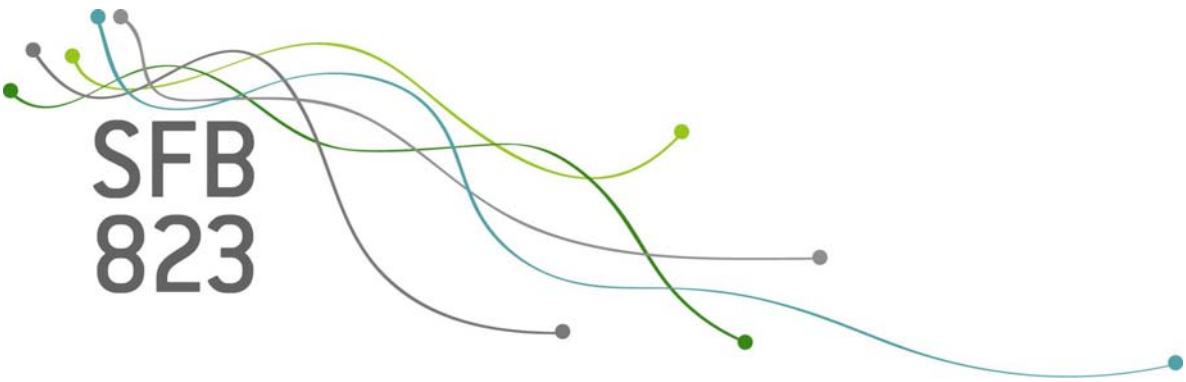
SFB 823

Partial frontier efficiency analysis for Stata

Harald Tauchmann

Nr. 25/2011

Discussion Paper



Partial frontier efficiency analysis for Stata

Harald Tauchmann
Rheinisch-Westfälisches Institut für Wirtschaftsforschung (RWI)
Hohenzollernstraße 1-3, 45128 Essen, Germany
harald.tauchmann@rwi-essen.de

Abstract. Despite its frequent use in applied work, nonparametric approaches to efficiency analysis, namely data envelopment analysis (DEA) and free disposal hull (FDH), have bad reputations among econometricians. This is mainly due to DEA and FDH representing deterministic approaches that are highly sensitive to outliers and measurement errors. However, recently, so-called partial frontier approaches namely order-m and order-α have been developed. They generalize FDH by allowing for super-efficient observations to be located beyond the estimated production-possibility frontier. Although these methods are purely non-parametric too, sensitivity to outliers is substantially reduced by partial frontier approaches enveloping just a sub-sample of observations. We introduce the new Stata commands `orderm` and `orderalpha` that implement order-m, order-α, and FDH efficiency analysis in Stata. The commands allow for several options, such as statistical inference based on sub-sampling bootstrap.

Keywords: orderalpha, orderm, non-parametric, efficiency, partial frontier, free disposal hull, outlier-robust, decision making unit.

1 Introduction

Countless empirical analyses address the efficiency of production units, which in this literature are frequently referred to as decision making units (DMUs). Two major methodical approaches to efficiency measurement exist: parametric and non-parametric ones. Among the former the most common are stochastic frontier models (Aigner et al. 1977), which augment a classical regression model by a non-positive error term capturing inefficiency in production. Stochastic frontier analysis is implemented in Stata by the `frontier` command. In contrast, non-parametric approaches, namely data envelopment analysis (DEA) introduced by Charnes et al. (1978) and the free disposal hull (FDH) introduced by Deprins et al. (1984), are not embedded in a regression framework familiar to econometricians. Rather, they are based on non-parametrically enveloping a given sample of data by a piecewise linear hull. While DEA assumes a convex technology and employs linear programming for enveloping the data, FDH is based an the principle of weak dominance and departs from the convexity assumption inherent to DEA. That is, FDH envelopes the data by a non-convex staircase-hull; see e.g. Cooper et al. (2007) for a comprehensive discussion of DEA and FDH. Data envelopment analysis 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 non-parametric approaches have intensely been debated. The former 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 only accommodating single-output technologies.[1] The latter have been criticized by econometricians for being deterministic approaches, lacking a well-defined data generating process, and, most relevant, for being extremely vulnerable to outliers and measurement error.

The final objection to non-parametric efficiency measurement has recently been addressed by so-called partial frontier approaches, namely order-m (Cazals et al. 2002) and order-α (Aragon et al. 2005) efficiency. These approaches generalize FDH by allowing for super-efficient observations to be located beyond the estimated production-possibility frontier. 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. The present paper contributes to non-parametric efficiency analysis by introducing the new Stata commands `orderm` and `orderalpha` that 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` by a simple example. Section 5 summarizes and concludes the article.

2 The concept of partial frontier analysis

Consider a sample of N decision making units. For each DMU $i = 1, \ldots, N$ 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. The prime objective of efficiency measurement is calculating an efficiency score θ_i for each DMU. Typically, two variants are considered: (i) input-oriented efficiency θ_i^{inp}, i.e. the factor by which input consumption of DMU i can proportionally be reduced leaving outputs unchanged, and (ii) output-oriented efficiency θ_i^{out}, i.e. the factor by which output generation can proportionally be increased leaving input consumption unchanged. Both 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 measured the relative radial distance in input-direction, output-oriented efficiency measures the relative radial distance in output direction.[2] For full frontier models for which all DMUs are enveloped by the production possibility frontier, $\theta_i^{inp} \in (0, 1]$ and $\theta_i^{out} \in [1, \infty)$ holds. That is, efficient DUMs are characterized by efficiency scores

1. This objection does not apply if a cost frontier – rather than a production frontier – is estimated.
2. Notwithstanding may consider other directions too, yet the above ones are most common. Note that for full frontier models (DEA, FDH) the estimated production possibility frontier is the same for input- and output-oriented efficiency. Nevertheless, DMUs which are located at the FDH-frontier – but not at one of its corners – are FDH-efficient only in terms of either output- or input-oriented efficiency; cf. the discussion about 'slack values' in the context of DEA. For partial frontier models the estimated frontier depends on direction.

taking the value of one, 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 one. In order to avoid redundancies, in the following, we focus on input-oriented efficiency. Yet, all arguments below analogously apply to output-oriented efficiency.

2.1 The Freed Disposal Hull

As partial frontier approaches generalize FDH, we first shortly discuss the latter. Here (input-oriented) efficiency is estimated by comparing each DMU $i = 1, \dots, N$ with all other DMUs $j = 1, \dots, N$ 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 reference to i, and $\hat{\theta}_i^{\text{FDH}}$ is calculated as relative input use:[3]

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

Decision making units that exhibit minimum input consumption among all their peers serve as their own reference. For these DMUs, which span the estimated production possibility frontier, $\hat{\theta}^{\text{FDH}}$ takes the value of one. Evidently, even a single DMU in the data that exhibits abnormally little – possibly misreported – input consumption renders all its peers inefficient. 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. That is, rather than benchmarking a DMU by the best performing peer in the sample at hand, order-m is based an the idea of benchmarking the DMU by expected best performance in sample of m peers. In computational terms order-m efficiency follows a four steps procedure (Daraio and Simar 2007, 72):

1. From B_i a sample of m peer DMUs is randomly drawn with replacement.
2. Pseudo FDH efficiency $\hat{\theta}_{mi}^{\widetilde{\text{FDH}}_d}$ is calculated using this artificial reference sample.
3. Steps 2 and 3 are repeated D times.
4. Order-m efficiency is calculated as the average of pseudo FDH scores:

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

3. Equation (1) focusses on calculating $\hat{\theta}_i^{\text{FDH}}$ from a given sample of data. This analogously applies to (2) and (4). For a more theory oriented coverage of FDH, order-m, and order-α, see Daraio and Simar (2007), pages 34, 68, and 72, respectively.

Not that due to random re-sampling, 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 one. That is, order-m allows for super-efficient DMUs that are located beyond the estimated production possibility frontier. This is the key difference from FDH, where a decision making unit 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 to the expense of prolonged computing time, the choice of m is critical. The smaller one chooses the value for m, the larger the share of super-efficient DMUs gets. For $m \to \infty$ order-m coincides with FDH, while for $m = N$ super-efficient DMUs will still occur. Unlike FDH and order-α, for order-m no reference DMU exists that serves as unique[4] benchmark for DMU i. One may, nevertheless, determine a pseudo-reference DMU j_i^{pref} as:

$$j_i^{\text{pref}} = \underset{j \in B_i}{\text{argmin}} \left| \max_{k=1,\ldots,K} \left\{ \frac{x_{kj}}{x_{ki}} \right\} - \hat{\theta}_{mi}^{\text{OM}} \right|. \tag{3}$$

2.3 Order-α efficiency

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

$$\hat{\theta}_{\alpha i}^{\text{OA}} = \text{P}_{\substack{(100-\alpha) \\ j \in B_i}} \left\{ \max_{k=1,\ldots,K} \left\{ \frac{x_{kj}}{x_{ki}} \right\} \right\} \tag{4}$$

For $\alpha = 100$ order-α coincides with FDH, while for $\alpha < 100$ some DMUs will be classified as super-efficient and not be enveloped by the estimated production possibility frontier. That is, just like m for order-m efficiency, α can be regarded as a tuning parameter that determines the number of super-efficient DMUs. Since calculating order-α efficiency scores does not involve a re-sampling procedure, $\hat{\theta}_{\alpha i}^{\text{OA}}$ can much faster be computed than $\hat{\theta}_{mi}^{\text{OM}}$.

2.4 Graphical illustration

Figures 1 and 2 provide a graphical illustration of the non-parametric frontier approaches discussed above. The former displays generated[5] input use for 40 artificial DMUs, which are characterized by a two-inputs, single-output technology. The output level is uniform across all decision making units. The production possibility frontier, hence, represent an isoquant. For 36 DMUs, the data is 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 which, according to the true frontier, are impossibly

4. For ties in the data, uniqueness may be violated for FDH and order-α.
5. The Stata do-file used for generating the data and the figure is available upon request.

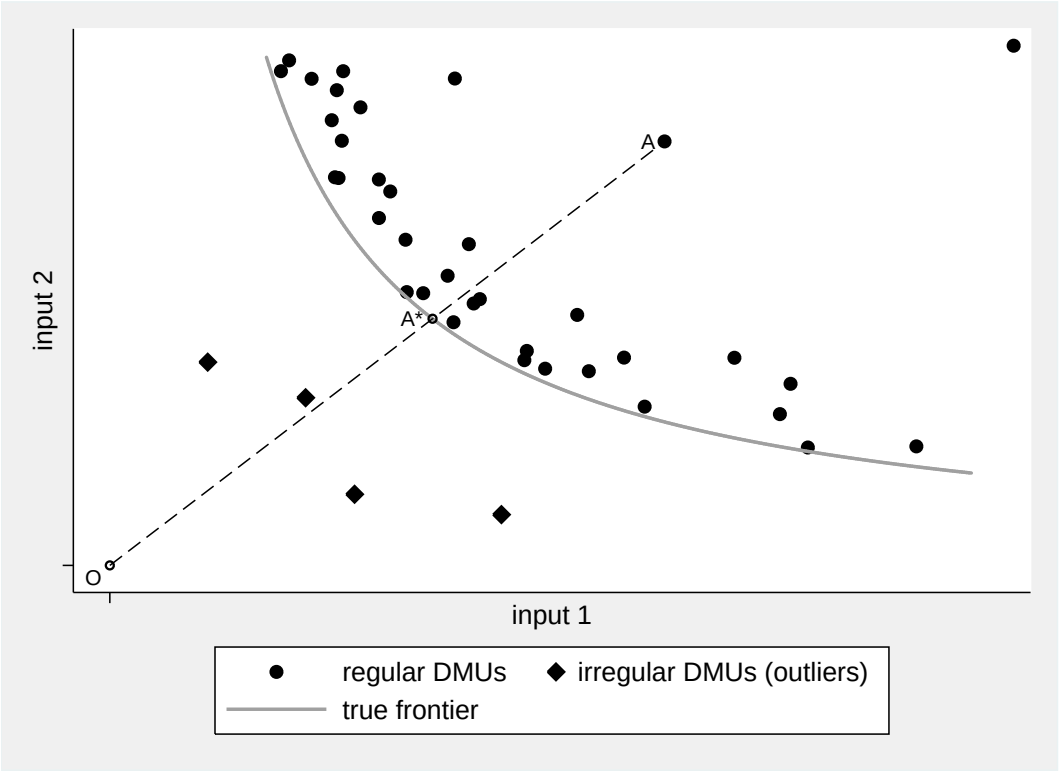


Figure 1: Scatter plot of input use and true production possibility frontier (isoquant). *Source*: Own calculations based on artificial data.

small. These DMUs represent outliers or, alternatively, observations that suffer from severe measurement error.

In addition, Figure 1 graphicly 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, i.e. $\overline{\mathrm{O\,A^*}}/\overline{\mathrm{O\,A}}$, where O denotes the origin. That is, hypothetical efficient input consumption is related to its actually observed counterpart. Yet, as 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 all 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 ($\alpha = 95$) and order-m ($m = 12$) allow the abnormal DMUs to be located outside of the estimated production possibility frontiers. By this, order-α and order-m use the information on the regular DMUs for estimating the frontier, which in turn are compared to more appropriate benchmark.

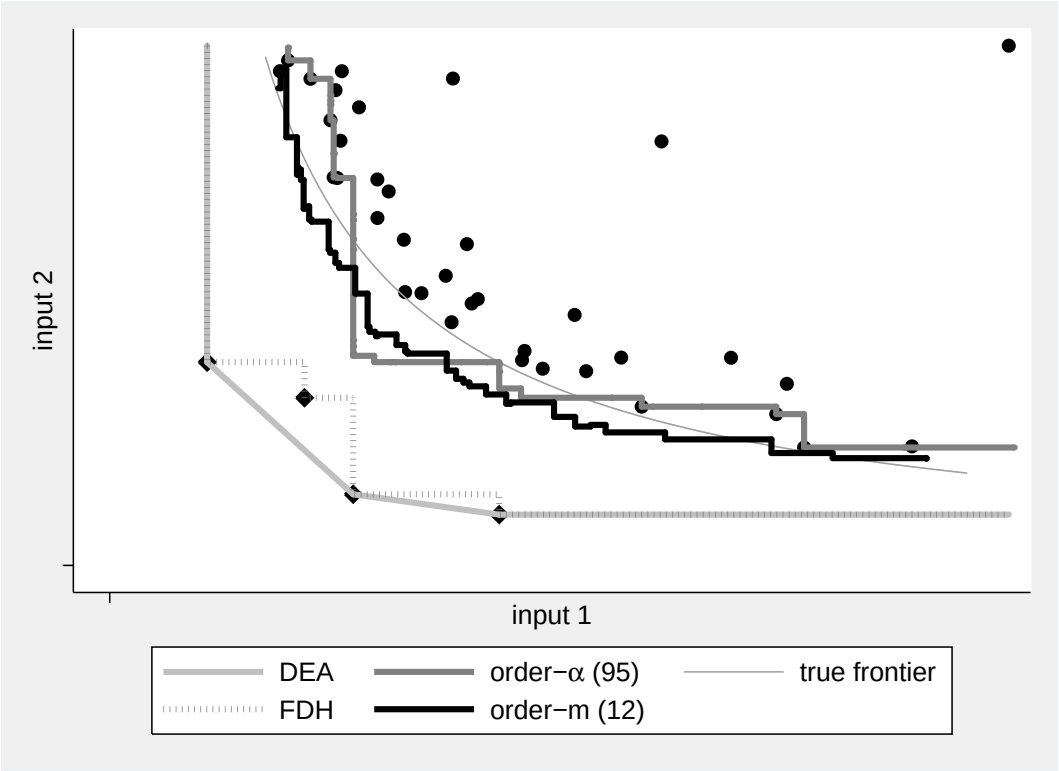


Figure 2: Non-parametrically estimated production possibility frontiers (isoquants). *Source*: Own calculations based on artificial data.

2.5 Statistical inference

Bootstrapping allows for determining standard errors for efficiency scores obtained from non-parametric efficiency analysis. However, due to the boundary estimation nature of (full) frontier analysis, the naive bootstrap does not yield as consistent approximation of the desired sampling distribution. Yet, sub-sampling 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. For relatively small values for α and m, respectively, the boundary nature of the estimation procedure vanishes and one may use the naive bootstrap instead. As calculating order-m efficiency scores already involves a re-sampling procedure, bootstrapping `orderm` results in nested re-sampling 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 non-parametric efficiency analysis by DEA or FDH; see Daraio and Simar (2007, 79). The suggested approach rests on (i) carrying out a series of partial frontier analyses for different values of α or m, (ii) plotting the share of super-efficient DMUs against α or m, respectively, and (iii) identifying discontinuities in the resulting curve. Such discontinuities which point at those DMUs being outliers that are classified as super-efficient for the corresponding values of α and m, respectively. This procedure may also be used for determining appropriate choices for α and m. The forthcoming[6] Stata command `oaoutlier` implements order-α based outlier-detection. Yet, discussing `oaoutlier` in detail goes beyond the scope of this article.

3 The orderalpha and orderm commands

`orderalpha` and `orderm` require Stata 11 or higher. `weights` and prefix commands such as `bootstrap`, `by`, and `svy` are not allowed. The number of DUMs 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 `orderalpha` reads as follows:

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

where *varname* is an identifier that must uniquely identify DUMs. It may be either a numeric or a string variable. *varlist1* and *varlist2* specify inputs and outputs, respectively. Both lists of variables must be mutually exclusive. At least one input-variable and one output-variable is required. Any variable in *varlist1* and *varlist2* needs to be numeric and strictly positive. DMUs with missing or non-positive values in any input-variable or output-variable are dropped.

3.2 Options for orderalpha

`ort(input|output)` specifies whether input- or output-oriented efficiency is considered, where input-oriented efficiency is the default.

`alpha(`#`)` specifies the `alpha()`th percentile as benchmark. The default is `alpha(100)`

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

that is FDH. Note that 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 non-zero SEs for DMUs, for which no (or only few) peers are available in the sample, apart from the DUM 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`, besides allowing for choosing the number of bootstrap replications.

`tune(#)` determines the size of the bootstrap samples as $\text{int}(N^{\texttt{tune()}})$. The default value is $(1 + \exp(50 - \alpha/2))/(2 + \exp(50 - \alpha/2))$ that is 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 displaying a results table. For `table(scores)` estimated efficiency scores are displayed as if thy 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 as N scores are computed. For this reason, suppressing table display is the default. `table(full)` is not allowed for $N > 2994$ and cannot be re-displayed by typing the command without arguments.

`dots(1|2)` invokes displaying 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` makes output-oriented efficiency being reported analogously to input-oriented efficiency by taking the reciprocal. That is, with `invert` specified, inefficient DMUs exhibit efficiency scores smaller than one, irrespective of how `ort()` is specified. `invert` has no effect on input-oriented efficiency.

`generate(`*newvarlist*`)` specifies the names of a 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 DUMs. If – because of ties in the data – for some DMUs more than one reference DMU is identified, further variables *newvar3_2*, *newvar3_3*, ... are created. If `generate()` is not specified or less than three names are assigned, default names are *_oa*`_ort_alpha`, *_oarank*`_ort_alpha`, and *_oaref*`_ort_alpha`. For FDH default names are *_fdh*`_ort`, *_fdhrank*`_ort`, and *_fdhref*`_ort`.

`replace` specifies that existing variables named *newvar1*, *newvar2*, and *newvar3* are replaced.

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

3.3 Saved results for orderalpha

`orderalpha` saves the following results to `e()`:

Scalars

`e(N)`	number of DMUs	`e(super)`	share of super-efficient 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(med_e)`	median estimated efficiency
`e(efficient)`	share of efficient DMUs	`e(level)`	confidence level

Macros

`e(cmd)`	`orderalpha`	`e(table)`	`scores`, `full`, or `no`
`e(cmdline)`	command as typed	`e(invert)`	either `inverted` or `notinverted` (not for `ort(input)`)
`e(title)`	Order-alpha efficiency analysis	`e(ort)`	either `input` or `output`
`e(dmuid)`	*varname*	`e(properties)`	either `b` or `b V`
`e(model)`	either Order-alpha or FDH	`e(depvar)`	dmu
`e(saved)`	names of new variables (not for option `nogenerate`)		

Matrices

`e(b)`	vector of efficiency scores	`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 `boot` or `reps()` is specified:

Scalars

`e(N_reps)`	number of bootstrap repetitions	`e(N_bs)`	size of bootstrap samples
`e(tune)`	value of `alpha()`		

Macros

`e(vce)`	bootstrap	`e(vcetype)`	Bootstrap

Matrices

`e(b)`	variance-covariance matrix	`e(reps)`	number of non-missing results
`e(bias)`	estimated biases	`e(b_bs)`	bootstrap estimates

3.4 Syntax for orderm

The syntax for `orderm` reads as follows:

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

That is, the syntax for `orderm` differs from the syntax for `orderalpha` only by the options `m()` and `draws()`, which replace the option `alpha()`.

3.5 Special options for orderm

`m(#)` specifies the size of the artificial reference sample. The default is ceil($N^{2/3}$). Non-integer and non-positive values are not allowed. Most applications choose values substantially smaller than N. Though `orderm` coincides with FDH for large values of `m`, FDH efficiency analysis can by carried out more efficiently using `orderalpha`.

`draws(#)` specifies the number of re-sampling replications. The default is `draws(200)`, as suggested by Daraio and Simar (2007). Yet, depending an the data at hand, making estimated efficiency scores converge may require values that substantially exceed the default. Non-integer and non-positive values are not allowed.

`tune(#)` *as above ...* . The default value is $(2 + \exp(-m/N))/3$ that is 2/3 for FDH.

`generate(`*newvarlist*`)` *as above ... newvar3* denotes the name of the pdeudo-reference DUM. ... 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 above, except for the scalar `e(alpha)` that is not saved to `e()` and:

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 use Stata's famous `auto.dta` example dataset for a simply example, which is only meant for illustrating the Stata commands. For serious real data applications of partial frontier approaches see, for example Pilyavsky and Staat (2008) and Binder and Broekel (2008). In the present example, the string variable *make* serves as identifier. We consider a cars' repair record (*rep78*)[7], its headroom (*headroom*), and its trunk space (*trunk*) as outputs from the cars' service production. Inputs are inverse milage, i.e. gallons per mile (*gpm*), weight (*weight*), length (*length*), and displacement (*displacement*). As partial frontier analysis may involve re-sampling procedures, we firstly set the see of the random number generator to guarantee replicability. Confining us to foreign cars, running the basis syntax of `orderalpha` yields some information on model specifications and descriptive statistics for input-oriented FDH efficiency scores.

(*Continued on next page*)

7. Since *rep78* is measured on an ordinal scale, it is ill-suited for entering an efficiency analysis. Yet, as the example is only for illustrating the syntax, we ignore this caveat.

```
. sysuse auto.dta, clear
(1978 Automobile Data)
. gen gpm = 1/mpg
. set seed 987654321
. orderalpha make if foreign, inp(weight length displacement gpm) out(rep78 hea
> droom trunk)
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
```

Yet, no DUM-level results are displayed, which, nevertheless, 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. The option `nog` prevents Stata from re-saving results to the data.

```
. orderalpha make if foreign, inp(weight length displacement gpm) out(rep78 hea
> droom trunk) reps(200) tab(full) nog
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.

If output-oriented efficiency is requested instead, the option `ort(output)` has to be specified.

```
. orderalpha make if foreign, inp(weight length displacement gpm) out(rep78 hea
> droom trunk) ort(output) reps(200) tab(full) nog

FDH output-oriented efficiency scores generated (no variable saved)

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 either choosing the input-oriented or the output-oriented approach clearly makes a difference.

4.2 Order-α

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

```
. orderalpha make if foreign, inp(weight length displacement gpm) out(rep78 hea
> droom trunk) alp(90) reps(200) tab(full) gen(escore erank eref) replace
```

(*Continued on next page*)

```
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 a super-efficient, while *Audi 5000* and *Volvo 260* perform worst. In order to determine whether the latter two are more or less equally inefficient, or whether a statistical significant efficiency differential exists, on 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`. Yet, if necessary, one has to convert DMUs names 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)`. In order to improve accuracy we request a large number of re-sampling replications with `d(1000)`. As `orderm` requires substantial computing time, we neither

specify `bootstrap` nor `reps()` and abstain from calculating standard errors.

```
. orderm make if foreign, inp(weight length displacement gpm) out(rep78 headroo
> m trunk) m(16) d(1000) tab(full)
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 super-efficiency DUMs.

5 Summary and conclusions

In this article the new commands `orderalpha` and `orderm` were introduced that implement non-parametric order-α, order-m, and FDH efficiency analysis in Stata. Besides calculating point estimates, the commands also accommodate sub-sampling bootstrap based inference. Implementing partial frontier analysis may open up further areas of application to Stata, as non-parametric efficiency analysis is frequently applied in many fields such es managerial economics und health economics. In this, the present article complements to Ji and Lee (2010), who have already introduced data envelopment analysis to Stata.

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 author is grateful to Peter Grösche, Hendrik Schmitz, Manuel Frondel, and participants of the 2011 German Stata Users Group meeting for many valuable comments. This article is meant for being submitted to the Stata Journal.

7 References

Aigner, D., 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 quantilebased approach. *Econometric Theory* 21: 358–389.

Binder, M., and T. Broekel. 2008. Conversion Efficiency as a Complementing Measure of Welfare in Capability Space. MPRA Paper 7583, University Library of Munich, Germany.

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 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.

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 laborefficiency in post offices. In *The Performance of Public Enterprises: Concepts and Measurement*, ed. M. Marchand, P. Pestieau, and H. Tulkens, 243–267. Amsterdam: Elsevier.

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

Pilyavsky, A., and M. Staat. 2008. Efficiency and Productivity Change in Ukrainian Health Care. *Journal of Productivity Analysis* 29: 143–154.