The Stata Journal

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

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

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

http://www.stata-journal.com

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

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

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

U.S. and Canada		Elsewhere	
1-year subscription	$ 79	1-year subscription	$115
2-year subscription	$155	2-year subscription	$225
3-year subscription	$225	3-year subscription	$329
3-year subscription (electronic only)	$210	3-year subscription (electronic only)	$210
1-year student subscription	$ 48	1-year student subscription	$ 79
1-year university library subscription	$ 99	1-year university library subscription	$135
2-year university library subscription	$195	2-year university library subscription	$265
3-year university library subscription	$289	3-year university library subscription	$395
1-year institutional subscription	$225	1-year institutional subscription	$259
2-year institutional subscription	$445	2-year institutional subscription	$510
3-year institutional subscription	$650	3-year institutional subscription	$750

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

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

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

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

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

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

The Stata Journal (2012)
12, Number 3, pp. 461–478

Partial frontier efficiency analysis

Harald Tauchmann
Rheinisch-Westfälisches Institut für Wirtschaftsforschung (RWI)
and CINCH (Centre of Health Economics Research)
Essen, Germany
harald.tauchmann@rwi-essen.de

Abstract. Despite their frequent use in applied work, nonparametric approaches to efficiency analysis—namely, data envelopment analysis and free disposal hull—have bad reputations among econometricians. This is mainly because data envelopment analysis and free disposal hull represent deterministic approaches that are highly sensitive to outliers and measurement errors. However, so-called partial frontier approaches have recently been developed, namely, order-m and order-α. These approaches generalize free disposal hull by allowing for superefficient observations to be located beyond the estimated production-possibility frontier. Although these methods are also purely nonparametric, the sensitivity to outliers is substantially reduced by partial frontier approaches enveloping just a subsample of observations. In this article, I introduce the new Stata commands `orderm` and `orderalpha`, which implement order-m, order-α, and free disposal hull efficiency analysis in Stata. The commands allow for several options, such as statistical inference based on subsampling bootstrapping.

Keywords: st0270, orderalpha, orderm, nonparametric, efficiency, partial frontier, free disposal hull, outlier-robust, decision-making unit

1 Introduction

Countless empirical analyses address the efficiency of production units, which in the relevant literature are frequently referred to as decision-making units (DMUs). There are two major methodical approaches to efficiency measurement: parametric and nonparametric approaches. Among the former, the most common are stochastic frontier models (Aigner, Lovell, and Schmidt 1977), which augment a classical regression model with a nonpositive error term, capturing inefficiency in production. Stochastic frontier analysis is implemented in Stata by the `frontier` command.

In contrast, nonparametric approaches—data envelopment analysis (DEA), introduced by Charnes, Cooper, and Rhodes (1978); and the free disposal hull (FDH), introduced by Deprins, Simar, and Tulkens (1984)—are not embedded in a regression framework familiar to econometricians. Rather, they are based on nonparametrically enveloping a given sample of data with a piecewise linear hull. While DEA assumes a convex technology and employs linear programming for enveloping the data, FDH is based on the principle of weak dominance and departs from the convexity assumption inherent in DEA. That is, FDH envelops the data with a nonconvex staircase-hull (see Cooper, Seiford, and Tone [2007] for a comprehensive discussion of DEA and FDH). DEA has recently been made available to Stata users through the ado-file `dea`, written by Yong-Bae Ji and Choonjoo Lee (Ji and Lee 2010).

The pros and cons of parametric and nonparametric approaches have been intensely debated. The parametric approaches have been criticized for relying on restrictive assumptions concerning the functional form and the distribution of random errors, for relying on input quantities as explanatory (which in all likelihood are endogenous), and for accommodating only single-output technologies.[1] The nonparametric approaches have been criticized by econometricians for being deterministic approaches, lacking a well-defined data-generating process, and, more relevant, for being extremely vulnerable to outliers and measurement error.

The final objection to nonparametric efficiency measurement has recently been addressed by so-called partial frontier approaches, in particular, order-m (Cazals, Florens, and Simar 2002) and order-α (Aragon, Daouia, and Thomas-Agnan 2005) efficiency. These approaches generalize FDH by allowing for superefficient observations to be located beyond the estimated production-possibility frontier.[2] Hence, the estimated frontier will not entirely be shaped by few abnormal observations, which might represent artifacts of measurement error. This renders partial frontier approaches less vulnerable to outliers than DEA or FDH. This article contributes to nonparametric efficiency analysis by introducing the new Stata commands `orderm` and `orderalpha`, which implement order-m and order-α, respectively.

The following section sets out the framework of partial frontier efficiency analysis. The syntax of `orderalpha` and `orderm` is described in section 3. Section 4 illustrates the application of `orderalpha` and `orderm` with a simple example. Section 5 summarizes and concludes the article.

2 The concept of partial frontier analysis

Consider a sample of N DMUs. A set of inputs to production $(x_{i1}, \ldots, x_{iK})$ and a set of outputs from production $(y_{i1}, \ldots, y_{iL})$ is observed for each DMU, $i = 1, \ldots, N$. The prime objective of efficiency measurement is to calculate an efficiency score θ_i for each DMU. Typically, two variants are considered: 1) input-oriented efficiency θ_i^{inp}, the factor by which input consumption of DMU i can proportionally be reduced while leaving outputs unchanged; and 2) output-oriented efficiency θ_i^{out}, the factor by which output generation can be proportionally increased while leaving input consumption unchanged. These concepts differ in terms of the direction in which the distance of an observed data point from the efficiency frontier is measured. While input-oriented efficiency measures the relative radial distance in input direction, output-oriented efficiency measures the

1. The latter two objections do not apply if a cost frontier rather than a production frontier is estimated.
2. For partial frontier approaches, superefficiency has predominantly been discussed in the context of dealing with outliers and measurement error. Yet with respect to DEA, for which superefficiency can be achieved by excluding a DMU from its own reference set, several other uses of superefficiency have been proposed, such as overcoming truncation problems and ranking efficient DMUs (see Lovell and Rouse [2003]).

relative radial distance in output direction.[3] For full frontier models for which all DMUs are enveloped by the production-possibility frontier, $\theta_i^{\text{inp}} \in (0,1]$ and $\theta_i^{\text{out}} \in [1,\infty)$ apply. That is, efficient DMUs are characterized by efficiency scores taking the value of 1, while downward (input-oriented) and upward (output-oriented) deviations from unity indicate inefficiency. In contrast, partial frontier approaches allow scores to exceed (input-oriented) or to fall short of (output-oriented) the value of 1. To avoid redundancies, in the following, we will focus on input-oriented efficiency. Yet, all arguments below analogously apply to output-oriented efficiency.

2.1 The free disposal hull

Because partial frontier approaches generalize FDH, a short preliminary discussion of the latter is required. Here (input-oriented) efficiency is estimated by comparing each DMU, $i = 1, \ldots, N$, with all other DMUs, $j = 1, \ldots, N$, in the data that produce at least as much of any output as DMU i. The set of peer DMUs in the sample that satisfy the condition $y_{lj} \geq y_{li} \, \forall \, l$ is denoted as B_i. Among the peer DMUs, the one that exhibits minimum input consumption serves as a reference to i, and $\widehat{\theta}_i^{\text{FDH}}$ is calculated as relative input use:[4]

$$\widehat{\theta}_i^{\text{FDH}} = \min_{j \in B_i} \left\{ \max_{k=1,\ldots,K} \left(\frac{x_{kj}}{x_{ki}} \right) \right\} \tag{1}$$

DMUs that exhibit minimum input consumption among all their peers serve as their own reference. For these DMUs, which span the estimated production-possibility frontier, $\widehat{\theta}_i^{\text{FDH}}$ takes the value of 1. Unfortunately, even a single DMU in the data that exhibits abnormally little—possibly misreported—input consumption renders inefficient all other DMUs to which it is a peer. Thus FDH is highly sensitive to outliers and measurement error.

2.2 Order-m efficiency

Order-m generalizes FDH by adding a layer of randomness to the computation of efficiency scores. Rather than benchmarking a DMU by the best-performing peer in the sample at hand, order-m is based on the idea of benchmarking the DMU by expected best performance in a sample of m peers. In computational terms, order-m efficiency follows a four-step procedure (Daraio and Simar 2007, 72):

3. One may consider other directions, too, yet these are the most common. Note that the estimated production-possibility frontier is the same for input-oriented and output-oriented efficiency in full frontier models (DEA and FDH). Nevertheless, DMUs that are located at the FDH frontier—but not at one of its corners—are FDH efficient only in terms of either output-oriented or input-oriented efficiency (compare to the discussion about "slack values" in the context of DEA). For partial frontier models, the fitted frontier depends on direction.
4. Equation (1) focuses on calculating $\widehat{\theta}_i^{\text{FDH}}$ from a given sample of data. This analogously applies to (2) and (3). For a more theory-oriented coverage of FDH, order-m, and order-α, see Daraio and Simar (2007), pages 34, 68, and 72, respectively.

1. From B_i, a sample of m peer DMUs is randomly drawn with replacement.

2. Pseudo-FDH efficiency $\widehat{\theta}_{mi}^{\widetilde{\text{FDH}}_d}$ is calculated using this artificial reference sample.

3. Steps 1 and 2 are repeated D times.

4. Order-m efficiency is calculated as the average of pseudo-FDH scores:

$$\widehat{\theta}_{mi}^{\text{OM}} = \frac{1}{D}\sum_{d=1}^{D}\widehat{\theta}_{mi}^{\widetilde{\text{FDH}}_d} \tag{2}$$

Because of random resampling, in each replication d, DMU i may or may not be available as its own peer. For this reason, (input-oriented) order-m efficiency scores may exceed the value of 1. Consequently, order-m allows for superefficient DMUs located beyond the estimated production-possibility frontier. This is the key difference to FDH, where a DMU is always available as its own peer, which rules out that relative input consumption exceeds unity. Calculating order-m efficiency requires choosing values for two parameters, D and m. While the choice of D is a pure matter of accuracy, where improving accuracy comes at the expense of prolonged computing time, the choice of m is critical. The smaller one makes the value for m, the larger is the share of superefficient DMUs. For $m \to \infty$, order-m coincides with FDH; for $m = N$, superefficient DMUs may still occur. Unlike FDH and order-α, there is no reference DMU for order-m that can serve as a unique[5] benchmark for DMU i. One may, nevertheless, determine a pseudo-reference DMU j_i^{pref} as

$$j_i^{\text{pref}} = \underset{j\in B_i}{\operatorname{argmin}} \left| \max_{k=1,\ldots,K}\left(\frac{x_{kj}}{x_{ki}}\right) - \widehat{\theta}_{mi}^{\text{OM}} \right|$$

2.3 Order-α efficiency

Order-α also generalizes FDH but in a different way. Rather than using minimum input consumption among the available peers as a benchmark, order-α uses the $(100-\alpha)$th percentile:

$$\widehat{\theta}_{\alpha i}^{\text{OA}} = \underset{j\in B_i}{\text{P}_{(100-\alpha)}}\left\{\max_{k=1,\ldots,K}\left(\frac{x_{kj}}{x_{ki}}\right)\right\} \tag{3}$$

When $\alpha = 100$, order-α coincides with FDH. When $\alpha < 100$, some DMUs may be classified as superefficient and not be enveloped by the estimated production-possibility frontier. Just like m for order-m efficiency, α can be regarded as a tuning parameter that determines the number of superefficient DMUs. Because calculating order-α efficiency scores does not involve a resampling procedure, $\widehat{\theta}_{\alpha i}^{\text{OA}}$ can be computed much faster than $\widehat{\theta}_{mi}^{\text{OM}}$.

5. For ties in the data, uniqueness may be violated for FDH and order-α.

2.4 Graphical illustration

Figures 1 and 2 provide a graphical illustration of the nonparametric frontier approaches discussed above. Figure 1 shows the generated[6] input use for 40 artificial DMUs, which are characterized by a two-inputs, single-output technology. The output level is uniform across all DMUs. Hence, the production-possibility frontier represents an isoquant. For 36 DMUs, the data are generated using a Cobb–Douglas technology with random excess use of inputs. The true Cobb–Douglas isoquant is displayed together with the artificial observations. For four DMUs, input consumption is inconsistent with this technology, exhibiting values that, according to the true frontier, are impossibly small. These DMUs represent outliers or, alternatively, observations that suffer from severe measurement error.

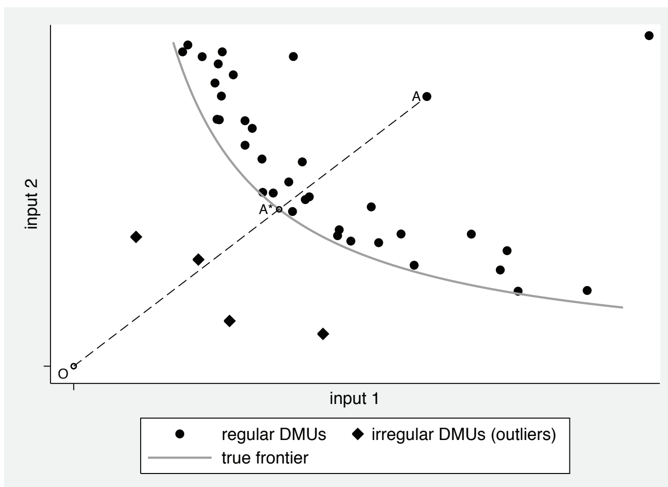

Figure 1. Scatterplot of input use and true production-possibility frontier (isoquant). *Source*: The author's own calculations based on artificial data.

In addition, figure 1 graphically illustrates the concept of input-oriented efficiency. Consider DMU A, for instance. Here the true efficiency score θ_A^{inp} can graphically be expressed as the ratio of two distances, $\overline{\text{O A*}}/\overline{\text{O A}}$, where O denotes the origin. Hypothetical efficient input consumption is related to its actually observed counterpart. Yet because the true frontier is typically unknown, an estimate is required.

Figure 2 displays the frontiers estimated by DEA, FDH, order-α, and order-m. The points at the estimated frontiers are constructed as observed input consumption scaled by the relevant estimated efficiency score. For DEA and FDH, the irregular DMUs span the estimated frontiers, rendering the rest of the DMUs highly inefficient. In fact, the regular observations do not at all affect the frontiers estimated by DEA and FDH. In contrast, order-α (α = 95) and order-m (m = 12) allow the abnormal DMUs to be

6. The Stata do-file used for generating the data and the figures is available upon request.

located outside the estimated production-possibility frontiers. To achieve this, order-α and order-m use the information on the regular DMUs for estimating the frontier, which in turn are compared with a more appropriate benchmark.

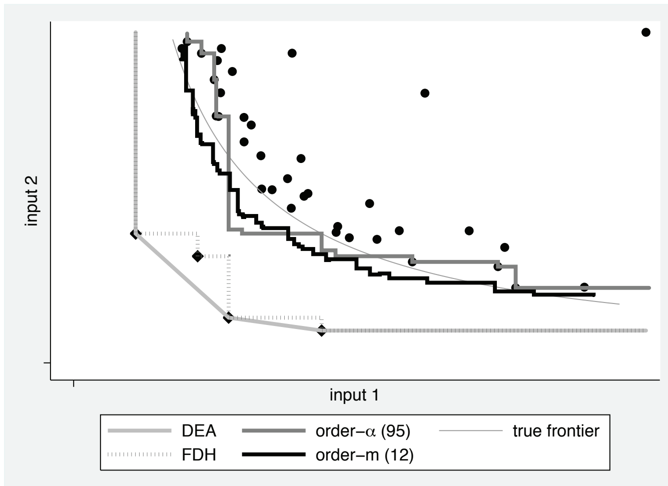

Figure 2. Nonparametrically estimated production-possibility frontiers (isoquants). *Source*: The author's own calculations based on artificial data.

2.5 Statistical inference

Bootstrapping allows for determining standard errors for efficiency scores obtained from nonparametric efficiency analysis. However, because of the boundary estimation nature of (full) frontier analysis, the naïve bootstrap does not yield as consistent an approximation of the desired sampling distribution. But subsampling bootstrapping, which is based on bootstrap samples smaller than N, is consistent for boundary estimation (Daraio and Simar 2007, 57). Standard errors provided by `orderalpha` and `orderm` are calculated using this method.[7] For relatively small values for α and m, the boundary nature of the estimation procedure vanishes, and one may use the naïve bootstrap instead. Because calculating order-m efficiency scores already involves a resampling procedure, bootstrapping `orderm` results in nested resampling, which—unless the sample is very small—requires an enormous amount of computing time.

2.6 Partial frontier–based outlier detection

Partial frontier analysis can be used for detecting potential outliers in data meant for subsequent nonparametric efficiency analysis by DEA or FDH (see Daraio and Simar

7. As suggested by Daraio and Simar (2007), the size of the bootstrap samples is determined as $\text{int}(N^t)$, with $0.5 \leq t \leq 1$.

[2007, 79]). The suggested approach rests on 1) carrying out a series of partial frontier analyses for different values of α or m; 2) plotting the share of superefficient DMUs against α or m; and 3) identifying discontinuities in the resulting curve. Such discontinuities point at those DMUs being outliers that are classified as superefficient for the corresponding values of α and m. This procedure may also be used for determining appropriate choices for α and m. The forthcoming[8] Stata command `oaoutlier` implements order-α-based outlier detection. Discussing `oaoutlier` in detail goes beyond the scope of this article.

3 The orderalpha and orderm commands

`orderalpha` and `orderm` require Stata 11.1 or higher. *weight*s and prefix commands such as `bootstrap`, `by`, and `svy` are not allowed. The number of DMUs is limited to the value of `matsize`. For `orderm`, the maximum allowed number of DMUs may further be reduced if bootstrapping is requested or a large value is specified for m.

3.1 Syntax for orderalpha

The syntax for the `orderalpha` command is

`orderalpha` *varlist1* `=` *varlist2* [*if*] [*in*] [`,` `dmu(`*varname*`)` `ort(input|output)` `alpha(#)` `bootstrap` `reps(#)` `tune(#)` `level(#)` `table(full|scores)` `dots(1|2)` `invert` `generate(`*newvarlist*`)` `replace` `nogenerate`]

where *varlist1* specifies inputs to production and *varlist2* specifies outputs from production. Both lists of variables must be mutually exclusive. At least one input variable and one output variable are required. Any variable in *varlist1* and *varlist2* needs to be numeric and strictly positive. DMUs with missing or nonpositive values in any input variable or output variable are dropped.

3.2 Options for orderalpha

`dmu(`*varname*`)` specifies an identifier for the considered DMUs. *varname* must uniquely identify DMUs. It may be either a numeric or a string variable. If no identifier is specified, the observation number *_n* is used. To make estimation results easily accessible and result tables informative, one should choose an informative variable name such as the real names of the DMUs.

`ort(input|output)` specifies whether `input` or `output` efficiency is computed. The default is `ort(input)`. For the former, inefficiency is defined in terms of possible proportional reduction in input consumption. For the latter, inefficiency is defined

8. A beta version of `oaoutlier` is available at http://www.stata.com/meeting/germany11/abstracts.html.

in terms of possible proportional increase in output generation. For `ort(input)`, efficiency scores are smaller than 1 for inefficient DMUs; for `ort(output)`, efficiency scores are greater than 1 for inefficient DMUs unless the `invert` option is specified. Efficient DMUs in either case are indicated by efficiency scores taking the value 1. Superefficient DMUs located beyond the estimated production-possibility frontier exhibit input-oriented efficiency greater than 1 and output-oriented efficiency smaller than unity.

`alpha(#)` specifies the #th percentile as benchmark. The default is `alpha(100)`, that is, FDH. Specified values smaller than unity are still interpreted in terms of percentiles, not quantiles. Values outside $(0, 100]$ are not allowed.

`bootstrap` invokes bootstrapping using 100 replications. If neither `bootstrap` nor `reps()` is specified, `orderalpha` does not compute standard errors for the estimated efficiency scores. The bootstrap will fail in determining nonzero standard errors for a DMU for which no (or only few) peers are available in the sample apart from the DMU itself. For large samples, bootstrapping generates a huge $N \times N$ variance–covariance matrix and requires substantial computing time, which quadratically increases in N.

`reps(#)` is equivalent to option `bootstrap`, except it allows for choosing the number of bootstrap replications.

`tune(#)` determines the size of the bootstrap samples as $\text{int}(N^{\#})$. Values within the $[0.5, 1]$ interval are allowed. Subsampling is applied to account for the naïve bootstrap being inconsistent in a boundary estimation framework. The boundary nature of the estimation problem vanishes as `alpha()` departs from 100. For values of `alpha()` substantially smaller than 100, one may apply the naïve bootstrap, `tune(1)`. For FDH, the specified value should be smaller than unity. The default is `tune(`$\{1 + \exp(50 - \alpha/2)\}/\{2 + \exp(50 - \alpha/2)\}$`)`. This is equal to 2/3 for FDH.

`level(#)` specifies the confidence level, as a percentage, for confidence intervals. The default is `level(95)` or as set by `set level`.

`table(full | scores)` invokes the display of a results table. For `table(scores)`, estimated efficiency scores are displayed as if they were regression coefficients. For `table(full)`, efficiency ranks and reference DMUs are also displayed. Displayed results are sorted by the values of *varname*. `orderalpha` may generate a huge table because N scores are computed. For this reason, suppressing table display is the default. `table(full)` is not allowed for $N > 2994$ and cannot be redisplayed by typing `orderalpha` without arguments.

`dots(1 | 2)` invokes a display of replication dots and loop dots. For `dots(1)`, one dot character is displayed for each bootstrap replication. For `dots(2)`, one dot character is also displayed for each DMU being analyzed. Type `2` dots are not displayed during bootstrap replications.

`invert` enables output-oriented efficiency to be reported analogously to input-oriented efficiency by taking the reciprocal: with `invert` specified, inefficient DMUs exhibit efficiency scores smaller than 1, regardless of how `ort()` is specified. `invert` has no effect on input-oriented efficiency.

`generate(`*newvarlist*`)` specifies the names of new variables containing estimation results. *newvarlist* may consist of up to three names. *newvar1* denotes estimated efficiency scores, *newvar2* denotes efficiency ranks, and *newvar3* denotes the reference DMU. If—because of ties in the data—more than one reference DMU is identified for some DMUs, further variables *newvar3_2*, *newvar3_3*, ... are created. If `generate()` is not specified or fewer than three names are assigned, the default names are `_oa_ort_alpha`, `_oarank_ort_alpha`, and `_oaref_ort_alpha`. For FDH, the default names are `_fdh_ort`, `_fdhrank_ort`, and `_fdhref_ort`.

`replace` specifies that existing variables named *newvar1*, *newvar2*, or *newvar3* be replaced.

`nogenerate` specifies that results not be saved to new variables.

3.3 Saved results for orderalpha

`orderalpha` saves the following in `e()`:

Scalars			
`e(N)`	number of observations	`e(super)`	share of superefficient DMUs
`e(alpha)`	value of `alpha()`	`e(mean_e)`	mean estimated efficiency
`e(inputs)`	number of inputs	`e(med_e)`	median estimated efficiency
`e(outputs)`	number of outputs	`e(level)`	confidence level
`e(efficient)`	share of efficient DMUs		
Macros			
`e(cmd)`	`orderalpha`	`e(table)`	`scores`, `full`, or `no`
`e(cmdline)`	command as typed	`e(invert)`	either `inverted` or `notinverted` (not saved for `ort(input)`)
`e(depvar)`	name of dependent variable		
`e(title)`	`Order-alpha efficiency analysis`	`e(ort)`	either `input` or `output`
		`e(outputlist)`	*varlist2* (list of outputs)
`e(dmuid)`	*varname* (name of DMU identifier)	`e(inputlist)`	*varlist1* (list of inputs)
		`e(properties)`	`b V`
`e(model)`	either `Order-alpha` or `FDH`		
`e(saved)`	names of new variables (not for option `nogenerate`)		
Matrices			
`e(b)`	coefficient vector	`e(reference)`	matrix of reference DMUs (not if *varname* is string)
`e(ranks)`	vector of efficiency ranks		
Functions			
`e(sample)`	marks estimation sample		

Further results are saved in `e()` if the option `bootstrap` or `reps()` is specified:

Scalars			
`e(N_reps)`	number of bootstrap replications	`e(tune)`	value of tuning parameter
		`e(N_bs)`	size of bootstrap samples
Macros			
`e(vce)`	`bootstrap`	`e(vcetype)`	`Bootstrap`
Matrices			
`e(V)`	variance–covariance matrix of the estimators	`e(reps)`	number of nonmissing results
		`e(b_bs)`	bootstrap estimates
`e(bias)`	estimated biases		

3.4 Syntax for orderm

The syntax for the `orderm` command is

`orderm` *varlist1* `=` *varlist2* [*if*] [*in*] [`, dmu(`*varname*`) ort(input|output)` `m(`#`) draws(`#`) bootstrap reps(`#`) tune(`#`) level(`#`)` `table(full|scores) dots(1|2) invert generate(`*newvarlist*`) replace` `nogenerate`]

The syntax for `orderm` differs from the syntax for `orderalpha` only by the options `m()` and `draws()`, which replace the option `alpha()`. Most of the options behave in the same way as they do in `orderalpha`; the options that behave differently are noted in the next section.

3.5 Special options for orderm

`m(`#`)` specifies the size of the artificial reference sample. The default is `m(`ceil($N^{2/3}$)`)`. Noninteger and nonpositive values are not allowed. Most applications choose values substantially smaller than N. Note: Even for `m(`N`)`, `orderm` does not yield results for FDH efficiency analysis. This requires `m()` to approach infinity. Yet rather than choosing a very large value for `m()`, one can carry out FDH efficiency analysis more efficiently by using `orderalpha`.

`draws(`#`)` specifies the number of resampling replications. The default is `draws(200)`, as suggested by Daraio and Simar (2007). Yet depending on the data, making estimated efficiency scores converge may require values that substantially exceed the default. Noninteger and nonpositive values are not allowed.

`bootstrap` invokes bootstrapping using 50 replications. Unless standard errors are definitely required, users are strongly advised not to request bootstrapping for large (and even moderately sized) samples. Because of nested resampling, computing time required by bootstrapping may become excessive. One may also consider `orderalpha` as an alternative. If neither `bootstrap` nor `reps()` is specified, `orderm` does not compute standard errors for the estimated efficiency scores. The bootstrap will fail in determining nonzero standard errors for a DMU for which no peers are available in the sample apart from the DMU itself.

`tune(#)` determines the size of the bootstrap samples as $\text{int}(N^{\#})$. Values within the $[0.5, 1]$ interval are allowed. Subsampling is applied to account for the naïve bootstrap being inconsistent in a boundary estimation framework. The boundary nature of the estimation problem vanishes as `m()` departs from infinity. For small values of `m()`, one may apply the naïve bootstrap, `tune(1)`. The default is `tune(`$\{2 + \exp(-m/N)\}/3$`)`, which is equal to 2/3 for FDH.

`generate(`*newvarlist*`)` specifies the names of new variables containing estimation results. *newvarlist* may consist of up to three names. *newvar1* denotes estimated efficiency scores, *newvar2* denotes efficiency ranks, and *newvar3* denotes the name of the pseudo-reference DMU. If—because of ties in the data—more than one pseudo-reference DMU is identified for some DMUs, further variables *newvar3_2*, *newvar3_3*, ... are created. If `generate(`*newvarlist*`)` is not specified or fewer than three names are assigned, the default names are `_om_ort_m`, `_omrank_ort_m`, and `_omref_ort_m`.

3.6 Saved results for orderm

Saved results for `orderm` are the same as they are for `orderalpha` except for the scalar `e(alpha)` that is not saved to `e()` and the following:

Scalars			
`e(m)`	value of `m()`	`e(draws)`	value of `draws()`
Macros			
`e(cmd)`	`orderm`	`e(model)`	`Order-m`
`e(title)`	`Order-m efficiency analysis`		

4 Examples for orderalpha and orderm

4.1 Basic syntax and FDH

We will use Stata's famous `auto.dta` example dataset for a simple example, only meant to illustrate the Stata commands. For real-data applications of partial frontier approaches, see, for example, Pilyavsky and Staat (2008), Cunha Marques and De Witte (2011), and Felder and Tauchmann (forthcoming).

We consider a car's repair record (`rep78`),[9] its headroom (`headroom`), and its trunk space (`trunk`) as outputs from the car's service production. Inputs are inverse mileage, that is, gallons per mile (`gpm`), weight (`weight`), length (`length`), and displacement (`displacement`).

Because a partial frontier analysis may involve resampling procedures, we first set the seed of the random-number generator to guarantee replicability. Confining the analysis to foreign cars, running the basic syntax of `orderalpha` yields some information on model specifications and descriptive statistics for input-oriented FDH efficiency scores.

```
. sysuse auto
(1978 Automobile Data)
. generate gpm = 1/mpg
. set seed 987654321
. orderalpha weight length displacement gpm = rep78 headroom trunk if foreign
FDH input-oriented efficiency scores estimated (variable _fdh_input)
Number of dmus                 = 21
Number of inputs               = 4
Number of outputs              = 3
Mean efficiency                = .9344
Median efficiency              = .9324
Share of efficient dmus        = .381
```

No DMU-level results are displayed, although they are saved to the data. To request a table of DMU-level results, we specify the options `table(full)` and `reps(200)`, where the latter requests bootstrapped standard errors. To make the results table informative, we use the string variable `make` as an identifier by specifying `dmu(make)`. The option `nogenerate` prevents Stata from resaving results to the data.

9. Because `rep78` is measured on an ordinal scale, it is ill-suited to enter in an efficiency analysis. But because this example is only for illustrating the syntax, we will ignore this caveat.

```
. orderalpha weight length displacement gpm = rep78 headroom trunk if foreign,
> dmu(make) reps(200) table(full) nogenerate

FDH input-oriented efficiency scores estimated (no variable saved)

Number of dmus                = 21
Number of inputs              = 4
Number of outputs             = 3
Mean efficiency               = .9344
Median efficiency             = .9324
Share of efficient dmus       = .381

     dmu (make)   Eff. Score   Std. Err.    z Stat.   Eff. Rank    Ref. DMU

      Audi 5000     .8201058    .0869334   2.069334          20   VW Diesel
       Audi Fox     .9323671    .0933896    .724201          11   VW Rabbit
       BMW 320i     .8757062    .2033932   .6111012          18   VW Diesel
     Datsun 200     .9058824    .0280815   3.351586          14   Mazda GLC
     Datsun 210            1    .2981914          0           1  Datsun 210
     Datsun 510     .9058824    .0624617   1.506806          14   Mazda GLC
     Datsun 810     .8369565    .0403014     4.0456          19   Mazda GLC
    Fiat Strada            1           .          .           1  Fiat Strad
   Honda Accord     .9107143    .2199731   .4058938          13   VW Diesel
    Honda Civic            1    .1142507          0           1  Honda Civi
      Mazda GLC            1           .          .           1   Mazda GLC
 Renault Le Car            1    .1531418          0           1  Renault Le
         Subaru      .995122    .4936048   .0098825           9   VW Diesel
  Toyota Celica     .8908046    .1553441   .7029261          16   VW Diesel
 Toyota Corolla     .9393939    .3038749   .1994441          10   VW Diesel
  Toyota Corona     .8857143    .0943252   1.211614          17   VW Diesel
      VW Dasher          .92    .2036947   .3927448          12   VW Rabbit
      VW Diesel            1    .6822315          0           1   VW Diesel
      VW Rabbit            1    .1951341          0           1   VW Rabbit
    VW Scirocco            1           .          .           1  VW Scirocc
      Volvo 260     .8031088    .0886606   2.220728          21   VW Diesel

Note: z-Statistic is abs(Eff.Score - 1)/Std.Err.
```

Standard errors are missing for three DMUs. For these, besides the DMU itself, no peers are available in the estimation sample. This makes the bootstrap fail in determining standard errors for these observations. If output-oriented efficiency is requested instead, the option `ort(output)` must be specified.

```
. orderalpha weight length displacement gpm = rep78 headroom trunk if foreign,
> dmu(make) ort(output) reps(200) table(full)
FDH output-oriented efficiency scores estimated (variable _fdh_output)
Number of dmus               = 21
Number of inputs             = 4
Number of outputs            = 3
Mean efficiency              = 1.06
Median efficiency            = 1
Share of efficient dmus      = .7143
```

dmu (make)	Eff. Score	Std. Err.	z Stat.	Eff. Rank	Ref. DMU
Audi 5000	1	.1123844	0	1	Audi 5000
Audi Fox	1.2	.1473373	1.35743	16	VW Diesel
BMW 320i	1.2	.135962	1.471	16	VW Diesel
Datsun 200	1.25	.1104965	2.262514	21	Datsun 210
Datsun 210	1	.	.	1	Datsun 210
Datsun 510	1.2	.1148945	1.740727	16	VW Diesel
Datsun 810	1.2	.0999874	2.000253	16	Honda Acco
Fiat Strada	1	.1710402	0	1	Fiat Strad
Honda Accord	1	.1079517	0	1	Honda Acco
Honda Civic	1	.	.	1	Honda Civi
Mazda GLC	1	.	.	1	Mazda GLC
Renault Le Car	1	.	.	1	Renault Le
Subaru	1	.	.	1	Subaru
Toyota Celica	1	.1142397	0	1	Toyota Cel
Toyota Corolla	1	.1180334	0	1	Toyota Cor
Toyota Corona	1	.0752533	0	1	Subaru
VW Dasher	1.2	.1563238	1.279396	16	VW Diesel
VW Diesel	1	.	.	1	VW Diesel
VW Rabbit	1	.1592883	0	1	VW Rabbit
VW Scirocco	1	.2130136	0	1	VW Scirocc
Volvo 260	1	.0897516	0	1	Audi 5000

```
Note: z-Statistic is abs(Eff.Score - 1)/Std.Err.
```

The above output indicates that the choice of input-oriented or output-oriented approach clearly makes a difference.

4.2 Order-α

The examples above all apply FDH efficiency, because `alpha()` is left unspecified and the default `alpha(100)` is used. To carry out a nondegenerated partial frontier order-α analysis, we choose the 90th percentile as the benchmark by specifying `alpha(90)`. Moreover, we assign our own names to the new generated variables.

```
. orderalpha weight length displacement gpm = rep78 headroom trunk if foreign,
> dmu(make) alpha(90) reps(200) table(full) generate(escore erank eref) replace
Order-alpha(90) input-oriented efficiency scores estimated (variable escore)
Number of dmus                 = 21
Number of inputs               = 4
Number of outputs              = 3
Mean efficiency                = .9421
Median efficiency              = .9394
Share of efficient dmus        = .3333
Share of super-efficient dmus  = .0476
```

dmu (make)	Eff. Score	Std. Err.	z Stat.	Eff. Rank	Ref. DMU
Audi 5000	.8201058	.081434	2.209081	20	VW Diesel
Audi Fox	.9565217	.0229673	1.893054	10	Mazda GLC
BMW 320i	.8757062	.0747637	1.66249	18	VW Diesel
Datsun 200	.9117647	.0140217	6.292751	13	VW Diesel
Datsun 210	1	.0933255	0	2	Datsun 210
Datsun 510	.9117647	.0308705	2.858237	13	VW Diesel
Datsun 810	.8423913	.0198544	7.938216	19	VW Diesel
Fiat Strada	1	.	.	2	Fiat Strad
Honda Accord	.9107143	.1729695	.5161934	15	VW Diesel
Honda Civic	1.12	.0597623	2.007952	1	VW Rabbit
Mazda GLC	1	.	.	2	Mazda GLC
Renault Le Car	1	.0694423	0	2	Renault Le
Subaru	.995122	.3621871	.0134683	9	VW Diesel
Toyota Celica	.8908046	.1177655	.9272276	16	VW Diesel
Toyota Corolla	.9393939	.1665016	.363997	11	VW Diesel
Toyota Corona	.8857143	.0450777	2.535306	17	VW Diesel
VW Dasher	.92	.0655968	1.219572	12	VW Rabbit
VW Diesel	1	.6390733	0	2	VW Diesel
VW Rabbit	1	.1328049	0	2	VW Rabbit
VW Scirocco	1	.	.	2	VW Scirocc
Volvo 260	.8031088	.0801618	2.456172	21	VW Diesel

```
Note: z-Statistic is abs(Eff.Score - 1)/Std.Err.
```

Here only the `Honda Civic` is classified as superefficient, while `Audi 5000` and `Volvo 260` perform worst. To determine whether the latter two are more or less equally inefficient or whether a statistically significant efficiency differential exists, one can use Stata's `test` command in the same way as for performing tests on regression coefficients. This also applies to `testnl`, `lincom`, and `nlcom`. If necessary, one must convert the names of DMUs provided by the identifier to Stata names when used with `test`:

```
. test _b[Audi_5000]-_b[Volvo_260]=0
 ( 1)  [make]Audi_5000 - [make]Volvo_260 = 0
           chi2(  1) =    0.24
         Prob > chi2 =    0.6260
```

4.3 Order-m

Finally, we also run `orderm` on the data, choosing a reference sample of size 16 by specifying `m(16)`. To improve accuracy, we request a large number of resampling replications with `d(1000)`. Because `orderm` requires substantial computing time (about 20 seconds for the present example), we specify neither `bootstrap` nor `reps()`, and we abstain from calculating standard errors. Instead, we specify `dots(2)` for our convenience.

```
. orderm weight length displacement gpm = rep78 headroom trunk if foreign,
> dmu(make) m(16) draws(1000) table(full) dots(2)
looping through data:
..................... 21
Order-m(16) input-oriented efficiency scores estimated (variable _om_input_16)
Number of dmus                = 21
Number of inputs              = 4
Number of outputs             = 3
Mean efficiency               = .9386
Median efficiency             = .9387
Share of efficient dmus       = .2381
Share of super-efficient dmus  = .1429
```

dmu (make)	Eff. Score	Std. Err.	z Stat.	Eff. Rank	Pseudo Ref
Audi 5000	.8201058	.	.	20	VW Diesel
Audi Fox	.9387439	.	.	11	VW Rabbit
BMW 320i	.8855254	.	.	18	VW Diesel
Datsun 200	.9097788	.	.	15	VW Diesel
Datsun 210	1.011718	.	.	2	Datsun 210
Datsun 510	.9109988	.	.	13	VW Diesel
Datsun 810	.8390924	.	.	19	Mazda GLC
Fiat Strada	1	.	.	4	Fiat Strad
Honda Accord	.9108928	.	.	14	VW Diesel
Honda Civic	1.03713	.	.	1	Honda Civi
Mazda GLC	1	.	.	4	Mazda GLC
Renault Le Car	1.00376	.	.	3	Renault Le
Subaru	.9953073	.	.	9	VW Diesel
Toyota Celica	.8916782	.	.	16	VW Diesel
Toyota Corolla	.9400606	.	.	10	VW Diesel
Toyota Corona	.88872	.	.	17	VW Diesel
VW Dasher	.9223844	.	.	12	VW Rabbit
VW Diesel	1	.	.	4	VW Diesel
VW Rabbit	1	.	.	4	VW Rabbit
VW Scirocco	1	.	.	4	VW Scirocc
Volvo 260	.8045222	.	.	21	VW Diesel

```
Note: no bootstrapping; no standard errors computed
```

Results are similar to those obtained from order-α (`90`), yet order-m (`16`) yields a larger share of superefficient DMUs.

5 Summary and conclusions

In this article, I introduced the new commands `orderalpha` and `orderm`, which implement nonparametric order-α, order-m, and FDH efficiency analysis in Stata. In addition to calculating point estimates, the commands accommodate subsampling bootstrap-based inference. Implementing partial frontier analysis may open up further areas of application to Stata: nonparametric efficiency analysis is frequently applied in many fields such as managerial economics and health economics. In this, the article complements the contribution of Ji and Lee (2010), who have already introduced data envelopment analysis to Stata.

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 (DFG). I am grateful to Peter Grösche, Hendrik Schmitz, Manuel Frondel, participants of the 2011 German Stata Users Group meeting, one anonymous referee, and H. Joseph Newton for many valuable comments.

7 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.

Aragon, Y., A. Daouia, and C. Thomas-Agnan. 2005. Nonparametric frontier estimation: A conditional quantile-based approach. *Econometric Theory* 21: 358–389.

Cazals, C., J.-P. Florens, and L. Simar. 2002. Nonparametric frontier estimation: A robust approach. *Journal of Econometrics* 106: 1–25.

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

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.

Cunha Marques, R., and K. De Witte. 2011. Is big better? On scale and scope economies in the Portuguese water sector. *Economic Modelling* 28: 1009–1016.

Daraio, C., and L. Simar. 2007. *Advanced Robust and Nonparametric Methods in Efficiency Analysis: Methodology and Applications*. New York: Springer.

Deprins, D., L. Simar, and H. Tulkens. 1984. Measuring labor-efficiency in post offices. In *The Performance of Public Enterprises: Concepts and Measurement*, ed. M. Marchand, P. Pestieau, and H. Tulkens, 243–267. Amsterdam: North-Holland.

Felder, S., and H. Tauchmann. Forthcoming. Federal state differentials in the efficiency of health production in Germany: An artifact of spatial dependence? *European Journal of Health Economics*.

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

Lovell, C. A. K., and A. P. B. Rouse. 2003. Equivalent standard DEA models to provide super-efficiency scores. *Journal of the Operational Research Society* 54: 101–108.

Pilyavsky, A., and M. Staat. 2008. Efficiency and productivity change in Ukrainian health care. *Journal of Productivity Analysis* 29: 143–154.

About the author

Harald Tauchmann is a senior researcher at Rheinisch-Westfälisches Institut für Wirtschaftsforschung (RWI) and a research group leader at the Centre of Health Economics Research (CINCH), both in Essen, Germany. His research interests include statistical methods, applied econometrics, and health economics.