlangen-Nürnberg, Institute for Economics, Erlangen

This Version is available at:
http://hdl.handle.net/10419/179503

Terms of use:
Documents in EconStor may be saved and copied for your personal and scholarly purposes.
You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.
If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.
Simar and Wilson two-stage efficiency analysis for Stata

Oleg Badunenko
University of Portsmouth

Harald Tauchmann
University of Erlangen-Nürnberg,
RWI - Leibniz Institut für Wirtschaftsforschung,
and CINCH - Health Economics Research Center

ISSN 1867-6707
Simar and Wilson two-stage efficiency analysis for Stata

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

Harald Tauchmann
Universität Erlangen-Nürnberg,
and RWI – Leibniz Institut für Wirtschaftsforschung,
and CINCH – Health Economics Research Center
Findelgasse 7/9, 90402 Nürnberg, Germany
harald.tauchmann@fau.de

May, 2018

Abstract. When analyzing what determines the efficiency of production, regressing efficiency scores estimated by DEA on explanatory variables has much intuitive appeal. Simar and Wilson (2007) show that this naïve two-stage estimation procedure suffers from severe flaws, that render its results, and in particular statistical inference based on them, questionable. At the same time they propose a statistically grounded bootstrap based two-stage estimator that eliminates the above mentioned weaknesses of its naïve predecessors and comes in two variants. This article introduces the new Stata command simarwilson that 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 the new command 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 of a country for its efficiency of output production.

Keywords: DEA, two-stage estimation, truncated regression, bootstrap, efficiency, bias correction, environmental variables.
1 Introduction

Analyzing the technical efficiency of production/decision making units (DMUs) has developed into a major field in empirical economics and management science. From a methodological perspective, two major strands of the literature can be distinguished: (i) analyses that rest on parametric, regression based methods, namely stochastic frontier analysis (SFA, Aigner et al. 1977), and (ii) analyses that use non-parametric methods, namely data envelopment analysis (DEA, Charnes et al. 1978). The pros and cons of either approach have been discussed extensively (e.g. Hjalmarsson et al. 1996; Murillo-Zamorano 2004).

One of the advantages of the parametric approaches, namely the truncated-normal stochastic frontier model, is that it does not only allow for measuring inefficiency but also incorporates a model of the determinants of inefficiency. In contrast, non-parametric 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. Yet, shedding light on what determines the magnitude of this distance is out of the narrow scope of non-parametric approaches such as DEA.

For many research questions, however, identifying determinants of inefficiency is of much greater relevance than determining its magnitude for specific DMUs. For this reason, in the domain of non-parametric efficiency analysis, semi-parametric 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 fifty published articles and mention hundreds of unpublished papers that employ such two-stage procedures. In these (early) applications the second stage is typically a censored (tobit like) regression to account for the bounded nature of DEA efficiency scores, or just simply OLS (Simar and Wilson 2007).

Despite their popularity and their intuitive appeal, such naïve two-stage estimators are criticized by Simar and Wilson (2007) mainly for two reasons. Firstly, they stress the absence of a clear theory of the underlying data generating process, that would justify the naïve two-stage approach. Secondly, they criticize the conventional inference.
that is pursued in most of the two-stage applications, for ignoring that estimated DEA efficiency scores are calculated from a common sample of data. Treating them as if they were independent observations is not appropriate since the problems related to invalid inference due to serial correlation arise. Simar and Wilson (2007) develop a two-stage procedure that takes the above mentioned issues into account. They construct an underlying data generating process that is consistent with a two-stage estimation procedure, which – as the most obvious difference to the earlier naïve 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 develop a parametric bootstrap procedure that is consistent with the assumed data generating process and addresses the second issue. It yields 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. This popular, yet technically involved estimator has not yet been available to Stata users, unless they developed their own code. The present paper introduces the new Stata command simarwilson that allows for applying this estimator in Stata. In doing this, it greatly benefits from the recently published user written Stata 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.

The remainder of this paper is organized as follows. The following section gives a brief summary of the Simar and Wilson (2007) two-stage estimator. The syntax of simarwilson is described in section 3. 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 \(x_i, y_i,\) and \(z_i,\) for a sample of \(i = 1, \ldots, N\) DMUs. \(x_i\) denotes a vector of \(P\) in statistical theory. With respect to the latter they argue that the claimed desirable statistical properties rely on very restrictive assumptions.

5. Google Scholar (2018, March 27) lists more than 2200 citations (for instance Fragkiadakis et al. 2016; Pérez Urdiales et al. 2016; Glass et al. 2015; Chortareas et al. 2013, just to mention few recent applications). Interestingly, the inventors of this popular method dissociate themselves from advocating two-stage approaches (Simar and Wilson 2011, p. 216).

6. An earlier, less powerful version of the ado-file that accompanies this article has been made available through ssc 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.
inputs to production. $y_i$ is a vector of $Q$ outputs from production. $z_i$ 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 $z_i$ on efficiency 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 $(y_i^*, x_i^*)$ that are fully efficient in the sense that no output can be increased without decreasing at least one other output or increasing at least one input. A crucial assumption is that the shape of the production-possibility frontier does not depend on $z_i$, which is referred to as separability in Simar and Wilson (2007).

The output-input set $(y_i, x_i)$ observed for DMU $i$, will regularly fail in realizing a point at the frontier. This deviation is necessarily directional, i.e. $i$ 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 $\theta_i$. That is $\theta_i$ denotes the factor by which output generation $y_i$ of DMU $i$ has to be proportionally increased in order to project $(y_i, x_i)$ onto the frontier. $\theta_i$ is hence a measure of inefficiency that is bounded to the $[0,1]$ interval. Alternatively, one may measure the Farrell distance in input direction as $\psi_i$, that is the factor by which input consumption $x_i$ of DMU $i$ has to be proportionally reduced in order to project $(y_i, x_i)$ onto the frontier. $\psi_i$ is hence a measure of efficiency that is bounded to the $(0,1]$ interval. Yet, in Simar and Wilson (2007) the focus is on $\theta_i$.

The key idea in Simar and Wilson (2007) about the data generating process is that efficiency $\theta_i$ linearly depends on $z_i$:

$$\theta_i = z_i \beta + \varepsilon_i$$

where $\beta$ denotes a column vector of coefficients, estimating which is the ultimate objective of the empirical analysis. The disturbances $\varepsilon_i$ are assumed to be – conditionally on $z_i$ – independently, truncated normally distributed, with parameters $\mu = 0$ and $\sigma$, and left-truncation at $1 - z_i \beta$. This assumption guarantees that $\theta_i$ cannot be smaller than unity, irrespective of the values the variables in $z_i$ may take. Though full efficiency ($\theta_i = 1$) is in principle possible, it occurs with zero probability. Conditional on $\theta_i$, DMU $i$ chooses a set of outputs and inputs $(y_i, x_i)$ as $(1/\theta_i y_i^*, x_i^*)$, with $(y_i^*, x_i^*)$ denoting some point at the production-possibility frontier.

8. An alternative to the Farrell (1957) distance measure is one introduced by Shephard (1970), which is just the reciprocal of $\theta_i$ and $\psi_i$, respectively. simarwilson accommodates the Shephard measure by the option invert; see sections 2.3 and 3.
9. Simar and Wilson (2007) do not explicitly consider input oriented efficiency, i.e. $\psi_i$. 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- or other types of clustered data.
11. The first and the second moment of the conditional error distribution are $E(\varepsilon_i | z_i) = \sigma \lambda_i$ and $\text{Var}(\varepsilon_i | z_i) = \sigma^2 (1 + (1-z_i \beta)/\sigma)(\lambda_i - \lambda_i^2)$, with $\lambda_i$ denoting the inverse Mills ratio $\frac{\phi((1-z_i \beta)/\sigma)}{1-\Phi((1-z_i \beta)/\sigma)}$.
12. The data generating process can hence be described as sampling from the joint distribution $f(x, y, z)$. 

\[begin{align*}
\theta_i &= z_i \beta + \varepsilon_i \\
\text{Var}(\varepsilon_i | z_i) &= \sigma^2 (1 + (1-z_i \beta)/\sigma)(\lambda_i - \lambda_i^2), \\
\lambda_i &= \frac{\phi((1-z_i \beta)/\sigma)}{1-\Phi((1-z_i \beta)/\sigma)}.
\end{align*}\]
It is key for understanding the shortcomings of na"ıve two-stage approaches that $\theta_i$ is genuinely unobservable. In consequence the estimated efficiency score $\hat{\theta}_i$ one obtains from running a DEA is not $\theta_i$. In other words, $\hat{\theta}_i$ is not the distance of $(y_i, x_i)$ to the true production-possibility frontier but the distances to an estimate of the latter. Due to the boundary estimation framework of DEA, this estimate suffers from finite sample bias and in turn $\hat{\theta}_i$ is biased towards the value of one. That means that (1) cannot be estimated straightforwardly and $\theta_i$ has to be replaced in (1) by the biased estimate $\hat{\theta}_i$ in order to formulate an operational regression equation. As pointed out in Simar and Wilson (2007), this generates two major problems for na"ıve two step approaches. Firstly, although the errors $\varepsilon_i$ are assumed to be statistically independent across DMUs, the operational errors in a regression of $\hat{\theta}_i$ on $z_i$ are not, since the $\hat{\theta}_i$ are estimated from a common sample of data. Secondly, in any application of DEA some – usually numerous – $\hat{\theta}_i$ take the value of one, though according to (1) $\theta_i$ takes this value with zero probability.

In the procedure suggested in Simar and Wilson (2007), the former issue is addressed by estimating standard errors and confidence intervals for $\hat{\beta}$ by the means of a parametric bootstrap procedure, in which artificial pseudo errors are independently drawn from the truncated normal distribution with left-truncation at $1 - \hat{\varepsilon}_i\hat{\beta}$. The latter issue is addressed in Simar and Wilson (2007) in two different 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 $\hat{\theta}_i$ that equal one. 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 left-hand-side variable. Estimating this model yields $\hat{\beta}$, which, together with the estimate for the variance parameter $\hat{\sigma}$, enters the bootstrap procedure mentioned above. The second suggested approach (algorithm #2) is more involved and rests on bias corrected DEA scores $\hat{\theta}_i^{bc}$ as left-hand-side variable. Since $\hat{\theta}_i^{bc} > 1$ holds for $i = 1 \ldots N$, 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 process that generates $\theta_i$, i.e. equation (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 et al. 2008) and have recently been made available to Stata users by the user written command teradialbc (Badunenko and Mozharovskyi 2016).

Figure 1 graphically illustrates the concepts of true, DEA estimated, and bias which can be written as $f(x, y|\theta, z) \cdot f(\theta|z) \cdot f(z)$. (Simar and Wilson 2007, p. 35). The focus of the empirical analysis is on $f(\theta|z)$.

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

14. The present article focuses on the intuition behind the Simar and Wilson (2007) estimator, and its practical implementation in Stata. For this reason, its statistical properties are not discussed 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: **Graphical illustration of true and estimated inefficiency.** Considering DMU A, true inefficiency $\theta_A$ is $y_A^*/y_A$, (uncorrected) DEA estimated inefficiency $\hat{\theta}_A$ is $y_{A\text{DEA}}^*/y_A$, and bias corrected estimated inefficiency $\hat{\theta}_{Abc}$ is $y_{A\text{bc}}^*/y_A$. In this finite and small ($N = 20$) artificial sample, DEA systematically underestimates true inefficiency. Bias correction adjusts estimated inefficiency upwards. 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 $\hat{\theta}_{Bbc}$. Unlike for conventional DEA, with bias correction the estimated production-possibility set is not convex, the estimated frontier is not even monotone. Notes: 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 $\beta = 0$ 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: Own calculations.

2.2 **The procedures suggested in Simar and Wilson (2007)**

This subsection in detail describes the suggested procedures algorithm #1 and algorithm #2, mentioned above. In doing this, it almost one-to-one reproduces what is found at pages 41-43 in Simar and Wilson (2007). This, in particular, applies to the subsequent
paragraphs that describe the steps of the estimation procedure(s) in almost exactly the same way as they are described in the key reference.

**Algorithm #1** consists of the following steps:

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

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

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

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

   3.2 Calculate artificial efficiency scores $\tilde{\theta}_i$ as $z_i \hat{\beta} + \tilde{\varepsilon}_i$ for each DMU $i = 1, \ldots, M$.

   3.3 Run a truncated regression (left-truncation at 1) of $\tilde{\theta}_i$ on $z_i$ to obtain maximum likelihood, bootstrap estimates $\hat{\beta}^b$ and $\hat{\sigma}^b$.

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

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

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

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

3. Loop over the following steps 3.1–3.4 $B_1$ times, in order to obtain a set of $B_1$ bootstrap estimates $\hat{\theta}^b_i$ 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 $\tilde{\varepsilon}_i$ from the truncated $N(0, \hat{\sigma})$ distribution with left-truncation at $1 - z_i \hat{\beta}$.

   3.2 Calculate artificial efficiency scores $\tilde{\theta}_i$ as $z_i \hat{\beta} + \tilde{\varepsilon}_i$ for each DMU $i = 1, \ldots, N$.

   3.3 Generate $i = 1, \ldots, N$ artificial DMUs with input quantities $\tilde{x}_i = x_i$ and output quantities $\tilde{y}_i = (\hat{\theta}_i / \tilde{\theta}_i) y_i$.

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

4. For each DMU $i = 1, \ldots, N$, calculate a bias corrected efficiency score $\hat{\theta}^b_{i\text{bc}}$ as $\hat{\theta} - \left( \frac{1}{m} \sum_{b=1}^{B_1} \hat{\theta}^b_i - \hat{\theta} \right)$. 

7
5. Run a truncated regression (left-truncation at 1) of $\hat{\theta}_i^{bc}$ on $z_i$ to obtain coefficient estimates $\hat{\beta}$ and an estimate for variance parameter $\hat{\sigma}$ by maximum likelihood.

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

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

6.2 Calculate artificial efficiency scores $\tilde{\theta}_i$ as $z_i \hat{\beta} + \tilde{\varepsilon}_i$ for each DMU $i = 1, \ldots, N$.

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

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

__simarwilson__ uses the inverse transform method for generating pseudo truncated normal random variates. 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 the estimation of percentile based confidence intervals yielding meaningful results. For $B_1$ and $B_2$ __simarwilson__ uses the default of 100 and 1000 bootstrap repetitions, respectively. The former default value is suggested in [Simar and Wilson (2007)](https://doi.org/10.1111/j.1467-9892.2006.00502.x), 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 $B_1$, and $B_2$ respectively. Running __simarwilson__, in particular algorithm #2, requires a substantial amount of computing time, which rapidly increases in the number of observations. For small samples, looping over truncated regression takes the lion’s share in computing time. If the sample is large, looping over DEA consumes relatively more time.

15. In rare cases, for which the linear prediction $z_i \hat{\beta}$ takes extreme values, generating pseudo truncated normal random variates may fail ([Chopin, 2011](https://doi.org/10.1093/biomet/asr012)). Therefore __simarwilson__ stops and issues an error message, if the initial truncated regression (step 2 or step 5) yields $\text{abs}(1 - z_i \hat{\beta})/\hat{\sigma} > 37.5$ 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 $B = 2000$, run time for algorithm #1 is 162 seconds, while it is 170 seconds for algorithm #2, with $B_1 = 1000$ and $B_2 = 2000$ (Stata/SE 15.1, Windows 10 Enterprise, Intel® Core™ i7-3520M 2.90 GHz, 8 GB RAM). Using the default values for $B$, $B_1$, and $B_2$ reduces run time to 79 and 81 seconds, respectively. If the the sample is expanded by the factors 2, 5, 10, and 25, run time for algorithm #1 is increased by the factors 1.1, 1.3, 1.6, and 2.7, respectively. The corresponding factors for algorithm #2 are 1.2, 1.7, 2.4, and 5.0.
2.3 Some minor extensions

The new Stata command simarwilson is meant to implement the above procedures one-to-one in Stata. It only deviates from what is suggested in Simar and Wilson (2007) 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) only consider the output oriented counterpart. This requires estimating an input oriented efficiency measure $\hat{\vartheta}_i$ in step 1 (algorithm #1) and steps 1 and 3.4 (algorithm #2) and interchanging the roles of inputs and outputs in the 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 $-z_i\hat{\beta}$ and right-truncation at $1-z_i\hat{\beta}$ (algorithm #2, step 6.1 $-z_i\hat{\beta}$ and 1 $-z_i\hat{\beta}$, respectively). By this, it is taken 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 only consider truncation from the right when analyzing input oriented efficiency. Using option notwosided seems questionable insofar as it rests on simulating a data generating process that is inconsistent with the non-negative nature of $\theta_i$. In particular, notwosided is not recommendable with algorithm #2.

- 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 non-increasing returns, this one-to-one correspondence does not hold. For option invert being 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 efficiency measures.

---

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 $\hat{\vartheta}_i < 0$ (step 3.2) and, in consequence, may make the bias correction fail ($\tilde{x}_i = (\hat{\vartheta}_i/\tilde{\vartheta}_i)x_i < 0$; step 3.3, alg. #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 how to respond to this problem is up to the user.

- One may assume a data generating process that deviates from (1) by considering log-(in)efficiency as left-hand-side variable (option **log**score), that is
  \[
  \ln(\theta_i) = z_i \beta + \epsilon_i
  \]
  Here \(\epsilon_i\) is assumed to be truncated normally distributed, with left-truncation at \(-z_i \beta\). If \(\ln(\theta_i)\) is considered as left-hand-side variable, truncation at \(-z_i \beta\) is from the right. If all \(\hat{\theta}_i\) are close to unity, specifying the option **log**score will make little difference. Yet, if the data include DMUs that according to the DEA are very inefficient, specifying **log**score 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**()); cf. Figure 3. Unlike **teradial**, it does not allow for considering DMUs as elements of the reference set for which no efficiency scores are estimated. Restricting the reference set to a sub-sample of the considered DMUs will regularly result in some irregular (super-efficient) 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). Users should, hence, carefully think about whether using the option **reference**() really makes sense.

- **simarwilson** allows for using efficiency scores which were beforehand estimated by some estimation procedure, using Stata or any other software. This effectively means that step one 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 respect (cf. **Simar and Wilson** 2007), deviate from what is computed in steps 1–4 of algorithm #2. Since using any kind of numeric, non-negative variable as 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.

- **simarwilson** allows for weighted estimation (only **pweights** and **iweights** are allowed). It is important to note that weights are immaterial for the DEA steps.

19. Note that (2) is not equivalent to \(\ln(\theta_i - 1) = z_i \beta + \zeta_i\), with \(\zeta_i \sim N(0, \sigma^2)\), 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 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 neighbourhood of \(\hat{\theta}_i = 1\) is relatively unlikely, i.e. \(\Pr(1 \leq \theta_i < 1 + \tau) < \Pr(1 + \tau \leq \theta_i < 1 + 2\tau)\) \(\forall i\) if \(\tau \to 0\).

20. This is technically infeasible since 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 user written command **dea** (Ji and Lee 2010) allows for this.
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 user written ado `teradial` including the associated plugin (st0444). With internal DEA the number of observations is limited by the value of [R] `matsize` that 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, `aweight`s and `fweight`s are not allowed; see [U] 11.1.6 weight. Weights only affect the truncated regression steps within `simarwilson` but not the DEA steps. If `iweight`s are used, (regression) numbers of observations are expressed in terms of rounded sums of weights.

3.1 Syntax for `simarwilson`

The syntax for `simarwilson` reads as follows:

```
simarwilson [ (outputs = inputs) ] [ depvar ] indepvars [ if ] [ in ] [ ,
    algorithm(1|2) notow sided logscore nounit rts(crs|nirs|vrs)
    base(output|input) reference(varname) invert tenure(newvar)
    tebc(newvar) biaste(newvar) reps(#) bcreps(#) saveall(name)
    bcsaveall(name) dots cnormal bbootstrap level(#) nomitted
    baselevels noprint nodeaprint trnoiseily ]
```

`outputs` is the list of outputs from the production process, while `inputs` is the corresponding list of inputs. Either `varlist` may only include numeric, non-negative variables. Factor variables and times-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 dependent variable. Specifying `depvar` is only possible, if `(outputs = inputs)` is not specified. That means, with `(outputs = inputs)` specified, any variable in the following `varlist` is interpreted as element of `indepvars`. `simarwilson` expects `depvar` to be a radial efficiency measure that is either bounded to the (0, 1] interval or to the [1, ∞) interval. This implies that `depvar` must not be measured in percent. If some values of `depvar` are smaller than one while others exceed one, `simarwilson` issues a warning and ignores observations, depending on how the option `nounit` is specified. This may happen if the preceding efficiency analysis was 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] 1.4.3 Factor variables. Time-series operators such as L. and F. are not allowed.

3.2 Options for simarwilson

algorithm(1|2) specifies whether algorithm #1 or #2 is applied. algorithm(2) requires (outputs = inputs). algorithm(1) is the default. If external DEA scores are used as depvar, one has to opt for algorithm(1) even if the externally estimated scores are bias corrected.\footnote{With bias corrected (externally estimated) scores in hand, steps 2–4 of algorithm #1 are fully equivalent to steps 5–7 of algorithm #2; cf. section 2.2}

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, ∞) 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 takes into account that input oriented (Farrell) efficiency scores are not only less than or equal to 1 but are also strictly positive. The latter is ignored with notwosided. Hence, with notwosided, simarwilson mirror-inverted applies the procedure suggested in Simar and Wilson (2007) – that only consider scores within [1, ∞) – to efficiency scores within (0, 1]. For (regular) efficiency scores ≥ 1, specifying notwosided has no effect. notwosided is not recommended in conjunction with algorithm(2).

logscore makes simarwilson use the natural logarithm of the efficiency score as 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. One rather has to specify the original untransformed score 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, ∞) interval, specifying the nounit option is also not required, since simarwilson recognizes which DMUs are inefficient and which are efficient. Only if external scores are used that are neither bounded to the (0, 1] interval nor to the [1, ∞) interval, nounit is required to specify which observation of depvar are regular (inefficient) ones and which are irregular (super-efficient) ones. Note that Simar and Wilson (2007) do not consider irregular (super-efficient) 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 non-increasing returns to scale, 
and vrs requests variable returns to scale. \textit{rts(crs)} is the default. \textit{rts()} is passed 
through to \textit{teradial}. If externally estimated scores are used, specifying \textit{rts()} has no effect.

\texttt{base(output|input)} specifies orientation/base of the radial measure of technical 
efficiency. output requests output orientation while input requests input orientation. 
\texttt{base(output)} is the default. \texttt{base()} is passed through to \textit{teradial} and has no effect if externally estimated scores are used.

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

\texttt{invert} makes \texttt{simarwilson} calculate and use the Shephard instead of the Farrell 
(default) efficiency measure. That is all estimated efficiency scores are inverted, 
unless they where externally estimated. \texttt{invert} is redundant for \texttt{base(crs)} since 
for constant returns to scale input oriented efficiency is just the reciprocal of output 
oriented efficiency. Hence rather than specifying \texttt{invert} one can just switch the 
base. Yet, this does not hold for \texttt{base(nirs)} and \texttt{base(vrs)}. With externally 
estimated scores, specifying \texttt{invert} has no effect. One rather has to manually invert 
the externally estimated scores prior to running \texttt{simarwilson}, if one wants to switch 
between the Farrell and the Shephard measure.

\texttt{tename(newvar)} creates the new variable \texttt{newvar} that contains estimates of radial 
technical efficiency (DEA scores).

\texttt{tebc(newvar)} creates the new variable \texttt{newvar} that contains bias corrected estimates 
of radial technical efficiency (bias corrected DEA scores). \texttt{tebc(newvar)} requires 
algorithm(2).

\texttt{biaste(newvar)} creates the new variable \texttt{newvar} that contains bootstrap bias estimate 
for original radial measures of technical efficiency. \texttt{biaste(newvar)} requires 
algorithm(2).

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

\texttt{bcreps(#)} specifies the number of bootstrap replications for the bias correction of DEA
scores. The default is 100 replications 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.

`bcsaveall(name)` makes `simarwilson` save all bootstrap efficiency scores that are estimated in the bias correction procedure to the \((bcreps \times N_{dea})\) 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 only specifying the option `cinormal`.

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

`level(#)`; see [R] `level` estimation options. One may change the reported confidence level by retyping `simarwilson` without arguments and only specifying 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 to be displayed. The default is to include in the results table any variables omitted because of collinearity and to label them as omitted by the `o.` prefix.

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

`noprint` prevents `simarwilson` from displaying warnings. Error messages are displayed irrespective of whether or not `noprint` is specified.

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

`trnoisily` makes `simarwilson` display genuine output of `truncreg` for the initial truncated regression(s) (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.

In addition, `simarwilson` 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 Saved results for simarwilson

simarwilson saves the following results to e():

Scalars

| e(N) | # of observations (inefficient DMUs) |
| e(N_lim) | # of limit observations (efficient DMUs) |
| e(N_irreg) | # of irregular observations (super-efficient DMUs) |
| e(N_all) | overall # of observations (all DMUs with eff. score) |
| e(wgtsum) | sum of weights (if weights are specified) |
| e(sigma) | estimate of sigma (initial truncated reg.) |
| e(ll) | pseudo log-likelihood (initial truncated reg.) |
| e(ic) | # of iterations (initial truncated reg.) |
| e(converged) | 1 converged, 0 otherwise (initial truncated reg.) |
| e(rc) | return code |
| e(k_eq) | # of equations in e(b) |
| e(df_m) | model degrees of freedom |

| e(chi2) | model chi-squared |
| e(p) | model significance, p-value |
| e(N_reps) | # of completed bootstrap reps |
| e(N_reps) | # of limit observations (efficient DMUs) |
| e(N_reg) | # of limit observations (efficient DMUs) |
| e(N) | # of observations (inefficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMUs) |
| e(N) | # of limit observations (efficient DMU
Matrices
- \( e(b) \): vector of estimated coefficients
- \( e(V) \): estimated coefficient variance-covariance matrix
- \( e(Cns) \): constraints matrix (if constraints are specified)
- \( e(b_{bstr}) \): bootstrap estimates of coefficients
- \( e(bias_{bstr}) \): bootstrap estimated biases
- \( e(ci\_percentile) \): bootstrap percentile confidence intervals

Functions
- \( e(sample) \): marks estimation sample

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

### 3.4 simarwilson postestimation

The postestimation commands that are 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 \( E(\theta_i | \theta_i > 1, z_i) \) (Farrell output oriented) and \( E(\theta_i | 0 < \theta_i < 1, z_i) \) (Farrell input oriented), respectively. 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`.

Users should, in general, be careful in interpreting the results one obtains from postestimation commands, such as `predict`, used after `simarwilson`. The postestimation commands treat the results of `simarwilson` as if they where generated by `truncreg`. One should, however, 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].

[1] Since the data generating process (1) assumes a truncated distribution for \( \varepsilon_i \), \( E(\theta_i | \theta_i > 1, z_i) \) coincides with \( E(\theta_i | z_i) \). i.e. \( \theta_i < 1 \) is not only not observed, it is rather impossible according to [1]. The same line of argument applies to \( E(\theta_i | 0 < \theta_i < 1, z_i) \) 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 \( E(\theta_i | \theta_i < 1, z_i) \) that is the option `predict(e(.,1))` is internally set.

[2] With option `logscore` the default is `predict(e(0,))` and `predict(e(.,0))`, respectively. That is `margins`, by default, yields semi-elasticities.
4 An application of estimation methods

4.1 Comparison of estimation methods

In order 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 that is provided through the Penn World Table data base, version 9 (Feenstra et al. 2015) and the World Economic Forum, Global Competitiveness Report, version 2018-02-26 (World Economic Forum 2018; Schwab 2017). Though both data bases are publicly available on the internet, only the Penn World Table allows for being straightforwardly used with Stata. For this reason, this article is accompanied by the user written ado-file gciget.ado that facilitates the retrieval of the Global Competitiveness Index data using Stata. See subsection 4.3 for a more detailed description of gciget. The Stata log below is from using gciget to load three selected variables (EOSQ048, EOSQ051, EOSQ144) of the Global Competitiveness Index 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/gcri17. 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. As key explanatory variables we consider the ‘burden of government regulation’ (EOSQ048), ‘property rights protection’ (EOSQ051), and ‘judicial independence’ (EOSQ144). While all 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 administrations requirements (e.g., 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). In order 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 allow for 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. Since simarwilson does not allow for times-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 that contains the comprehensive list of explanatory variables. Since 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. 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 a randomness, this should be avoided in own applications.

```
. qui gen regu = EOSQ048[_n-1] if countrycode == countrycode[_n-1]
. qui gen prop = EOSQ051[_n-1] if countrycode == countrycode[_n-1]
. qui gen judi = EOSQ144[_n-1] if countrycode == countrycode[_n-1]
. qui gen 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. Since we do not define a reference set that deviates from the sample for which efficiency measures are estimated, the option base(o) makes te_vrs_o taking values equal or larger than one. Then we let Stata report descriptive statistics for

---

25. The default random number generator (mt64) of Stata 15 is used.
26. This is makes the DEA steps in algorithms #1 and #2 using the same sample. Only for the latter, the right-hand-side variables are required for estimating (bias corrected) efficiency scores; cf. step 3, alg. #2, p. 7.
27. Unless option invert is used, positive coefficient of a variable in simarwilson would imply it has a negative effect on efficiency. Conversely, negative coefficient would mean that the variable has positive effect on efficiency.
the variables used in the subsequent regressions. Due to 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 <. & lp > op <., te(te_vrs_o) rts(v) base(o) nopr
. sum te_vrs_o regu prop judi lp if e(sample)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Obs</th>
<th>Mean</th>
<th>Std. Dev.</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>te_vrs_o</td>
<td>131</td>
<td>1.6999</td>
<td>.6236905</td>
<td>1</td>
<td>5.5138</td>
</tr>
<tr>
<td>regu</td>
<td>131</td>
<td>3.4351</td>
<td>.6711715</td>
<td>1.8462</td>
<td>5.4226</td>
</tr>
<tr>
<td>prop</td>
<td>131</td>
<td>4.3046</td>
<td>1.030568</td>
<td>1.6103</td>
<td>6.3783</td>
</tr>
<tr>
<td>judi</td>
<td>131</td>
<td>3.8971</td>
<td>1.315987</td>
<td>1.1132</td>
<td>6.6783</td>
</tr>
<tr>
<td>lp</td>
<td>131</td>
<td>2.5665</td>
<td>1.586448</td>
<td>-1.2641</td>
<td>7.2171</td>
</tr>
</tbody>
</table>

As the next step of the analysis, we use four empirical models to explain (in)efficiency in GDP generation. Besides simarwilson, algorithm(1) and algorithm(2) we also consider tobit and truncreg as references. Since the model coefficients themselves do not allow for being straightforwardly 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 outcome of the underlying data generating process rather than an artifact of finite sample bias. 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 judical independence being the only exception, the governance variables are individually significant at the 10% level and bear the expected negative signs. However, because of the model including 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 trough less business regulation and better protection of property rights are first of all a matter of small countries.

28. The options nolstretch and vsquish are just for making the displayed output fit on a printed page.
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      | -0.3925008 | 0.2014823 | -1.95 | 0.054 | [-0.7264042, -0.0585973] |
| prop      | -0.5199721 | 0.2574393 | -2.02 | 0.046 | [-0.9466096, -0.0933347] |
| judi      | 0.2488415  | 0.1888903 | 1.32  | 0.190 | [-0.0641949, 0.5618771] |
| lpop      | -0.8211409 | 0.2667289 | -3.08 | 0.003 | [-1.263173, -0.3791084] |
| c.regu#   | 0.1484147  | 0.0687327 | 2.16  | 0.033 | [0.0345084, 0.2623209] |
| c.lpop#   | 0.1251518  | 0.0871451 | 1.44  | 0.153 | [-0.0192682, 0.2695717] |
| c.prop#   | -0.0858924 | 0.0693701 | -1.24 | 0.218 | [-0.2008849, 0.0290701] |
| c.judi#   | 4.589835   | 0.7747277 | 5.92  | 0.000 | [3.305929, 5.873741]   |
| _cons     | 4.589835   | 0.7747277 | 5.92  | 0.000 | [3.305929, 5.873741]   |

.var(e.te_v~o) .4098449 .0562185 .3265083 .514452

Then we turn to the truncated regression by using \texttt{truncreg}. Unlike \texttt{tobit}, this approach does not consider observations for which \texttt{te_vrs_o} = 1 holds. For this reason, we use the option \texttt{predict(e(1,.))} when estimating marginal effects. The estimated coefficients look quite different compared to their counterparts from \texttt{tobit}. Yet in terms of the signs, the results are similar to their counterparts from \texttt{tobit}. According to the results from \texttt{truncreg} judicial independence seem to matter for efficiency, since both \texttt{judi} and its interaction with \texttt{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 = -68.518139
Iteration 2:  log likelihood = -67.617016
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

|        | Coef.  | Std. Err. | z     | P>|z|  | [90% Conf. Interval] |
|---|--------|-----------|------|------|----------------------|
| regu | -.9258069 | .4299484 | -2.15 | 0.031 | -1.633009 to -0.2186048 |
| prop | -1.243902 | .4991533 | -2.49 | 0.013 | -2.064936 to -0.4228676 |
| judi | 0.7784162 | .3780368 | 2.06  | 0.039 | 1.166010 to 1.380231  |
| lpop | -1.7399993 | .5952224 | -2.92 | 0.003 | -2.719046 to -0.7609389 |
| c.regu# | .4253728 | .1720618 | 2.47  | 0.013 | 0.1423563 to 0.7083894 |
| c.lpop | .2581352 | .1794841 | 1.44  | 0.150 | -.0370899 to 0.5533604 |
| c.prop# | -0.2592945 | 0.1497392 | -1.73 | 0.083 | -0.5055935 to -0.0129955 |
| c.lpop | 7.447817 | 1.629842 | 4.57  | 0.000 | 4.766965 to 10.12867 |

/\sigma | .72229122 | .0925133 | 7.81  | 0.000 | 0.5701204 to 0.8744621 |

. qui margins, dydx($g_list) predict(e(1,.)) post
. estimates store truncreg

Hence, in the next step, we turn to simarwilson, algorithm(1). Since externally estimated efficiency scores are already available, we do not rerun the DEA within simarwilson but use te_vrs_o as dependent variable. Using the (rgdpo = ck emp hc) syntax instead, and specifying the options rts(v) and base(o) would have generated identical results. Since we report percentile confidence intervals for the coefficients, we request a large number (2000) for the bootstrap replications. This choice results in a substantial computing time of 162 seconds (Stata/SE 15.1). Specifying the option predict() is not required for margins, since the appropriate specification is set internally. As a practical matter we advise to use 1 processor in the MP version of Stata by typing set processors 1 before executing simarwilson. The estimated coefficient necessarily coincide with what we got from truncreg, since simarwilson, algorithm(1) only affects the estimated standard errors and confidence intervals. Yet, even with respect to the latter two, the deviation from their naïve counterparts from truncreg is rather moderate. This is in line with what is frequently found in applications of algorithm #1.

29. Carrying out the DEA internally affects computing time just marginally.

21
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

| inefficiency | Observed Coef. | Bootstrap Coef. | Std. Err. | z    | P>|z| | [90% Conf. Interval] |
|--------------|----------------|-----------------|-----------|------|------|----------------------|
|              |                |                 |           |      |      |                      |
| te_vrs_o     |                |                 |           |      |      |                      |
| regu         | -0.9258069     | 0.4021808       | -2.30     | 0.021| -1.61701| -1.61701 - 0.2838173 |
| prop         | -1.243902      | 0.4715584       | -2.64     | 0.008| -2.042034| -2.042034 - 0.3567566 |
| judi         | 0.7784156      | 0.356048        | 2.19      | 0.029| 1.985666| 1.985666 - 3.517163 |
| lpop         | -1.739993      | 0.5868841       | -3.06     | 0.002| -2.670876| -2.670876 - 0.8296476 |
| c.regu#c.l~p | 0.4253728      | 0.1611459       | 2.64      | 0.008| 1.649802| 1.649802 - 0.203981 |
| c.prop#c.l~p | 0.2581362      | 0.1692766       | 1.52      | 0.127| -0.13744| -0.13744 - 0.5414851 |
| c.judi#c.l~p | -0.2592945     | 0.1400465       | -1.85     | 0.064| -0.4829779| -0.4829779 - 0.301317 |
| _cons        | 7.447817       | 1.557957        | 4.78      | 0.000| 4.985354| 4.985354 - 9.974977 |

/obsigma .7222912 .0877174 8.23 0.000 .5537709 .8368937

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 let simarwilson carry out the bias correction internally. This requires the (rgdpo = ck emp hc) syntax along with the options rts(v) and base(o). The latter 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 1000 replications in the bias correction bootstrap, which is well above the default suggested in Simar and Wilson (2007). Estimating this model takes 170 seconds. Due to the relatively small sample size, using algorithm(2) increases computing time by only 5%; cf. footnote 16. Since we do not use externally estimated scores as left-hand-side variable, but let simarwilson run the DEA internally, the reported output also involves comprehensive information about the DEA model used. In the present application, using bias corrected instead of uncorrected scores has just a moderate impact on the estimated coefficients and the associated estimated confidence intervals, see the output below.

30. The option nodeaprint suppresses displaying the DEA related information.
Simar & Wilson (2007) eff. analysis Number of obs = 131
(algorithm #2) Number of efficient DMUs = 0
inefficient if tebc_vrs_o > 1 Prob > chi2(7) = 0.0036

Data Envelopment Analysis: Number of 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

| inefficiency  | Observed 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      | .508764   | -2.39| 0.017 | -2.062226 - .4182389 |
| judi          | .6717764       | .3638639  | 1.85 | 0.065 | .0856124 - 1.271846 |
| lpop          | -.796633       | .5587313  | -1.42| 0.001 | -.8933189 - .2.755797 |
| c.regu#c.l~p  | .4237473       | .17585503 | 1.85 | 0.065 | .1775733 - .6736762  |
| c.prop#c.l~p  | .2335061       | .17585503 | 1.33 | 0.083 | -.0346361 - .53319   |
| c.judi#c.l~p  | -.2302095      | .17585503 | 1.33 | 0.083 | -.1.45712 - .0115268 |
| _cons         | 7.887194       | 1.604777  | 4.91 | 0.000 | 5.281076 - 10.45693 |

/\sigma = .8735555 .1007678 8.67 0.000 .6807801 1.010869

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

<table>
<thead>
<tr>
<th>tebc_vrs_o</th>
<th>Obs</th>
<th>Mean</th>
<th>Std. Dev.</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>131</td>
<td>1.956169</td>
<td>.7083735</td>
<td>1.068883</td>
<td>6.482759</td>
</tr>
</tbody>
</table>

In order to allow for interpreting the results in qualitative terms 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 an 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.
Measuring effects in terms of Farrell (output oriented) efficiency units appears not to be particularly telling. One may, hence, prefer a scaled efficiency measure, that allows for interpreting marginal effects in terms of percentage points. This calls for switching from the Farrell 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 the present 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 inputs variables.

While switching to the Shephard measure was straightforward for algorithm(1) – one just has to use the reciprocal of \( te_{vrs.o} \) as dependent variable – in the present application it causes difficulties with algorithm(2). As indicated by a warning issued by simarwilson (see below), the bias correction yields some negative scores; cf. subsection 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 left-hand-side variable does not change the general pattern of results. As expected (see footnote 27), the signs of all coefficients are just reversed and all coefficients remain statistically significant.
One may force `simarwilson` to use negative bias corrected scores in the regression analysis by combining `invert` with the option `notwosided`. By doing this one accepts, however, two inconsistencies. 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 noticeable impact on the estimated coefficients.
In order to specify a model which renders interpreting estimation results in quantitative terms more convenient, using the option \texttt{logscore} is a possible alternative to \texttt{invert}. By considering log-inefficiency as left-hand-side variable, marginal effects can be interpreted as percentage reductions in inefficiency. Hence we rerun our preferred model (algorithm #2, Farrell output-oriented efficiency) using the option \texttt{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, alg(2) rts(v) base(o > ) reps(2000) bcreps(1000) logscore

Simar & Wilson (2007) eff. analysis
(algorithm #2)

Number of obs = 131
Number of efficient DMUs = 0
Number of bootstr. reps = 2000
Wald chi2(7) = 37.48
inefficient if ln(bcscore) > 0
Prob > chi2(7) = 0.0000

Data Envelopment Analysis:
Number of DMUs = 131
Number of ref. DMUs = 131
Number of outputs = 1
Number of inputs = 3
bias corrected efficiency measure
Number of reps (bc) = 1000

<table>
<thead>
<tr>
<th>Observed</th>
<th>Bootstrap</th>
<th>Percentile</th>
</tr>
</thead>
<tbody>
<tr>
<td>ln(bcscore)</td>
<td>Coef.</td>
<td>Std. Err.</td>
</tr>
<tr>
<td>regu</td>
<td>-.2212588</td>
<td>.0942768</td>
</tr>
<tr>
<td>prop</td>
<td>-.3127128</td>
<td>.1199882</td>
</tr>
<tr>
<td>judi</td>
<td>.1617935</td>
<td>.0877577</td>
</tr>
<tr>
<td>lpop</td>
<td>-.4737086</td>
<td>.1279737</td>
</tr>
<tr>
<td>c.regu#c.l~p</td>
<td>.0927766</td>
<td>.0340446</td>
</tr>
<tr>
<td>c.prop#c.l~p</td>
<td>.0777964</td>
<td>.0414616</td>
</tr>
<tr>
<td>c.judi#c.l~p</td>
<td>-.0644932</td>
<td>.0323986</td>
</tr>
<tr>
<td>_cons</td>
<td>2.312936</td>
<td>.3706691</td>
</tr>
<tr>
<td>/sigma</td>
<td>.2882745</td>
<td>.020727</td>
</tr>
</tbody>
</table>

. qui 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; cf. Figure 1. One way of addressing this issue, is to once again envelope the non-convex bias corrected estimated frontier by a convex hull and to use the distance to this convexified bias corrected frontier as dependent variable in the regression analysis (cf. Badunenko et al. 2013, and Figure 5). The \textit{(ref\_outputs = ref\_inputs)} specification of \textit{teradial} allows for straightforwardly implementing this procedure; see the below Stata log and Badunenko and Mozharovsky (2016). Compared to its direct counterpart (\textit{simarwilson, algorithm}(2) without \textit{invert} and \textit{logscore}), using this once more adjusted efficiency measure changes the estimated coefficients markedly. Yet, qualitatively, the pattern of estimates remains the same.
Simar & Wilson (2007) eff. analysis
(algorithm #1)
Number of obs = 131
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

| inefficiency | Observed Coef. | Bootstrap Std. Err. | z  | P>|z| | Percentile 90% Conf. Interval |
|--------------|----------------|---------------------|----|------|-----------------------------|
| tebc_vrs_o-x | -0.8525049     | 0.351338            | -2.43 | 0.015 | -1.436492 to -0.2731832 |
| regu         | -1.104847      | 0.4583591           | -2.41 | 0.016 | -1.884811 to -0.3711119    |
| prop         | -0.5759623     | 0.3329715           | 1.73  | 0.084 | 0.0401372 to 1.122519     |
| judi         | -1.748744      | 0.5217818           | -3.35 | 0.001 | -2.625194 to -1.919087    |
| lpop         | 0.3752556      | 0.1349932           | 2.78  | 0.005 | 0.1569001 to 0.5928116    |
| c.regu#c.l-p | 0.2194959      | 0.1349932           | 1.69  | 0.094 | -0.0314325 to 0.4085952   |
| c.prop#c.l-p | 0.1856799      | 0.1233833           | -1.50 | 0.132 | -0.3868211 to 0.0144541   |
| _cons        | 7.956469       | 1.475931            | 5.39  | 0.000 | 5.576847 to 10.362        |
| /sigma       | 0.8986468      | 0.0911453           | 9.86  | 0.000 | 0.7240245 to 1.017877     |

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 improving efficiency (p-values 0.039 and 0.096, respectively). One may, hence, speculate that regu not only captures detrimental but also beneficial facets of business regulation. In terms of the point estimates, all model specification yield a positive association of property right protection and GDP efficiency. Only for the Shephard measure as left-hand-side variable (without option notwosided) the average marginal effect of prop turns 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 percent reduction in inefficiency. With respect to judi, the estimated marginal effects are throughout statistically insignificant. In terms of estimated average marginal effects, basing the analysis on a convex estimated hull has almost no effect as compared to using the non-convex, bias corrected estimated frontier.
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 heterogenous effects. We graphically examine possible effect heterogeneity using [R] marginsplot command; see [R] marginsplot and the Stata log below. We consider two dimensions of heterogeneity: heterogeneity with respect to country size measured by lpop (Fig. 2 right panel), and heterogeneity with respect to the respective considered dimension of governance quality (Fig. 2 left panel).

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

29
Figure 2: Estimated marginal effects of governance quality indices on inefficiency by country size (right column) and its respective own value (left column). Notes: Farrell output oriented efficiency as dependent variable; Simar and Wilson (2007), algorithm #2 used for estimation; 90% confidence bands indicated by shaded areas. Source: Own calculations based on Penn World Table and World Economic Forum Global Competitiveness Report data.
The left panel of Fig. 2 does not suggest that effect heterogeneity with respect to the respective category of governance quality is a big issue, at least qualitatively. The effects of both, ‘burden of government regulation’ and ‘judicial independence’ on inefficiency are statistically insignificant at any level of \textit{regu} and \textit{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 \textit{regu} and for a moment ignores statistical significance, then Fig. 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 much 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 \textit{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 \textit{prop}.

The overall picture is somewhat different for heterogeneity with respect to country size (Fig. 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, Fig. 2 suggests that relaxing government regulation reduces inefficiency in small countries. Yet in big countries, it seems to exerts a negative effect on national efficiency. This pattern corroborates our earlier hypothesis of ‘regulatory burden’ being an ambiguous concept, since in certain circumstances some regulation may be well required for efficient production. A similar pattern of heterogeneity is found for the effect of \textit{prop}. In small countries, improving the protection of property rights is clearly beneficial for efficiency of GDP production, while for big counties such effect is not found in the data. The reverse pattern of heterogeneity is found with respect to judicial independence. While the effect of \textit{judi} on efficiency is statistically insignificant for a wide range of values of \textit{lpop}, Fig. 2 suggests a statistically significant, efficiency reducing effect for very small countries. This somewhat surprising finding has, however, to be interpreted with caution. Near collinearity might be a technical explanation for the mirror inverted patterns found for \textit{prop} and \textit{judi}. Both variables are strongly correlated (0.903) in the estimation sample, while their respective correlations with \textit{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 Stata command \texttt{gciget} to get the indices directly into the memory of Stata from the World Economic Forum’s Global Competitiveness Report.

\texttt{gciget} proceeds in three steps. First it downloads the excel file \texttt{GCI_Dataset_2007-2017.xlsx} from the The Global Competitiveness Report section (http://wef.ch/gcr17) of the World Economic Forum website. The user can optionally indicate the path to the excel file \texttt{GCI_Dataset_2007-2017.xlsx} stored locally. Second, \texttt{gciget} imports the
Excel file. See \texttt{import excel} regarding the requirement for the version of Stata.

Third, \texttt{gciget} processes the variables that use has specified after \texttt{gciget}. The resulting data are in a long format and are by default declared to be panel data, see \texttt{xtset}.

The syntax for \texttt{gciget} reads as follows:

\begin{verbatim}
gciget [varlist] [, options]
\end{verbatim}

The user can optionally specify the \texttt{varlist} from the list of indices in the Global Competitiveness Report (see the excel file \texttt{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:

\begin{itemize}
  \item \texttt{clear} replace data in memory
  \item \texttt{noxtset} do not declare the loaded data to be panel data
  \item \texttt{noquery} suppress summary calculations by \texttt{xtset}
  \item \texttt{panelvar(newvarname)} generate numeric panelvar \texttt{newvarname}
  \item \texttt{url(filename)} download link
  \item \texttt{sheet(''sheetname'')} excel worksheet to load
  \item \texttt{cellrange([start][:end])} excel cell range to load
  \item \texttt{nowarnings} do not display warnings
\end{itemize}

\texttt{gciget} only helps user get the data from the World Economic Forum into Stata. Thus any liability for the data or its usage is disclaimed. That the data comes 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 illustrate a simple import of four indices and plotting GCI index for four countries.

\begin{verbatim}
. gciget EOSQ048 EOSQ051 GCI GCI.A.02.01, clear
\end{verbatim}

\texttt{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.

\texttt{Downloading the GCI_Dataset_2007-2017.xlsx file}
\texttt{Importing the GCI_Dataset_2007-2017.xlsx file}
\texttt{Processing EOSQ048: 1.09 Burden of government regulation, 1-7 (best)}
5 Summary and conclusions

In this article, the new user written Stata command *simarwilson* was introduced that implements Simar and Wilson (2007) two-stage efficiency analysis. This estimator has substantial value for applied efficiency analysis as it puts regression analysis of DEA scores on firm statistical ground. The new Stata command extends the originally proposed procedure in some (minor) respects, which increases its applicability in applied empirical work. *simarwilson* complements the contributions of Ji and Lee (2010), Tauchmann (2012), and in particular Badunenko and Mozharovskyi (2016), who have already made related methods of non-parametric efficiency analysis available to Stata users.
6  Acknowledgements

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 (DFG). 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, Christian Merkl, and participants of the 2015 German Stata Users Group meeting 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 was used to illustrate DEA in Figure 1.

Figure 4: Estimated inefficiency for sub-sample of DMUs used as reference. Considering only as sub-sample of DMUs as reference set renders DMU B seemingly super-efficient, both according to conventional DEA ($\hat{\theta}_B = \eta_B^{DEA_\ast}/\eta_B < 1$) and according to bias corrected DEA ($\hat{\theta}_B^{bc} = \eta_B^{bc_\ast}/\eta_B < 1$). DMU A, is still estimated to be inefficient, cf. Figure 1. Yet, the magnitude of estimated inefficiency is somewhat smaller. Note: Artificial data generated in the same way as for Figure 1. Source: Own calculations.
Figure 5: Convexified bias corrected estimated frontier. Measuring inefficiency relative to the convexified bias corrected frontier either does not affect estimated bias corrected inefficiency (e.g. DMU B) or increases estimated bias corrected inefficiency (e.g. DMU A). Note: Artificial data generated in the same way as for Figure 1. Source: Own calculations.

8 Software availability

The ado-files `simarwilson.ado` and `gciget.ado` and the accompanying help-files are available from `ssc`. Type

```
ssc install simarwilson
```

and

```
ssc install gciget
```

to install the ado-files on your machine.

9 References


