I.S.S.N: 1885-6888

ECONOMIC ANALYSIS

WORKING PAPER SERIES

A Data Envelopment Analysis Toolbox for MATLAB

♦

Inmaculada C. Álvarez, Javier Barbero and José L. Zofío

Working Paper 3/2016

UAM
UNIVERSIDAD AUTONOMA
DE MADRID

DEPARTAMENTO DE ANÁLISIS ECONÓMICO:
TEORÍA ECONÓMICA E HISTORIA ECONÓMICA

A Data Envelopment Analysis Toolbox for MATLAB

Inmaculada C. Álvarez **Javier Barbero** **José L. Zofío**
Universidad Autónoma de Madrid

Abstract

Data Envelopment Analysis Toolbox is a new package for MATLAB that includes functions to calculate the main DEA models. The package includes code for the standard additive and radial input and output measures, allowing for constant and variable returns to scale, as well as recent developments related to the directional distance function, and including both desirable and undesirable outputs when measuring efficiency and productivity; i.e., Malmquist and Malmquist-Luenberger indices. Bootstrapping to perform statistical analysis is also included. This paper describes the methodology and implementation of the functions and reports numerical results with well-known examples to illustrate their use.

Keywords: Data Envelopment Analysis, distance functions, technical efficiency, MATLAB

JEL codes: D20, 24, C44 .

Affiliation:

Inmaculada C.Álvarez, Javier Barbero, José L. Zofío
Department of Economics
Universidad Autónoma de Madrid
28049 Madrid, Spain
E-mail: inmaculada.alvarez@uam.es, javier.barbero@uam.es, jose.zofio@uam.es
URL: http://www.deatoolbox.com/

1. Introduction

Data Envelopment Analysis, DEA, has grown in importance over the past decades due to increase in the availability of data related to the performance of Decision Making Units, DMUs, regardless their market, governmental, or non-for-profit orientation. Basic DEA methods are included in some standard software packages used by econometricians (e.g., Stata, StataCorp (2015) with the user-written command by Ji and Lee (2010); LIMDEP, Econometric Software (2009)); available in dedicated non-commercial software accompanying academic handbooks–Cooper, Seiford, and Tone (2007), Wilson (2008), Bogetoft and Otto (2011) (these latter two implemented in R); commercial software–including trials versions, Emrouznejad and Cabanda (2014); free-ware programs–Sheel (2000); and even tutorials for spreadsheets, Sherman and Zhu (2006). Earlier versions of these programs have been reviewed, among others, by Hollingsworth (2004) and Barr (2004), and there are continuous proposals expanding these options. While these software packages implement the main DEA models, there is a lack of a full set of functions for MATLAB (The MathWorks, Inc. 2015), including some recent theoretical contributions that are missed in the existing software.

The **Data Envelopment Analysis Toolbox** introduces such set of functions, covering a wide range of efficiency and productivity models, and reporting numerical results based on classical examples presented in the literature. **Data Envelopment Analysis Toolbox** is available as free software, under the GNU General Public License version 3, and can be downloaded from `http://www.deatoolbox.com`, with all the supplementary material (data, examples and source code) to replicate all the results presented in this paper. The toolbox is also hosted on an open source repository on GitHub.[1]

The paper is organized as follows. The following section presents the data structures characterizing the production possibility sets, the structure of the functions, results, etc... Section 3 covers the standard DEA models introduced by Charnes, Cooper, and Rhodes (1978) and Banker, Charnes, and Cooper (1984) corresponding to the radial input and output efficiency measures, allowing for constant and variable returns to scale, as well as newer proposals based on the flexible directional distance function. The non-oriented additive model is also presented as well as the super-efficiency model for all the previous efficiency measures. Malmquist productivity indices and their decomposition into efficiency change and technical change are shown in section 4, while section 5 deals with the measurement of economic efficiency, and its decomposition into technical and allocative terms. Section 6 is devoted to the measurement of efficiency with undesirable outputs, most notably environmental efficiency, followed by Section 7 presenting the Malmquist-Luenberger index. Statistical analyses and hypotheses testing using bootstrapping techniques are presented in Section 8. Advanced options, including displaying and exporting results can be found in Section 9. Section 10 concludes.

2. Data structures

Data Envelopment Analysis measures productive and economic performance of a set of $j = 1, 2, \ldots, n$ observed DMUs (firms, activities, countries, individuals, etc.). These observations transform a vector of $i = 1, 2, \ldots, m$ inputs $\mathbf{x} \in \mathbb{R}^m_{++}$ into a vector of $i = 1, 2, \ldots, s$ outputs $\mathbf{y} \in \mathbb{R}^s_{++}$ using the technology represented by the following constant returns to scale production possibility set: $P_{\text{CRS}} = \{(\mathbf{x}, \mathbf{y}) \ |\mathbf{x} \geqslant X\lambda,\ \mathbf{y} \leqslant Y\lambda,\ \lambda \geqslant \mathbf{0}\}$, where $X = (\mathbf{x})_j \in \mathbb{R}^{s\times n}, Y =$

[1] The address of the repository is `https://github.com/javierbarbero/DEAMATLAB`

$(\mathbf{y})_j \in \mathbb{R}^{m\times n}$ and $\lambda = (\lambda_1, \ldots, \lambda_n)^T$ is a semipositive vector.

Data are managed as regular MATLAB vectors and matrices, constituting the inputs of the estimation functions. All estimation functions return a structure `deaout` that contains fields with the estimation results as well as the input of the estimation function. Fields can be accessed directly using the dot notation and the whole structure can be used as an input to other functions that print or export results (e.g., `deadisp`).

Some of the fields of the `deaout` structure are the following:[2]

- `X`, `Y` and `Yu`: contain the inputs, outputs and undesirable outputs variables, respectively.
- `n` and `neval`: number of DMUs, and number of evaluated DMUs.
- `m`, `s` and `r`: number of inputs, outputs and undesirable outputs.
- `model`, `orient`, `rts`: strings containing the model type, the orientation, and the returns to scale assumption.
- `eff`: computed efficiency measure.
- `slackX`, `slackY`, `slackYu`: computed input, output and undesirable output slacks.
- `names`: names of the DMUs.

3. Basic DEA models

3.1. Radial input oriented model: Constant and variable returns to scale

Based on the data matrix (X, Y), we measure the input oriented efficiency of each observation o by solving n times the following linear programming problem–known as the CCR model:[3]

$$\begin{aligned}
\min_{\theta,\lambda} \quad & \theta \\
\text{subject to} \quad & \\
& \theta \mathbf{x}_o \geq X\lambda \\
& Y\lambda \geq \mathbf{y}_o \\
& \lambda \geq \mathbf{0}.
\end{aligned} \tag{1}$$

The optimal solution to this program—characterizing a technology with constant returns to scale, is denoted by θ^*_{CRS}. The constraints require the observation $(\theta_{\text{CRS}}\mathbf{x}_o, \mathbf{y}_o)$ to belong to P_{CRS}, while the objective seeks the minimum θ_{CRS} that reduces the input vector $\mathbf{x}_o$ radially to $\theta_{\text{CRS}}\mathbf{x}_o$ while remaining in P_{CRS}. A feasible solution signaling radial efficiency is $\theta^*_{\text{CRS}} = 1$.

[2]For a full list see the help of the function typing `help deaout` in MATLAB.

[3]This program corresponds to the so-called "envelopment form" of the formulation introduced by Charnes *et al.* (1978). Correspondences with the dual approaches ("multipliers form") can be found in Cooper *et al.* (2007).

Therefore if $\theta^*_{\text{CRS}} < 1$, the observation is *radially* inefficient and $(\lambda X, \lambda Y)$ outperforms $(\mathbf{x}_o, \mathbf{y}_o)$. With regard to this property, we define the additional input excesses and outputs shortfalls by the following slack vectors: $\mathbf{s}^- \in \mathbb{R}^m$ and $\mathbf{s}^+ \in \mathbb{R}^s$, respectively. Therefore: $\mathbf{s}^- = \theta^*_{\text{CRS}}\mathbf{x}_o - X\lambda$, and $\mathbf{s}^+ = Y\lambda - \mathbf{y}_o$ with $\mathbf{s}^- \geq 0$ and $\mathbf{s}^+ \geq 0$ for any feasible solution (θ, λ).

To obtain the possible input excesses and output shortfalls, the following second stage program that incorporates the optimal value θ^*_{CRS} and corrects radial inefficiency is solved:

$$
\begin{aligned}
\max_{\lambda, \mathbf{s}^-, \mathbf{s}^+} \quad & \omega = \mathbf{e}\mathbf{s}^- + \mathbf{e}\mathbf{s}^+ \\
\text{subject to} \quad & \\
& \mathbf{s}^- = \theta^*_{\text{CRS}}\mathbf{x}_o - X\lambda \\
& \mathbf{s}^+ = Y\lambda - \mathbf{y}_o \\
& \lambda \geq \mathbf{0},\ \mathbf{s}^- \geq 0, \mathbf{s}^+ \geq 0,
\end{aligned}
\tag{2}
$$

where $\mathbf{e} = (1, \ldots, 1)^T$ so $\mathbf{e}\mathbf{s}^- = \sum_{i=1}^{m} \mathbf{s}_i^-$ and $\mathbf{e}\mathbf{s}^+ = \sum_{i=1}^{s} \mathbf{s}_i^+$.

As a result, an observation is *technically* efficient if the optimal solution $(\theta^*_{\text{CRS}}, \lambda^*, \mathbf{s}^{-*}, \mathbf{s}^{+*})$ of the two above programs satisfy $\theta^*_{\text{CRS}} = 1, \mathbf{s}^{-*} = 0$, and $\mathbf{s}^{+*} = 0$, so no equiproportional contraction of inputs, and individual inputs reductions and outputs increases are possible (Pareto-Koopmans Efficiency).[4]

The measurement of technical efficiency assuming variables returns to scale as introduced by Banker *et al.* (1984)–known as the BCC model, considers the following production possibility set $P_{\text{VRS}} = \{(\mathbf{x}, \mathbf{y})\ |\mathbf{x} \geqslant X\lambda,\ \mathbf{y} \leqslant Y\lambda,\ \mathbf{e}\lambda = \mathbf{1},\ \lambda \geqslant \mathbf{0}.\}$. Therefore, the only difference with the CRS model is the adjunction of the condition $\sum_{j=1}^{n} \lambda_j = 1$. Calculating the VRS efficiency, along with the subsequent second stage program analogous to (2), yields the corresponding optimal solution $(\theta^*_{\text{VRS}}, \lambda^*, \mathbf{s}^{-*}, \mathbf{s}^{+*})$. As before, an observation is efficient with respect to the VRS technology if the optimal solution of the two programs satisfy $\theta^*_{\text{VRS}} = 1, \mathbf{s}^{-*} = 0$, and $\mathbf{s}^{+*} = 0$.

Finally, the scale efficiency of each observation is calculated as the ratio of the VRS to the CRS scores: $SE = \theta^*_{\text{CRS}}/\theta^*_{\text{VRS}}$. As a result the radial technical efficiency of an observation can be decomposed into its variable returns to scale efficiency ("pure" technical efficiency, PTE) and scale efficiency: $TE = \theta^*_{\text{CRS}} = PTE \times SE = \theta^*_{\text{VRS}} \times SE$.

The radial input oriented model can be computed in MATLAB using the `dea(X, Y, ...)` function with the `orient` parameter set to `io` (input oriented). The returns to scale assumption can be specified by setting the `rts` parameter to `crs` (constant returns to scale; default) or `vrs` (variable returns to scale). Results are returned in a `deaout` structure and can be accessed directly (see Section 2) or displayed using the `deadisp` function.[5]

```
> load 'deadataFLS'
```

[4]It is possible to solve both programs in a single stage formulation employing a "non-Archimedian" infinitesimal constant ϵ, e.g., (Fried, Lovell, and Schmidt 1993, 140). However, this may result in computational inaccuracies and erroneous results.

[5]See Section 9.3 for advanced uses of the `deadisp` function. The example for the input, output, directional and additive models corresponds to (Fried *et al.* 1993, 122).

```
> io = dea(X, Y, 'orient', 'io');
> deadisp(io);

_______________________________
Data Envelopment Analysis (DEA)

DMUs: 11
Inputs: 2     Outputs: 1
Model: radial
Orientation: io (Input oriented)
Returns to scale: crs (Constant)

--------------------------------------------------------------------
 DMU|      X1|      X2|       Y|   Theta| slackX1| slackX2|  slackY|
--------------------------------------------------------------------
   1|  5.0000| 13.0000| 12.0000|  1.0000|  0.0000|  0.0000|  0.0000|
   2| 16.0000| 12.0000| 14.0000|  0.6223|  0.0000|  0.0000|  0.0000|
   3| 16.0000| 26.0000| 25.0000|  0.8199|  0.0000|  0.0000|  0.0000|
   4| 17.0000| 15.0000| 26.0000|  1.0000|  0.0000|  0.0000|  0.0000|
   5| 18.0000| 14.0000|  8.0000|  0.3104|  0.0000|  0.0000|  0.0000|
   6| 23.0000|  6.0000|  9.0000|  0.5556|  4.4444|  0.0000|  0.0000|
   7| 25.0000| 10.0000| 27.0000|  1.0000|  0.0000|  0.0000|  0.0000|
   8| 27.0000| 22.0000| 30.0000|  0.7577|  0.0000|  0.0000|  0.0000|
   9| 37.0000| 14.0000| 31.0000|  0.8201|  1.6402|  0.0000|  0.0000|
  10| 42.0000| 25.0000| 26.5000|  0.5000|  0.0000|  0.0000|  0.0000|
  11|  5.0000| 17.0000| 12.0000|  1.0000|  0.0000|  4.0000|  0.0000|
--------------------------------------------------------------------

> io_vrs = dea(X, Y, 'orient', 'io', 'rts', 'vrs');
> deadisp(io_vrs);

_______________________________
Data Envelopment Analysis (DEA)

DMUs: 11
Inputs: 2     Outputs: 1
Model: radial
Orientation: io (Input oriented)
Returns to scale: vrs (Variable)

--------------------------------------------------------------------
 DMU|      X1|      X2|       Y|   Theta| slackX1| slackX2|  slackY|
--------------------------------------------------------------------
   1|  5.0000| 13.0000| 12.0000|  1.0000|  0.0000|  0.0000|  0.0000|
   2| 16.0000| 12.0000| 14.0000|  0.8700|  0.0000|  0.0000|  0.0000|
   3| 16.0000| 26.0000| 25.0000|  1.0000|  0.0000|  0.0000|  0.0000|
   4| 17.0000| 15.0000| 26.0000|  1.0000|  0.0000|  0.0000|  0.0000|
```

```
   5| 18.0000| 14.0000|  8.0000|  0.7116|  0.0000|  0.0000|  2.6984|
   6| 23.0000|  6.0000|  9.0000|  1.0000|  0.0000|  0.0000|  0.0000|
   7| 25.0000| 10.0000| 27.0000|  1.0000|  0.0000|  0.0000|  0.0000|
   8| 27.0000| 22.0000| 30.0000|  1.0000|  0.0000|  0.0000|  0.0000|
   9| 37.0000| 14.0000| 31.0000|  1.0000|  0.0000|  0.0000|  0.0000|
  10| 42.0000| 25.0000| 26.5000|  0.5000|  0.0000|  0.0000|  0.0000|
  11|  5.0000| 17.0000| 12.0000|  1.0000|  0.0000|  4.0000|  0.0000|
---------------------------------------------------------------------
```

The scale efficiency can be calculated using the `deascaleeff(X, Y, ...)` function. The function parameters are the same as those of the `dea` function, although the `rts` parameter specified will be omitted since both are needed in order to compute scale efficiency.

```
> io_scale = deascale(X, Y, 'orient', 'io');
> deadisp(io_scale);
```

```
-------------------------------
Data Envelopment Analysis (DEA)

DMUs: 11
Inputs: 2     Outputs: 1
Model: radial
Orientation: io (Input oriented)
Returns to scale: scaleeff (Scale efficiency)

---------------------------------
 DMU|     CRS|     VRS| ScaleEff|
---------------------------------
   1|  1.0000|  1.0000|   1.0000|
   2|  0.6223|  0.8700|   0.7153|
   3|  0.8199|  1.0000|   0.8199|
   4|  1.0000|  1.0000|   1.0000|
   5|  0.3104|  0.7116|   0.4361|
   6|  0.5556|  1.0000|   0.5556|
   7|  1.0000|  1.0000|   1.0000|
   8|  0.7577|  1.0000|   0.7577|
   9|  0.8201|  1.0000|   0.8201|
  10|  0.5000|  0.5000|   1.0000|
  11|  1.0000|  1.0000|   1.0000|
---------------------------------
```

3.2. Radial output oriented model: Constant and variable returns to scale

It is possible to measure the output oriented technical efficiency of each observation by solving the following linear program, counterpart to (1):

$$\begin{aligned} \max_{\phi,\lambda} \quad & \phi \\ \text{subject to} \quad & \\ & X\lambda \leq \mathbf{x}_o \\ & \phi\mathbf{y}_o \leq Y\lambda \\ & \lambda \geq \mathbf{0}. \end{aligned} \tag{3}$$

In this case, the optimal solution is denoted by ϕ^*_{CRS} with the constraints ensuring that $(\mathbf{x}_o, \phi^*_{\text{CRS}}\mathbf{y}_o)$ belongs to P_{CRS}. Now the objective seeks the maximum ϕ_{CRS} that increases the output vector $\mathbf{y}_o$ radially to $\phi^*_{\text{CRS}}\mathbf{y}_o$ while remaining in P_{CRS}. A feasible solution signaling radial efficiency is $\phi^*_{\text{CRS}} = 1$. Therefore if $\phi^*_{\text{CRS}} > 1$, the observation is *radially* inefficient and $(\lambda X, \lambda Y)$ outperforms $(\mathbf{x}_o, \mathbf{y}_o)$. Again, there might be farther input excesses and outputs shortfalls, with: $\mathbf{s}^- = \mathbf{x}_o - X\lambda$, and $\mathbf{s}^+ = Y\lambda - \phi^*_{\text{CRS}}\mathbf{y}_o$ with $\mathbf{s}^- \geq 0$ and $\mathbf{s}^+ \geq 0$ for any feasible solution (ϕ, λ). To calculate these slacks in a second stage, the corresponding program incorporating the optimal value ϕ^*_{CRS} is needed:

$$\begin{aligned} \max_{\lambda,\mathbf{s}^-,\mathbf{s}^+} \quad & \omega = \mathbf{e}\mathbf{s}^- + \mathbf{e}\mathbf{s}^+ \\ \text{subject to} \quad & \\ & \mathbf{s}^- = \mathbf{x}_o - X\lambda \\ & \mathbf{s}^+ = Y\lambda \ - \phi^*_{\text{CRS}}\mathbf{y}_o \\ & \lambda \geq \mathbf{0},\ \mathbf{s}^- \geq 0, \mathbf{s}^+ \geq 0. \end{aligned} \tag{4}$$

Finally, it is also possible to calculate the technical efficiency with respect to P_{VRS} by solving the programs for the radial component and its associated input and output slacks analogous to (3) and (4), but adding the VRS contraint $\sum_{j=1}^{n} \lambda_j = 1$. If $\phi^*_{\text{VRS}} = 1$ the observation is radially efficient, while it is technically efficient if $\mathbf{s}^{-*} = 0$ and $\mathbf{s}^{+*} = 0$ in the second stage. The scale efficiency defines as $SE = \phi^*_{\text{VRS}}/\phi^*_{\text{CRS}}$ and radial efficiency can be now decomposed into pure technical efficiency, PTE, and scale efficiency: $TE = \phi^*_{\text{CRS}} = PTE \times SE = \phi^*_{\text{VRS}} \times SE$.

The radial output oriented model is computed in MATLAB using the same `dea(X, Y, ...)` function with the `orient` parameter set to `oo` (output oriented). Again, the returns to scale assumption can be specified by setting the `rts` parameter to `crs` (constant returns to scale; default) or `vrs` (variable returns to scale).

```
> oo = dea(X, Y, 'orient', 'oo');
> deadisp(oo);


_______________________________
Data Envelopment Analysis (DEA)

DMUs: 11
Inputs: 2     Outputs: 1
```

```
Model: radial
Orientation: oo (Output oriented)
Returns to scale: crs (Constant)

-------------------------------------------------------------------
 DMU|       X1|       X2|        Y|      Phi| slackX1| slackX2|  slackY|
-------------------------------------------------------------------
   1|   5.0000| 13.0000| 12.0000|   1.0000|  0.0000|  0.0000|  0.0000|
   2|  16.0000| 12.0000| 14.0000|   1.6070|  0.0000|  0.0000|  0.0000|
   3|  16.0000| 26.0000| 25.0000|   1.2197|  0.0000|  0.0000|  0.0000|
   4|  17.0000| 15.0000| 26.0000|   1.0000|  0.0000|  0.0000|  0.0000|
   5|  18.0000| 14.0000|  8.0000|   3.2220|  0.0000|  0.0000|  0.0000|
   6|  23.0000|  6.0000|  9.0000|   1.8000|  8.0000|  0.0000|  0.0000|
   7|  25.0000| 10.0000| 27.0000|   1.0000|  0.0000|  0.0000|  0.0000|
   8|  27.0000| 22.0000| 30.0000|   1.3198|  0.0000|  0.0000|  0.0000|
   9|  37.0000| 14.0000| 31.0000|   1.2194|  2.0000|  0.0000|  0.0000|
  10|  42.0000| 25.0000| 26.5000|   2.0000|  0.0000|  0.0000|  0.0000|
  11|   5.0000| 17.0000| 12.0000|   1.0000|  0.0000|  4.0000|  0.0000|
-------------------------------------------------------------------
```

```
> oo_vrs = dea(X, Y, 'orient', 'oo', 'rts', 'vrs');
> deadisp(oo_vrs);
```

```
_______________________________
Data Envelopment Analysis (DEA)

DMUs: 11
Inputs: 2     Outputs: 1
Model: radial
Orientation: oo (Output oriented)
Returns to scale: vrs (Variable)

-------------------------------------------------------------------
 DMU|       X1|       X2|        Y|      Phi| slackX1| slackX2|  slackY|
-------------------------------------------------------------------
   1|   5.0000| 13.0000| 12.0000|   1.0000|  0.0000|  0.0000|  0.0000|
   2|  16.0000| 12.0000| 14.0000|   1.5075|  0.0000|  0.0000|  0.0000|
   3|  16.0000| 26.0000| 25.0000|   1.0000|  0.0000|  0.0000|  0.0000|
   4|  17.0000| 15.0000| 26.0000|   1.0000|  0.0000|  0.0000|  0.0000|
   5|  18.0000| 14.0000|  8.0000|   3.2039|  0.0000|  0.0000|  0.0000|
   6|  23.0000|  6.0000|  9.0000|   1.0000|  0.0000|  0.0000|  0.0000|
   7|  25.0000| 10.0000| 27.0000|   1.0000|  0.0000|  0.0000|  0.0000|
   8|  27.0000| 22.0000| 30.0000|   1.0000|  0.0000|  0.0000|  0.0000|
   9|  37.0000| 14.0000| 31.0000|   1.0000|  0.0000|  0.0000|  0.0000|
  10|  42.0000| 25.0000| 26.5000|   1.1698|  5.0000| 11.0000|  0.0000|
  11|   5.0000| 17.0000| 12.0000|   1.0000|  0.0000|  4.0000|  0.0000|
-------------------------------------------------------------------
```

```
> oo_scale = deascale(X, Y, 'orient', 'oo');
> deadisp(oo_scale);

-------------------------------
Data Envelopment Analysis (DEA)

DMUs: 11
Inputs: 2     Outputs: 1
Model: radial
Orientation: oo (Output oriented)
Returns to scale: scaleeff (Scale efficiency)

---------------------------------
 DMU|      CRS|      VRS| ScaleEff|
---------------------------------
   1|   1.0000|   1.0000|   1.0000|
   2|   1.6070|   1.5075|   1.0660|
   3|   1.2197|   1.0000|   1.2197|
   4|   1.0000|   1.0000|   1.0000|
   5|   3.2220|   3.2039|   1.0056|
   6|   1.8000|   1.0000|   1.8000|
   7|   1.0000|   1.0000|   1.0000|
   8|   1.3198|   1.0000|   1.3198|
   9|   1.2194|   1.0000|   1.2194|
  10|   2.0000|   1.1698|   1.7097|
  11|   1.0000|   1.0000|   1.0000|
---------------------------------
```

3.3. The directional model: Constant and variable returns to scale

Chambers, Chung, and Färe (1996) introduced a measure of efficiency that projects observation $(\mathbf{x}_o, \mathbf{y}_o)$ in a pre-assigned direction $\mathbf{g} = \left(-\mathbf{g}_{\mathbf{x}}^{-}, \mathbf{g}_{\mathbf{y}}^{+}\right) \neq \mathbf{0}_{m+s}$, $\mathbf{g}_{\mathbf{x}}^{-} \in \mathbb{R}^m$ and $\mathbf{g}_{\mathbf{y}}^{+} \in \mathbb{R}^s$, in a proportion β. The associated linear program is:

$$
\begin{aligned}
\max_{\beta,\lambda} \quad & \beta \\
\text{subject to} \quad & \\
& X\lambda \leq \mathbf{x}_o - \beta \mathbf{g}_{\mathbf{x}}^{-} \\
& Y\lambda \geq \mathbf{y}_o + \beta \mathbf{g}_{\mathbf{y}}^{+} \\
& \lambda \geq \mathbf{0}.
\end{aligned} \tag{5}
$$

In this occasion the optimal solution to this program corresponds to β^*_{CRS}. Now $\beta^*_{\mathrm{CRS}} = 0$ signals *directional* inefficiency. Therefore if $\beta^*_{\mathrm{CRS}} > 0$, the observation is inefficient and $(\lambda X, \lambda Y)$ outperforms $(\mathbf{x}_o, \mathbf{y}_o)$, with $\left(\mathbf{x}_o - \beta^*_{\mathrm{CRS}}\mathbf{g}_{\mathbf{x}}^{-}, \mathbf{y}_o + \beta^*_{\mathrm{CRS}}\mathbf{g}_{\mathbf{y}}^{+}\right) \in P_{\mathrm{CRS}}$. It is again possible that further input excesses and outputs shortfalls exist. The slacks corresponding to $\mathbf{s}^- = \mathbf{x}_o - \beta\mathbf{g}_{\mathbf{x}}^- - X\lambda$, and $\mathbf{s}^+ = Y\lambda - \mathbf{y}_o + \beta\mathbf{g}_{\mathbf{y}}^+$, with $\mathbf{s}^- \geq 0$ and $\mathbf{s}^+ \geq 0$ for any feasible solution (β, λ). As a result, a second stage is once again needed to calculate these slacks. The next program incorporating the optimal value β^*_{CRS} allows determination of these values:

$$\begin{aligned}
\max_{\lambda,\mathbf{s}^-,\mathbf{s}^+} \quad & \omega = \mathbf{e}\mathbf{s}^- + \mathbf{e}\mathbf{s}^+ \\
\text{subject to} \quad & \\
& \mathbf{s}^- = \mathbf{x}_o - \beta\mathbf{g}_{\mathbf{x}}^- - X\lambda \\
& \mathbf{s}^+ = Y\lambda - \mathbf{y}_o + \beta\mathbf{g}_{\mathbf{y}}^+ \\
& \lambda \geq \mathbf{0},\ \mathbf{s}^- \geq 0, \mathbf{s}^+ \geq 0.
\end{aligned} \tag{6}$$

As in the input and output oriented models, one may also calculate technical efficiency with respect to P_{VRS}. This requires solving equivalent programs to (5) and (6) adding the VRS contraint $\sum_{j=1}^{n} \lambda_j = 1$. If $\beta^*_{\text{VRS}} = 0$ and $\mathbf{s}^{-*} = 0$ and $\mathbf{s}^{+*} = 0$, the observation is *technically* efficient. Consequently, we now define scale efficiency as $SE = \beta^*_{\text{CRS}} - \beta^*_{\text{VRS}}$ and directional efficiency is decomposed into pure technical efficiency, PTE, and the scale efficiency term: $TE = \beta^*_{\text{CRS}} = PTE + SE = \beta^*_{\text{VRS}} + SE$.

The directional model is oriented both in the input and output dimensions, while the choice of directional vector corresponds to the researcher. Customarily, to keep consistency with the radial models, the observed amounts of input and output set the direction: $\mathbf{g} = \left(-\mathbf{g}_{\mathbf{x}}^-, \mathbf{g}_{\mathbf{y}}^+\right) = (-\mathbf{x}_o, \mathbf{y}_o)$, coinciding with the generalized Farrell measure, Briec (1997). In this case it can be shown that the directional model nests the input and output oriented models. Indeed, if $\left(-\mathbf{g}_{\mathbf{x}}^-, \mathbf{g}_{\mathbf{y}}^+\right) = (-\mathbf{x}_o, \mathbf{0})$, then $\beta^* = 1 - \theta^*$, while if $\left(-\mathbf{g}_{\mathbf{x}}^-, \mathbf{g}_{\mathbf{y}}^+\right) = (\mathbf{0}, \mathbf{y}_o)$, $\beta^* = \phi^* - 1$. However, other choices are available; particulary $\left(-\mathbf{g}_{\mathbf{x}}^-, \mathbf{g}_{\mathbf{y}}^+\right) = (-\mathbf{1}, \mathbf{1})$ or the mean of the data: $\left(-\mathbf{g}_{\mathbf{x}}^-, \mathbf{g}_{\mathbf{y}}^+\right) = (-\bar{\mathbf{x}}_{\mathbf{o}}, \bar{\mathbf{y}}_{\mathbf{o}})$, which are neutral with respect to the orientation, as it does not use the individual weights corresponding to the observed amounts of inputs and outputs.[6] When deciding on the directional model, the researcher must declare whether the direction corresponds to the observed input or outputs mixes, the unitary vectors, or her own choice of directional vector. In this case she must introduce the directional input and output matrices.

The directional model can be computed in MATLAB using the `dea(X, Y, ...)` function with the `orient` parameter set to `ddf` (directional). The input and output directions are specified in the `Gx` and `Gy` parameters as a matrix or as a scalar (usually, 0 or 1). If omitted, `X` and `Y` will be used for `Gx` and `Gy` respectively. The returns to scale assumption can be specified by setting the `rts` parameter to `crs` (constant returns to scale; default) or `vrs` (variable returns to scale).

```
> ddf = dea(X, Y, 'orient', 'ddf', 'Gx', X, 'Gy', Y);
> deadisp(ddf);
```

```
_______________________________
Data Envelopment Analysis (DEA)
```

[6]Other directions are possible, including elaborated transformations driven by the data as proposed by Daraio and Simar (2016), or those projecting observations to economic optima as introduced by Zofío, Pastor, and Aparicio (2013), e.g. maximum profit–as shown in section 5.

```
DMUs: 11
Inputs: 2     Outputs: 1
Model: radial
Orientation: ddf (Directional distance function)
Returns to scale: crs (Constant)

-------------------------------------------------------------------
 DMU|      X1|      X2|       Y|    Beta| slackX1| slackX2|  slackY|
-------------------------------------------------------------------
   1|  5.0000| 13.0000| 12.0000|  0.0000|  0.0000|  0.0000|  0.0000|
   2| 16.0000| 12.0000| 14.0000|  0.2328|  0.0000|  0.0000|  0.0000|
   3| 16.0000| 26.0000| 25.0000|  0.0990|  0.0000|  0.0000|  0.0000|
   4| 17.0000| 15.0000| 26.0000|  0.0000|  0.0000|  0.0000|  0.0000|
   5| 18.0000| 14.0000|  8.0000|  0.5263|  0.0000|  0.0000|  0.0000|
   6| 23.0000|  6.0000|  9.0000|  0.2857|  5.7143|  0.0000|  0.0000|
   7| 25.0000| 10.0000| 27.0000|  0.0000|  0.0000|  0.0000|  0.0000|
   8| 27.0000| 22.0000| 30.0000|  0.1379|  0.0000|  0.0000|  0.0000|
   9| 37.0000| 14.0000| 31.0000|  0.0988|  1.8023|  0.0000|  0.0000|
  10| 42.0000| 25.0000| 26.5000|  0.3333|  0.0000|  0.0000|  0.0000|
  11|  5.0000| 17.0000| 12.0000|  0.0000|  0.0000|  4.0000|  0.0000|
-------------------------------------------------------------------
```

```
> ddf_vrs = dea(X, Y, 'orient', 'ddf', 'rts', 'vrs', 'Gx', X, 'Gy', Y);
> deadisp(ddf_vrs);
```

```
-------------------------------
Data Envelopment Analysis (DEA)

DMUs: 11
Inputs: 2     Outputs: 1
Model: radial
Orientation: ddf (Directional distance function)
Returns to scale: vrs (Variable)

-------------------------------------------------------------------
 DMU|      X1|      X2|       Y|    Beta| slackX1| slackX2|  slackY|
-------------------------------------------------------------------
   1|  5.0000| 13.0000| 12.0000|  0.0000|  0.0000|  0.0000|  0.0000|
   2| 16.0000| 12.0000| 14.0000|  0.1076|  0.0000|  0.0000|  0.0000|
   3| 16.0000| 26.0000| 25.0000|  0.0000|  0.0000|  0.0000|  0.0000|
   4| 17.0000| 15.0000| 26.0000|  0.0000|  0.0000|  0.0000|  0.0000|
   5| 18.0000| 14.0000|  8.0000|  0.2884|  0.0000|  0.0000|  0.3915|
   6| 23.0000|  6.0000|  9.0000|  0.0000|  0.0000|  0.0000|  0.0000|
   7| 25.0000| 10.0000| 27.0000|  0.0000|  0.0000|  0.0000|  0.0000|
   8| 27.0000| 22.0000| 30.0000|  0.0000|  0.0000|  0.0000|  0.0000|
   9| 37.0000| 14.0000| 31.0000|  0.0000|  0.0000|  0.0000|  0.0000|
```

```
  10| 42.0000| 25.0000| 26.5000|  0.1629|  0.0000|  5.4560|  0.0000|
  11|  5.0000| 17.0000| 12.0000|  0.0000|  0.0000|  4.0000|  0.0000|
---------------------------------------------------------------------

> ddf_scale = deascale(X, Y, 'orient', 'ddf', 'Gx', X, 'Gy', Y);
> deadisp(ddf_scale);

_______________________________
Data Envelopment Analysis (DEA)

DMUs: 11
Inputs: 2     Outputs: 1
Model: radial
Orientation: ddf (Directional distance function)
Returns to scale: scaleeff (Scale efficiency)

---------------------------------
 DMU|     CRS|     VRS| ScaleEff|
---------------------------------
   1|  0.0000|  0.0000|   0.0000|
   2|  0.2328|  0.1076|   0.1252|
   3|  0.0990|  0.0000|   0.0990|
   4|  0.0000|  0.0000|   0.0000|
   5|  0.5263|  0.2884|   0.2379|
   6|  0.2857|  0.0000|   0.2857|
   7|  0.0000|  0.0000|   0.0000|
   8|  0.1379|  0.0000|   0.1379|
   9|  0.0988|  0.0000|   0.0988|
  10|  0.3333|  0.1629|   0.1705|
  11|  0.0000|  0.0000|   0.0000|
---------------------------------
```

3.4. The additive model

The additive model measures technical efficiency based solely on input excesses and output shortfalls. It does not calculate efficiency scores corresponding to the radial or directional interpretation of technical efficiency *a la* Farrell (1957), but characterizes efficiency in terms of the input and output slacks: $\mathbf{s}^{-} \in \mathbb{R}^{m}$ and $\mathbf{s}^{+} \in \mathbb{R}^{s}$, respectively. The toolbox implements the weighted additive program of Lovell and Pastor (1995), whose associated linear program is:

$$
\begin{aligned}
\max_{\lambda, \mathbf{s}^{-}, \mathbf{s}^{+}} \quad & \omega = \rho_{\mathbf{x}}^{-} \mathbf{s}^{-} + \rho_{\mathbf{y}}^{+} \mathbf{s}^{+} \\
\text{subject to} \quad & \\
& X\lambda + \mathbf{s}^{-} = \\
& Y\lambda - \mathbf{s}^{+} = \mathbf{y}_o \\
& \mathbf{e}\lambda = \mathbf{1} \\
& \lambda \geq \mathbf{0},\ \mathbf{s}^{-} \geq 0, \mathbf{s}^{+} \geq 0,
\end{aligned}
\tag{7}
$$

where $(\rho_{\mathbf{x}}^{-}, \rho_{\mathbf{y}}^{+}) \in \mathbb{R}_{+}^{m} \times \mathbb{R}_{+}^{s}$ are the inputs and outputs weight vectors whose elements can vary across DMUs. Therefore, assigning unitary values, program (7) corresponds to the standard additive model, while it is worth noting that it encompasses a wide class of different DEA models known as General Efficiency Measures (GEMs). Particularly, for the Measure of Inefficiency Proportions (MIP): $(\rho_{\mathbf{x}}^{-}, \rho_{\mathbf{y}}^{+}) = (1/\mathbf{x}_o, 1/\mathbf{y}_o)$; for the range adjusted measure (RAM): $(\rho^{-}, \rho^{+}) = (1/(m+s)R^{-}, (1/(m+s)R^{+})$, where R^{-} and R^{+} are the variables' ranges; while for the Bounded Adjusted Measure (BAM): $(\rho_{\mathbf{x}}^{-}, \rho_{\mathbf{y}}^{+}) = (1/(m+s)(\mathbf{x}_o - \mathbf{x}), (1/(m+s)(\mathbf{y}_o - \mathbf{y}))$, where $\mathbf{x}$ and $\mathbf{y}$ are the minimum observed values. Additionally, it is possible to consider other fixed values with the weights representing value judgements. For observation $(\mathbf{x}_o, \mathbf{y}_o)$ the objective seeks the maximum feasible reduction in its inputs and increase in its outputs while remaining in P_{VRS}. An observation is *technically* efficient if the optimal solution $(\lambda^{*}, \mathbf{s}^{-*}, \mathbf{s}^{+*})$ of the the program is $\mathbf{s}^{-*} = 0$, and $\mathbf{s}^{+*} = 0$. Otherwise individual input reductions and output increases would be feasible, and the largest the sum of the slacks, ω_{VRS}^{*}, the largest the inefficiency.

The function `deaaddit(X, Y, ...)` solves the weighted additive model in MATLAB. The returns to scale assumption can be specified by setting the `rts` parameter to `vrs` (variable returns to scale). Inputs and outputs weights are specified in the `rhoX` and `rhoY` parameters. The default weights correspond to the MIP model if not included.

```
> add_vrs = deaaddit(X, Y, 'rts', 'vrs');
> deadisp(add_vrs);
```

```
_______________________________
Data Envelopment Analysis (DEA)

DMUs: 11
Inputs: 2     Outputs: 1
Model: additive
```

```
Orientation: none
Returns to scale: vrs (Variable)

-------------------------------------------------------------------
 DMU|      X1|      X2|       Y| slackX1| slackX2|  slackY|     Eff|
-------------------------------------------------------------------
   1|  5.0000| 13.0000| 12.0000|  0.0000|  0.0000|  0.0000|  0.0000|
   2| 16.0000| 12.0000| 14.0000|  0.0000|  0.0000|  7.1053|  0.5075|
   3| 16.0000| 26.0000| 25.0000|  0.0000|  0.0000|  0.0000|  0.0000|
   4| 17.0000| 15.0000| 26.0000|  0.0000|  0.0000|  0.0000|  0.0000|
   5| 18.0000| 14.0000|  8.0000|  0.0000|  0.0000| 17.6316|  2.2039|
   6| 23.0000|  6.0000|  9.0000|  0.0000|  0.0000|  0.0000|  0.0000|
   7| 25.0000| 10.0000| 27.0000|  0.0000|  0.0000|  0.0000|  0.0000|
   8| 27.0000| 22.0000| 30.0000|  0.0000|  0.0000|  0.0000|  0.0000|
   9| 37.0000| 14.0000| 31.0000|  0.0000|  0.0000|  0.0000|  0.0000|
  10| 42.0000| 25.0000| 26.5000| 17.0000| 15.0000|  0.5000|  1.0236|
  11|  5.0000| 17.0000| 12.0000|  0.0000|  4.0000|  0.0000|  0.2353|
-------------------------------------------------------------------
```

For illustration purposes, the Bounded Adjusted Measure (BAM) model can be computed by specifying the appropriate input and output slacks weights:

```
> n = size(X, 1);
> m = size(X, 2);
> s = size(Y, 2);
> rhoX = repelem(1 ./ ((m + s) * range(X, 1)), n, 1);
> rhoY = repelem(1 ./ ((m + s) * range(Y, 1)), n, 1);
> add_ram = deaaddit(X, Y, 'rts', 'vrs', 'rhoX', rhoX, 'rhoY', rhoY);
> deadisp(add_ram);
```

```
-------------------------------
Data Envelopment Analysis (DEA)

DMUs: 11
Inputs: 2     Outputs: 1
Model: additive
Orientation: none
Returns to scale: vrs (Variable)

-------------------------------------------------------------------
 DMU|      X1|      X2|       Y| slackX1| slackX2|  slackY|     Eff|
-------------------------------------------------------------------
   1|  5.0000| 13.0000| 12.0000|  0.0000|  0.0000|  0.0000|  0.0000|
   2| 16.0000| 12.0000| 14.0000|  0.0000|  0.0000|  7.1053|  0.1030|
   3| 16.0000| 26.0000| 25.0000|  0.0000|  0.0000|  0.0000|  0.0000|
   4| 17.0000| 15.0000| 26.0000|  0.0000|  0.0000|  0.0000|  0.0000|
   5| 18.0000| 14.0000|  8.0000|  0.0000|  0.0000| 17.6316|  0.2555|
```

```
    6|  23.0000|   6.0000|   9.0000|   0.0000|   0.0000|   0.0000|   0.0000|
    7|  25.0000|  10.0000|  27.0000|   0.0000|   0.0000|   0.0000|   0.0000|
    8|  27.0000|  22.0000|  30.0000|   0.0000|   0.0000|   0.0000|   0.0000|
    9|  37.0000|  14.0000|  31.0000|   0.0000|   0.0000|   0.0000|   0.0000|
   10|  42.0000|  25.0000|  26.5000|  17.0000|  15.0000|   0.5000|   0.4104|
   11|   5.0000|  17.0000|  12.0000|   0.0000|   4.0000|   0.0000|   0.0667|
---------------------------------------------------------------------------
```

3.5. Super-efficiency models

One interesting model that allows to differentiate across technically efficient observations is that proposed by Andersen and Petersen (1993). While DEA assigns the same value to all efficient units regardless of their performance, the super-efficiency scores allows discriminating across them depending on their values. These scores are obtained by individually solving for each observation any of the previous models, but excluding them from the reference data set, which therefore reduces to $n-1$ observations; i.e., $\tilde{P}_{\text{CRS}}\left(\mathbf{x}_o, \mathbf{y}_o\right) = \left\{(\mathbf{x},\mathbf{y})\ |\mathbf{x} \geqslant \sum_{j=1,\neq o}^{n-1} \lambda_j \mathbf{x}_j,\ \mathbf{y} \leqslant \sum_{j=1,\neq o}^{n-1} \lambda_j \mathbf{y}_j,\ \lambda \geqslant \mathbf{0}\right\}$. The magnitude of the super-efficiency score, as well as the number of observations whose efficiency changes as a result of each individual exclusion determines the importance of each efficient observation in the complete dataset.

Therefore, if an observation is inefficient, its super-efficiency score is the same as that previously calculated, while in the efficient case it is greater than one for the radially input oriented model, less than one for the output counterpart, and negative for the directional model–as inputs are to be increase and outputs reduced to reach the reference benchmarks.[7]

For these oriented models we show in what follows the formulation corresponding to the input orientation under CRS, while its output and directional counterparts are omitted for the sake of brevity. For the same reason, while the MATLAB toolbox internally solves a two stage process, the problem can be equivalently expressed according to the following single stage non-Archimedean formulation:

$$
\begin{aligned}
\min_{\tilde{\theta},\lambda,\mathbf{s}^-,\mathbf{s}^+} \quad & \tilde{\theta} - \varepsilon(\mathbf{1s}^- + \mathbf{1s}^+) \\
\text{subject to} \quad & \\
& \tilde{\theta}\mathbf{x}_o = \sum_{j=1,\neq o}^{n-1} \lambda_j \mathbf{x}_j + \mathbf{s}^- \\
& \mathbf{y}_o = \sum_{j=1,\neq o}^{n-1} \lambda_j \mathbf{y}_j - \mathbf{s}^+ \\
& \lambda \geq \mathbf{0},\ \mathbf{s}^- \geq 0,\ \mathbf{s}^+ \geq 0,
\end{aligned} \tag{8}
$$

where $\mathbf{s}^-\in \mathbb{R}^m$ and $\mathbf{s}^+\in \mathbb{R}^s$, and ϵ is an infinitesimal constant. The optimal solution–obtained from a first stage equivalent to (1) but excluding the DMU under evaluation from the reference

[7] When solving these models under the VRS assumption, it is likely that some unfeasible results are obtained.

set–is again denoted by $\tilde{\theta}^*_{\mathrm{CRS}}$. In this case, if $\tilde{\theta}^*_{\mathrm{CRS}} > 1$, the observation is super-efficient and the largest the score and the increase in the values of other observation with respect to the original calculations (1), the most relevant is the removed observation.

The radial super-efficiency model corresponds to the `deasuper(X, Y, ...)` function in MATLAB with the `orient` parameter set to the desired orientation (`io`, `oo`, or `ddf`). Once again, the returns to scale assumption can be specified by setting the `rts` parameter to `crs` (constant returns to scale; default) or `vrs` (variable returns to scale).

```
> super = deasuper(X, Y, 'orient', 'io');
> deadisp(super);


_______________________________
Data Envelopment Analysis (DEA)

DMUs: 11
Inputs: 2     Outputs: 1
Model: radial-supereff
Orientation: io (Input oriented)
Returns to scale: crs (Constant)

____________________________________________________________________
 DMU|      X1|      X2|       Y|   Theta| slackX1| slackX2|  slackY|
____________________________________________________________________
   1|  5.0000| 13.0000| 12.0000|  1.1303|  0.0000|  0.0000|  0.0000|
   2| 16.0000| 12.0000| 14.0000|  0.6223|  0.0000|  0.0000|  0.0000|
   3| 16.0000| 26.0000| 25.0000|  0.8199|  0.0000|  0.0000|  0.0000|
   4| 17.0000| 15.0000| 26.0000|  1.1168|  0.0000|  0.0000|  0.0000|
   5| 18.0000| 14.0000|  8.0000|  0.3104|  0.0000|  0.0000|  0.0000|
   6| 23.0000|  6.0000|  9.0000|  0.5556|  4.4444|  0.0000|  0.0000|
   7| 25.0000| 10.0000| 27.0000|  1.2449|  0.0000|  0.0000|  0.0000|
   8| 27.0000| 22.0000| 30.0000|  0.7577|  0.0000|  0.0000|  0.0000|
   9| 37.0000| 14.0000| 31.0000|  0.8201|  1.6402|  0.0000|  0.0000|
  10| 42.0000| 25.0000| 26.5000|  0.5000|  0.0000|  0.0000|  0.0000|
  11|  5.0000| 17.0000| 12.0000|  1.0000|  0.0000|  4.0000|  0.0000|
____________________________________________________________________

> superddf = deasuper(X, Y, 'orient', 'ddf', 'Gx', X, 'Gy', Y);
> deadisp(superddf);


_______________________________
Data Envelopment Analysis (DEA)

DMUs: 11
Inputs: 2     Outputs: 1
Model: directional-supereff
Orientation: ddf (Directional distance function)
```

```
Returns to scale: crs (Constant)

--------------------------------------------------------------------
 DMU|       X1|       X2|        Y|     Beta| slackX1| slackX2|  slackY|
--------------------------------------------------------------------
   1|   5.0000| 13.0000| 12.0000| -0.0612|  0.0000|  0.0000|  0.0000|
   2|  16.0000| 12.0000| 14.0000|  0.2328|  0.0000|  0.0000|  0.0000|
   3|  16.0000| 26.0000| 25.0000|  0.0990|  0.0000|  0.0000|  0.0000|
   4|  17.0000| 15.0000| 26.0000| -0.0552|  0.0000|  0.0000|  0.0000|
   5|  18.0000| 14.0000|  8.0000|  0.5263|  0.0000|  0.0000|  0.0000|
   6|  23.0000|  6.0000|  9.0000|  0.2857|  5.7143|  0.0000|  0.0000|
   7|  25.0000| 10.0000| 27.0000| -0.1091|  0.0000|  0.0000|  0.0000|
   8|  27.0000| 22.0000| 30.0000|  0.1379|  0.0000|  0.0000|  0.0000|
   9|  37.0000| 14.0000| 31.0000|  0.0988|  1.8023|  0.0000|  0.0000|
  10|  42.0000| 25.0000| 26.5000|  0.3333|  0.0000|  0.0000|  0.0000|
  11|   5.0000| 17.0000| 12.0000|  0.0000|  0.0000|  4.0000|  0.0000|
--------------------------------------------------------------------
```

As for the super-efficiency calculations in the additive model, we follow Du, Liang, and Zhu (2010), and solve the corresponding counterpart to (7):

$$\begin{aligned} \min_{\lambda, \mathbf{s}^-, \mathbf{s}^+} \quad & \omega = \mathbf{e}\mathbf{s}^- + \mathbf{e}\mathbf{s}^+ \\ \text{subject to} \quad & \\ & \mathbf{x}_o \geq \sum_{j=1,\neq o}^{n-1} \lambda_j \mathbf{x}_j - \mathbf{s}^- \\ & \mathbf{y}_o \leq \sum_{j=1,\neq o}^{n-1} \lambda_j \mathbf{y}_j + \mathbf{s}^+ \\ & \mathbf{e}\lambda = \mathbf{1} \\ & \lambda \geq \mathbf{0},\ \mathbf{s}^- \geq 0,\ \mathbf{s}^+ \geq 0. \end{aligned} \tag{9}$$

The constraints in the program are modified as inputs need to be increased and outputs reduced so as to reach the production possibility set. In this occasion an observation is super-efficient if the optimal solution $(\lambda^*, \mathbf{s}^{-*}, \mathbf{s}^{+*})$ yields positive slacks, and the largest their sum, the largest the super-efficiency.

The additive super-efficiency model can be computed in MATLAB using the `deaadditsuper(X, Y, ...)` function, and specifying returns to scale by setting the `rts` parameter to `crs` (constant returns to scale; default) or `vrs` (variable returns to scale). Inputs and outputs weights are specified in the `rhoX` and `rhoY` parameters. The default weights correspond to the MIP model if not included.

```
> additsuper = deaadditsuper(X, Y);
> deadisp(additsuper);
```

```
-------------------------------
Data Envelopment Analysis (DEA)

DMUs: 11
Inputs: 2     Outputs: 1
Model: additive-supereff
Orientation: none
Returns to scale: crs (Constant)

--------------------------------------------------------------------
 DMU|      X1|      X2|       Y| slackX1| slackX2|  slackY|     Eff|
--------------------------------------------------------------------
   1|  5.0000| 13.0000| 12.0000|  1.1298|  0.0000|  0.0000|  1.1298|
   2| 16.0000| 12.0000| 14.0000|     NaN|     NaN|     NaN|     NaN|
   3| 16.0000| 26.0000| 25.0000|     NaN|     NaN|     NaN|     NaN|
   4| 17.0000| 15.0000| 26.0000|  0.0000|  0.0000|  2.7200|  2.7200|
   5| 18.0000| 14.0000|  8.0000|     NaN|     NaN|     NaN|     NaN|
   6| 23.0000|  6.0000|  9.0000|     NaN|     NaN|     NaN|     NaN|
   7| 25.0000| 10.0000| 27.0000|  0.0000|  3.8713|  0.0000|  3.8713|
   8| 27.0000| 22.0000| 30.0000|     NaN|     NaN|     NaN|     NaN|
   9| 37.0000| 14.0000| 31.0000|     NaN|     NaN|     NaN|     NaN|
  10| 42.0000| 25.0000| 26.5000|     NaN|     NaN|     NaN|     NaN|
  11|  5.0000| 17.0000| 12.0000|     NaN|     NaN|     NaN|     NaN|
--------------------------------------------------------------------
```

4. Productivity change: The Malmquist index

The Malmquist index, introduced by Caves, Christensen, and Diewert (1982), measures the change in productivity of the observation under evaluation by comparing its relative performance with respect to reference technologies corresponding to two different time periods. The constant returns to scale production possibility set corresponding to $t = 1,\dots,T$ periods defines as $P^t_{\text{CRS}} = \left\{(\mathbf{x}, \mathbf{y}) \; | \mathbf{x} \geqslant X^t\lambda, \; \mathbf{y} \leqslant Y^t\lambda, \; \lambda \geqslant \mathbf{0}\right\}$, with the variable returns to scale counterpart denoted accordingly by P^t_{VRS}. The standard Malmquist index relies solely on the concept of radial efficiency and requires calculation of the input–or output–oriented scores of $(\mathbf{x}^t_o, \mathbf{y}^t_o)$ observed in two periods $t = 1, 2$, with respect to the constant returns to scale reference frontier of any period. Taking as the reference the first period, P^1_{CRS}, we denote both scores by $\theta^{1,1}_{\text{CRS}}$ and $\theta^{2,1}_{\text{CRS}}$, where the first superscript refers to the time period of the observation and the second one to that of the reference technology. While $\theta^{1,1}_{\text{CRS}}$ is the solution to program (1), the intertemporal score $\theta^{2,1}_{\text{CRS}}$ corresponds to the following program that evaluates period 2 observation $(\mathbf{x}^2_o, \mathbf{y}^2_o)$ with respect to period 1 technology:

$$\begin{aligned} \min_{\theta,\lambda} \quad & \theta \\ \text{subject to} \quad & \\ & \theta \mathbf{x}_o^2 \geq X^1\lambda \\ & Y^1\lambda \geq \mathbf{y}_o^2 \\ & \lambda \geq \mathbf{0}. \end{aligned} \tag{10}$$

Equivalently, the analogous intertemporal score $\theta_{\text{CRS}}^{1,2}$ using the second period technology as reference corresponds to the same program but reversing the time superscripts of the firm under evaluation from $(\mathbf{x}_o^2, \mathbf{y}_o^2)$ to $(\mathbf{x}_o^1, \mathbf{y}_o^1)$, and those of the reference technology from (X^1, Y^1) to (X^2, Y^2).

After the contemporary and intertemporal efficiency scores have been calculated it is possible to define the following Malmquist indices: $M_1 = \theta_{\text{CRS}}^{2,1}/\theta_{\text{CRS}}^{1,1}$ and $M_2 = \theta_{\text{CRS}}^{2,2}/\theta_{\text{CRS}}^{1,2}$. For both indices, if $M > 1$ there is productivity increase, while if $M = 1$ productivity remains unchanged and $M < 1$ signals productivity decline. Following Färe, Grosskopf, Norris, and Zhang (1994), productivity change can be decomposed into efficiency change and technical change.[8] The former corresponds to the so-called *catch-up* effect; i.e., the change in the technical efficiency of the observation under evaluation with respect to the two periods, which defines for both indices as $MTEC = \theta_{\text{CRS}}^{2,2}/\theta_{\text{CRS}}^{1,1}$. The latter corresponds to the *frontier-shift* effect, i.e. the change in the reference frontier between both periods, which defines for M_1 as $MTC_1 = (\theta_{\text{CRS}}^{2,1}/\theta_{\text{CRS}}^{2,2})$ using period 2 observation as the reference benchmark to evaluate the shift in the frontier. For M_2 it defines a $MTC_2 = (\theta_{\text{CRS}}^{1,1}/\theta_{\text{CRS}}^{1,2})$. As in the previous cases, if $MTEC > 1$ or $MTC > 1$ productivity change is driven respectively by both technical efficiency gains and technical change improvements (technical progress); while $MTEC < 1$ or $MTC < 1$ imply lower productivity due to greater inefficiency and technical regress. Finally, unitary values signal that the technical efficiency and the reference frontier remain unchanged. Therefore the decomposition of year-to-year productivity change defines as $M_1 = MTEC \times MTC_1 = \theta_{\text{CRS}}^{2,2}/\theta_{\text{CRS}}^{1,1} \times \theta_{\text{CRS}}^{2,1}/\theta_{\text{CRS}}^{2,2}$–and similarly for M_2, while the geometric mean of both indices and their corresponding decompositions represents a compromise between both values: $M = MTEC \times MTC = \theta_{\text{CRS}}^{2,2}/\theta_{\text{CRS}}^{1,1} \times (\theta_{\text{CRS}}^{2,1}/\theta_{\text{CRS}}^{2,2} \times \theta_{\text{CRS}}^{1,1}/\theta_{\text{CRS}}^{1,2})^{(1/2)}$.

Finally, it is also possible to decompose long term productivity change from an initial period to a final period into consecutive subperiods relying on the transitivity property of index numbers. This allows the analysis of productivity change by subperiods. Therefore, given a sequence of periods, i.e., $t = 1, 2, 3$, it is verified that the Malmquist index between the base and final periods can be expressed in terms of its chain components: $M^{1,3} = M^{1,2} \times M^{2,3}$. The toolbox calculates the sequence of year-to-year Malmquist indices and the cumulated Malmquist index taking the based period as reference.

To compute the Malmquist indices in MATLAB one calls the `deamalm(X, Y, ...)` function with the `orient` parameter set to any of the two orientations (`io` or oo). The parameters `X`

[8] The toolbox calculates a first level decomposition that does not take into account the contribution that scale efficiency change or returns to scale make to productivity change. Therefore, as productivity change is measured under the CRS assumptions, the input and output orientations are equal. To account for scale effects alternative second level decompositions have been proposed in the literature, see Zofío (2007). These terms can be calculated solving the necessary models under VRS.

and `Y` must be 3D arrays, with the third dimension corresponding to different time periods. By default, Malmquist indices are computed as the geometric mean of M_1 and M_2. However, for simplicity, the indices can be computed taking the first period as the base for technical change, M_1, by setting the parameter `geomean` to `0`.

```
> X = [2; 3; 5; 4; 4];
> X(:, :, 2) = [1; 2; 4; 3; 4];
> X(:, :, 3) = [0.5; 1.5; 3; 2; 4];

> Y = [1; 4; 6; 3; 5];
> Y(:, :, 2) = [1; 4; 6; 3; 3];
> Y(:, :, 3) = [  2;   4; 6; 3; 1];

> malmquist = deamalm(X, Y, 'orient', 'io');
> deadisp(malmquist)
```

```
-------------------------------
Data Envelopment Analysis (DEA)

DMUs: 5
Inputs: 1     Outputs: 1
Model: radial-malmquist
Orientation: io (Input oriented)
Returns to scale: crs (Constant)

Malmquist:
Base period is previous period

-----------------------------------------------------------
 DMU|      M1|      M2|   MTEC1|   MTEC2|    MTC1|    MTC2|
-----------------------------------------------------------
   1|  2.0000|  4.0000|  1.3333|  2.0000|  1.5000|  2.0000|
   2|  1.5000|  1.3333|  1.0000|  0.6667|  1.5000|  2.0000|
   3|  1.2500|  1.3333|  0.8333|  0.6667|  1.5000|  2.0000|
   4|  1.3333|  1.5000|  0.8889|  0.7500|  1.5000|  2.0000|
   5|  0.6000|  0.3333|  0.4000|  0.1667|  1.5000|  2.0000|
-----------------------------------------------------------
M = Malmquist. MTEC = Technical Efficiency Change. MTC = Technical Change.
```

The aggregate Malmquist index taking the first period as the base period can be computed by setting the `fixbaset` parameter to `1`.

```
> malmquist1 = deamalm(X, Y, 'orient', 'oo', 'fixbaset', 1);
> deadisp(malmquist1)
```

```
-------------------------------
Data Envelopment Analysis (DEA)
```

```
DMUs: 5
Inputs: 1     Outputs: 1
Model: radial-malmquist
Orientation: oo (Output oriented)
Returns to scale: crs (Constant)

Malmquist:
Base period is period 1

-----------------------------------------------------------
 DMU|      M1|      M2|   MTEC1|   MTEC2|    MTC1|    MTC2|
-----------------------------------------------------------
   1|  2.0000|  8.0000|  1.3333|  2.6667|  1.5000|  3.0000|
   2|  1.5000|  2.0000|  1.0000|  0.6667|  1.5000|  3.0000|
   3|  1.2500|  1.6667|  0.8333|  0.5556|  1.5000|  3.0000|
   4|  1.3333|  2.0000|  0.8889|  0.6667|  1.5000|  3.0000|
   5|  0.6000|  0.2000|  0.4000|  0.0667|  1.5000|  3.0000|
-----------------------------------------------------------
M = Malmquist. MTEC = Technical Efficiency Change. MTC = Technical Change.
```

5. Allocative models: Economic efficiency

Assuming an economic optimizing behaviour on the part of the observations, e.g., cost minimization, revenue maximization, or profit maximization, and the corresponding input and output prices: $\mathbf{w} \in \mathbb{R}^m_{++}$ and $\mathbf{p} \in \mathbb{R}^s_{++}$, it is possible to measure their economic efficiency and, based on duality theory, decompose it into the technical efficiency and allocative efficiency terms, Farrell (1957).

5.1. Cost efficiency

Since we are concerned with overall efficiency in the input space, we assume that observations minimize production costs. This implies that if observations succeed in using the inputs mix (bundle) resulting in the minimum cost of producing a given output level at the existing market prices, they are cost efficient. Let us denote by $C(\mathbf{y}, \mathbf{w})$ the minimum cost of producing the output level $\mathbf{y}$ given the input price vector $\mathbf{w}$: $C(\mathbf{y}, \mathbf{w}) = \min \left\{ \sum_{i=1}^{m} w_i x_i | \mathbf{x} \geqslant X\lambda \mathbf{y}_o \leqslant Y\lambda,\ \lambda \geqslant \mathbf{0} \right\}$, which considers the input possibility set capable of producing $\mathbf{y}_o$.[9] For the observed outputs levels we can calculate minimum cost and the associated optimal quantities of inputs $\mathbf{x}^*$ consistent with the production technology by solving the following program:

[9]For a recent discussion on the properties of the technology when decomposing economic efficiency into technical and allocative efficiencies see Aparicio, Pastor, and Zofio (2015).

$$\min_{\mathbf{x},\lambda} \quad C(\mathbf{y},\mathbf{w}) = \mathbf{w}\mathbf{x}^* \tag{11}$$
$$\text{subject to}$$
$$\mathbf{x} \geq X\lambda$$
$$Y\lambda \geq \mathbf{y}_o$$
$$\lambda \geq \mathbf{0}.$$

Once minimum cost is calculated, *cost efficiency* defines as the ratio of minimum cost to observed cost: $CE = C(\mathbf{y},\mathbf{w})/\mathbf{w}\mathbf{x_o}$. Thanks to duality results—Shephard(1953), CE can be decomposed into the technical efficiency measure associated to (1), θ^*_{CRS}, and the residual difference corresponding to the *allocative cost efficiency*: $AE = CE/\theta^*_{\text{CRS}}$. Therefore allocative efficiency defines as the ratio between minimum cost and production cost at the technically efficient projection of the observation: $AE = C(\mathbf{y},\mathbf{w})/\mathbf{w}\hat{\mathbf{x}}_o$–with $\hat{\mathbf{x}}_o = \theta^*_{\text{CRS}}\mathbf{x}$ and $CE = TE \times AE$. Consequently, if $CE < 1$ and the observation is technically efficient, $\theta^*_{\text{CRS}} = 1$, all cost inefficiency is allocative, while if the observation uses the right proportions of input quantities: $\hat{\mathbf{x}}_o = \mathbf{x}^*$, it is allocative efficient and $AE = 1$.

The cost efficiency model can be computed in MATLAB using the `deaalloc(X, Y, ...)` function. The parameter `Xprice` with inputs' prices, as a matrix or as a row vector, must be included.[10]

```
> X = [3 2; 1 3; 4 6];
> Y = [3; 5; 6];
> W = [4 2];
> P = 6;

> cost = deaalloc(X, Y, 'Xprice', W);
> deadisp(cost);
```

```
_______________________________
Data Envelopment Analysis (DEA)

DMUs: 3
Inputs: 2     Outputs: 1
Model: allocative-cost
Orientation: io (Input oriented)
Returns to scale: crs (Constant)

--------------------------------------------------------------------------------
 DMU|      X1|      X2| Xprice1| Xprice2|       Y| TechEff| AllocEff| CostEff|
--------------------------------------------------------------------------------
   1|  3.0000|  2.0000|  4.0000|  2.0000|  3.0000|  0.9000|   0.4167|  0.3750|
```

[10]If input prices are the same for all DMUs the toolbox generates a matrix with the same values. If inputs' prices differ between DMUs a matrix with individual price information can be supplied. On this occasion we illustrate the economic models with an example from (Cooper *et al.* 2007, 261).

```
   2|   1.0000|   3.0000|   4.0000|   2.0000|   5.0000|   1.0000|    1.0000|   1.0000|
   3|   4.0000|   6.0000|   4.0000|   2.0000|   6.0000|   0.6000|    0.7143|   0.4286|
-------------------------------------------------------------------------------
```

5.2. Revenue efficiency

In an equivalent way, from an output dimension one may be interested in *revenue efficiency.* Now if observations are able to produce the output mix (bundle) resulting in maximum revenue given their inputs levels at the existing market prices, they are revenue efficient. Let us denote by $R(\mathbf{x}, \mathbf{p})$ the maximum feasible revenue using inputs' levels $\mathbf{x}$ and given the outputs' prices $\mathbf{p}$: $R(\mathbf{x}, \mathbf{p}) = \max\left\{\sum_{i=1}^{s} p_i y_i | \mathbf{x}_o \geqslant X\lambda,\ \mathbf{y} \leqslant Y\lambda,\ \lambda \geqslant \mathbf{0}\right\}$; i.e., considering the output possibility set producible with $\mathbf{x}_o$.
In this case, we calculate maximum revenue along with the optimal output quantities $\mathbf{y}^*$ by solving the following program:

$$
\begin{aligned}
\max_{\mathbf{y},\lambda} \quad & R(\mathbf{x}_o, \mathbf{p}) = \mathbf{p}\mathbf{y}^* \\
\text{subject to} \quad & \\
& \mathbf{x}_o \geq X\lambda \\
& Y\lambda \geq \mathbf{y} \\
& \lambda \geq \mathbf{0}.
\end{aligned}
\tag{12}
$$

Revenue efficiency defines as the ratio of observed revenue to maximum revenue: $RE = \mathbf{p}\mathbf{y_o}/R(\mathbf{x}, \mathbf{p})$. Again, duality results from an output perspective allow us to decompose RE into the output technical efficiency measure associated to the inverse of (3), $TE = 1/\phi^*_{\text{CRS}}$, and the residual difference corresponding to the *allocative revenue efficiency*: $AE = RE \times \phi^*_{\text{CRS}}$. In this occasion allocative efficiency defines as the ratio between revenue at the technically efficient projection of the observation and maximum revenue: $AE = \mathbf{p}\hat{\mathbf{y}}_\mathbf{o}/R(\mathbf{x}, \mathbf{p})$–with $\hat{\mathbf{y}}_o = \phi^*_{\text{CRS}}\mathbf{y}_o$, and therefore $RE = TE \times AE$. Now, if $RE < 1$ and the observation is technically efficient, $\phi^*_{\text{CRS}} = 1$, all revenue inefficiency is allocative and the observation is unable to produce the right output mix. Contrarily, if the observation produces the right proportion of output quantities given their prices: $\hat{\mathbf{y}}_o = \mathbf{y}^*$, it is allocative efficient: $AE = 1$.

The revenue efficiency model can be computed in MATLAB using the `deaalloc(X, Y, ...)` function. The parameter `Yprice` with the ouputs' prices, as a matrix or as a row vector, must be included.

```
> revenue = deaalloc(X, Y, 'Yprice', P);
> deadisp(revenue);
```

```
_______________________________
Data Envelopment Analysis (DEA)

DMUs: 3
Inputs: 2     Outputs: 1
```

```
Model: allocative-revenue
Orientation: oo (Output oriented)
Returns to scale: crs (Constant)

-------------------------------------------------------------------------
 DMU|      X1|      X2|       Y|  Yprice| TechEff| AllocEff|  RevEff|
-------------------------------------------------------------------------
   1|  3.0000|  2.0000|  3.0000|  6.0000|  0.9000|   1.0000|  0.9000|
   2|  1.0000|  3.0000|  5.0000|  6.0000|  1.0000|   1.0000|  1.0000|
   3|  4.0000|  6.0000|  6.0000|  6.0000|  0.6000|   1.0000|  0.6000|
-------------------------------------------------------------------------
```

5.3. Profit efficiency: The directional approach

Profit efficiency allows studying the economic behavior of observations as profit maximizers. In this case, if observations are able to maximize the difference between revenue and cost given market prices by producing and using the right quantities of outputs and inputs respectively, they are profit efficient. The profit function defines as $\Pi(\mathbf{w},\mathbf{p}) = \max\left\{\sum_{i=1}^{s} p_i y_i - \sum_{i=1}^{m} w_i x_i \mid \mathbf{x} \geqslant X\lambda,\ \mathbf{y} \leqslant Y\lambda,\ \mathbf{e}\lambda = \mathbf{1}, \lambda \geqslant \mathbf{0}\right\}$.
Calculating maximum revenue along with the optimal output and input quantities $\mathbf{y}^*$and $\mathbf{x}^*$ requires solving:

$$
\begin{aligned}
\max_{\mathbf{x},\mathbf{y},\lambda} \quad & \Pi(\mathbf{w},\mathbf{p}) = \mathbf{p}\mathbf{y}^* - \mathbf{w}\mathbf{x}^* \\
\text{subject to} \quad & \\
& \mathbf{x} \geq X\lambda = \mathbf{x} \\
& \mathbf{y} \leq Y\lambda = \mathbf{y} \\
& \mathbf{e}\lambda = \mathbf{1} \\
& \lambda \geq \mathbf{0}.
\end{aligned}
\tag{13}
$$

Profit efficiency defines as the difference between maximum profit and observed profit. As duality results concerning its decomposition into technical and allocative profit efficiencies rely on the directional model (5), we can use the directional vector to define a normalized profit efficiency measure that is homogenous of degree one in prices–invariant to proportional price changes; i.e., $PE = (\Pi(\mathbf{w},\mathbf{p}) - (\mathbf{p}\mathbf{y}_o - \mathbf{w}\mathbf{x}_o))/(\mathbf{p}\mathbf{g}_{\mathbf{y}}^{+} + \mathbf{w}\mathbf{g}_{\mathbf{x}}^{-})$–see Chambers, Chung, and Färe (1998) and Zofío *et al.* (2013) for more recent proposals. Subsequently, profit efficiency can be decomposed into the directional technical efficiency measure associated to (5) under variable returns to scale, $TE = \beta^*_{\text{VRS}}$, and the residual difference corresponding to the *allocative profit efficiency* term: $AE = PE - \beta^*_{\text{VRS}}$. Therefore, allocative efficiency defines as the difference between maximum profit and profit at the technically efficient projection of the observation: $AE = (\Pi(\mathbf{w},\mathbf{p}) - (\mathbf{p}\hat{\mathbf{y}}_o - \mathbf{w}\hat{\mathbf{x}}_o))/(\mathbf{p}\mathbf{g}_{\mathbf{y}}^{+} + \mathbf{w}\mathbf{g}_{\mathbf{x}}^{-})$, with $(\hat{\mathbf{y}}_o, \hat{\mathbf{x}}_o) = (\mathbf{y}_o + \beta^*_{\text{VRS}}\mathbf{g}_{\mathbf{y}}^{+}, \mathbf{x}_o - \beta^*_{\text{VRS}}\mathbf{g}_{\mathbf{x}}^{-})$, and $PE = TE + AE$. Now, if $PE > 0$ and the observation is technically efficient, $\beta^*_{\text{VRS}} = 0$, all profit efficiency is allocative, and the observation is unable to produce with the optimal combination of outputs and inputs. Contrarily, if the observation supplies and demands the profit maximizing quantities of outputs and inputs, $(\hat{\mathbf{y}}_o, \hat{\mathbf{x}}_o) = (\mathbf{y}^*, \mathbf{x}^*)$, it is allocative efficient: $AE = 0$.

The profit efficiency model can be computed in MATLAB using the `deaalloc(X, Y, ...)` function. Both the parameters `Xprice` and `Yprice` with inputs and outputs prices must be included. The input and output directions are specified in the `Gx` and `Gy` parameters. If omitted, `X` and `Y` will be used for `Gx` and `Gy` respectively.

```
> profit = deaalloc(X, Y, 'Xprice', W, 'Yprice', P);
> deadisp(profit, 'names/X/Y/eff.T/eff.A/eff.P');
```

```
_______________________________
Data Envelopment Analysis (DEA)

DMUs: 3
Inputs: 2     Outputs: 1
Model: allocative-profit
Orientation: ddf (Directional distance function)
Returns to scale: crs (Constant)

____________________________________________________________
 DMU|      X1|      X2|       Y| TechEff| AllocEff| ProfEff|
____________________________________________________________
   1|  3.0000|  2.0000|  3.0000|  0.0000|   0.5294|  0.5294|
   2|  1.0000|  3.0000|  5.0000|  0.0000|  -0.0000| -0.0000|
   3|  4.0000|  6.0000|  6.0000|  0.0000|   0.1875|  0.1875|
____________________________________________________________
```

5.4. Profit efficiency: The weighted additive approach

An alternative decomposition of *profit efficiency* based on the weighted additive measure and duality results is possible, Cooper, Pastor, Aparicio, and Borras (2011). In contrast to the radial and directional approaches, the technical efficiency component of the decomposition accounts for all inefficiencies (i.e., it includes individual inputs and outputs slacks).

Profit efficiency defines in this case as the difference between maximum profit and observed profit, normalized by the minimum among the ratios of of market prices to weights. i.e., $PE = (\Pi(\mathbf{w}, \mathbf{p}) - (\mathbf{p}\mathbf{y}_o - \mathbf{w}\mathbf{x}_o))/\min\{(p_1/\rho_1^+), ..., (p_m/\rho_s^+), (w_1/\rho_1^-), ..., (\rho_1/\rho_m^-)\}$. This normalization ensures once again that the profit efficiency is homogenous of degree one in prices, rendering it units invariant.

Once the weighted additive measure (7) is calculated, profit efficiency can be decomposed into technical efficiency, $TE = \omega^*_{\text{VRS}}$, and the residual difference corresponding to the *allocative profit efficiency* term: $AE = PE - \omega^*_{\text{VRS}}$. Consequently, allocative efficiency defines as the difference between maximum profit and profit at the technically efficient projection of the observation: $AE = (\Pi(\mathbf{w}, \mathbf{p}) - (\mathbf{p}\hat{\mathbf{y}}_o - \mathbf{w}\hat{\mathbf{x}}_o))/\min\{(p_1/\rho_1^+), ..., (p_m/\rho_s^+), (w_1/\rho_1^-), ..., (\rho_1/\rho_m^-)\}$, with $(\hat{\mathbf{y}}_o, \hat{\mathbf{x}}_o) = (\mathbf{y_o} + \rho_{\mathbf{y}}^+\mathbf{s}^+, \mathbf{x_o} - \rho_{\mathbf{x}}^-\mathbf{s}^-)$, and once again $PE = TE + AE$. Now, if $PE > 0$ and the observation is technically efficient, $\omega^*_{\text{VRS}} = 0$, all profit efficiency is allocative, and the observation is unable to produce with the optimal combination of outputs and inputs. Contrarily, if the observation supplies and demands the profit maximizing quantities of outputs and inputs, $(\hat{\mathbf{y}}_o, \hat{\mathbf{x}}_o) = (\mathbf{y}^*, \mathbf{x}^*)$, it is allocative efficient: $AE = 0$.

In this section we have focused on the decomposition of profit efficiency, while the cost and revenue counterparts presented in the previous sections can be easily obtained by setting outputs and inputs weights equal to zero, and using cost and revenue as support functions, respectively.

The weighted additive profit efficiency model is computed with the `deaadditprofit(X, Y, ...)` function. Both the parameters `Xprice` and `Yprice` with inputs and outputs prices must be included. The returns to scale assumption, `rts` parameter, must be set to to `vrs` (variable returns to scale). Inputs and outputs weights are specified in the `rhoX` and `rhoY` parameters. As before, the default weights correspond to the MIP model if not included.

```
> addprofit = deaadditprofit(X, Y, 'rts', 'vrs', 'Xprice', W, 'Yprice', P);
> deadisp(addprofit, 'names/X/Y/eff.T/eff.A/eff.P');
```

```
_______________________________
Data Envelopment Analysis (DEA)

DMUs: 3
Inputs: 2     Outputs: 1
Model: additive-profit
Orientation: none
Returns to scale: vrs (Variable)

------------------------------------------------------------
 DMU|      X1|      X2|       Y| TechEff| AllocEff| ProfEff|
------------------------------------------------------------
   1|  3.0000|  2.0000|  3.0000|  0.0000|   4.5000|  4.5000|
   2|  1.0000|  3.0000|  5.0000|  0.0000|  -0.0000| -0.0000|
   3|  4.0000|  6.0000|  6.0000|  0.0000|   3.0000|  3.0000|
------------------------------------------------------------
```

6. Undesirable outputs

Along with desirable and market oriented products, observations may produce undesirable or detrimental outputs as byproducts, such as pollutants or hazardous wastes from an environmental perspective. As a result, efficiency and productivity measures that do not take into account the asymmetry between both types of production: desirable and undesirable, will result in biased assessments of performance and erroneous calculations; e.g., when assessing environmental performance and making recommendations to improve technical efficiency, one seeks increments in desirable production *but* reductions in undesirable outputs. To incorporate undesirable outputs into the efficiency and productivity change models, we rely on the directional measures (5) that treat both sets of outputs differently. This requires a redefinition of the production technology where the initial vector of $i = 1, 2, \ldots, s$ outputs $\mathbf{y} \in \mathbb{R}^s_{++}$ is partitioned into desirable and undesirable production, i.e., $\mathbf{y} = (\mathbf{y}^d, \mathbf{y}^u)$, with $\mathbf{y}^d \in \mathbb{R}^q_{++}$ and $\mathbf{y}^u \in \mathbb{R}^r_{++}$, respectively. This translates into the corresponding reference technology $P_{\text{CRS}} = \{(\mathbf{x}, \mathbf{y}^d, \mathbf{y}^u) \; | \mathbf{x} \geqslant X\lambda, \; \mathbf{y}^d \leqslant Y\lambda, \; \mathbf{y}^u = Y\lambda, \; \lambda \geqslant \mathbf{0}\}$, characterizing undesirable outputs as weakly disposable, (Cooper *et al.* 2007, ch. 12). As the directional efficiency measure resulting in increases in desirable outputs and reductions in undesirable outputs relative to the same amount of inputs is used in the following section to define the Malmquist-Luenberger productivity measure, we rely on Aparicio *et al.* (2015) contribution to calculate it, thereby preventing the inconsistencies of the original approach introduced by Chung, Färe, and Grosskopf (1997). In this case, the directional efficiency measure projecting observation $(\mathbf{x}_o, \mathbf{y}^d_o, \mathbf{y}^u_o)$ along the pre-assigned direction corresponding to the output vector $\mathbf{g}_\mathbf{y} = (\mathbf{y}^d, \mathbf{y}^u) \neq \mathbf{0}_{m+s}$, corresponds to the solution of the following program:

$$
\begin{aligned}
\max_{\beta,\lambda} \quad & \beta \\
\text{subject to} \quad & \\
& X\lambda \leq \mathbf{x}_o \\
& Y^d\lambda \geq \mathbf{y}^d_o + \beta\mathbf{y}^d_o \\
& Y^u\lambda \leq \mathbf{y}^u_o - \beta\mathbf{y}^u_o \\
& \max\{\mathbf{y}^u_i\} \geq \mathbf{y}^u_o - \beta\mathbf{y}^u_o \\
& \lambda \geq \mathbf{0}.
\end{aligned}
\tag{14}
$$

Again, the optimal solution corresponds to β^*_{CRS}, and if $\beta^*_{\text{CRS}} = 0$, with $\lambda_o = 1, \lambda_j = 0$ $(j \neq o)$, the observation is directional efficient. Otherwise, $\beta^*_{\text{CRS}} > 0$ signals inefficiency and $(\lambda X, \lambda Y^d, \lambda Y^u)$ outperforms $(\mathbf{x}_o, \mathbf{y}^d_o, \mathbf{y}^u_o)$. It is possible also to calculate the non-directional slacks, checking for further desirable output shortfalls or inputs and undesirable output excesses.

The undersirable outputs model is computed in MATLAB using the `deaund(X, Y, Yu, ...)` function, where `Y` is now reserved for the matrix of desirable outputs–i.e., from a computational perspective we drop the superscript d–, and `Yu` is the matrix of undesirable outputs.

```
> X0 = ones(5,1);
> Y0 = [7; 5; 1; 3; 4];
> Yu0 = [2; 5; 3; 3; 2];

> undesirable = deaund(X0, Y0, Yu0);
> deadisp(undesirable);
```

```
_________________________________
Data Envelopment Analysis (DEA)

DMUs: 5
Inputs: 1     Outputs: 1     Undesirable: 1
```

```
Model: directional-undesirable
Orientation: ddf (Directional distance function)
Returns to scale: crs (Constant)

--------------------------------------------------------------------
 DMU|        X|        Y|       Yu|     Beta|  slackX|  slackY| slackYu|
--------------------------------------------------------------------
   1|   1.0000|   7.0000|   2.0000|   0.0000|  0.0000|  0.0000|  0.0000|
   2|   1.0000|   5.0000|   5.0000|   0.4000|  0.0000|  0.0000|  1.0000|
   3|   1.0000|   1.0000|   3.0000|   0.8261|  0.7391|  0.0000|  0.0000|
   4|   1.0000|   3.0000|   3.0000|   0.5556|  0.3333|  0.0000|  0.0000|
   5|   1.0000|   4.0000|   2.0000|   0.2727|  0.2727|  0.0000|  0.0000|
--------------------------------------------------------------------
```

7. Productivity change: The Malmquist-Luenberger index

Finally, it is possible to define the undesirable outputs productivity counterpart to the Malmquist index. Chung *et al.* (1997) introduced the Malmquist-Luenberger (ML) productivity index measuring the change in productivity of the observation under evaluation by comparing its relative efficiency with respect to reference technologies corresponding to two different time periods. As its radial counterpart, the standard ML uses the directional efficiency score only–disregarding second stage slacks–and requires calculation of the mix period efficiency of $\left(\mathbf{x}_o^t, \mathbf{y}_o^{t,d}, \mathbf{y}_o^{t,u}\right)$ observed in two periods $t = 1, 2$. We denote both scores by $\beta_{\text{CRS}}^{1,1}$ and $\beta_{\text{CRS}}^{2,1}$, where once again the first superscript refers to the time period of the observation and the second one to that of the reference technology. While $\beta_{\text{CRS}}^{1,1}$ is the solution to program (14), the intertemporal score $\beta_{\text{CRS}}^{2,1}$ corresponds to the following program that evaluates period 2 observation $\left(\mathbf{x}_o^2, \mathbf{y}_o^{2,d}, \mathbf{y}_o^{2,u}\right)$ with respect to period 1 technology:

$$
\begin{aligned}
\max_{\beta,\lambda} \quad & \beta \\
\text{subject to} \quad & \\
& X^1\lambda \leq \mathbf{x}_o^2 \\
& Y^{1,d}\lambda \geq \mathbf{y}_o^{2,d} + \beta\mathbf{y}_o^{2,d} \\
& Y^{1,u}\lambda \leq \mathbf{y}_o^{2,u} - \beta\mathbf{y}_o^{2,u} \\
& \max\{\mathbf{y}_i^{t,u}\} \geq \mathbf{y}_o^{2,u} - \beta\mathbf{y}_o^{2,u} \\
& \lambda \geq \mathbf{0},
\end{aligned}
\tag{15}
$$

where $\max\{\mathbf{y}_i^{t,u}\}$ is the maximum observed amount of each undesirable output in all time periods, Aparicio, Pastor, and Zofio (2013). Correspondingly, the score $\beta_{\text{CRS}}^{1,2}$, using the second period technology as reference can be calculated solving an equivalent program that evaluates $(\mathbf{x}_o^1, \mathbf{y}_o^{1,d}, \mathbf{y}_o^{1,u})$ with respect to the reference technology $(X^2, Y^{2,d}, Y^{2,u})$.

Once these scores are calculated the ML indices define as follows: $ML_1 = (1+\beta^{1,1}_{\text{CRS}})/(1+\beta^{2,1}_{\text{CRS}})$ and $ML_2 = (1+\beta^{1,2}_{\text{CRS}})/(1+\beta^{2,2}_{\text{CRS}})$. If $ML > 1$ efficiency increases and the evaluated unit is capable of producing more desirable output with less undesirable production, while if $ML = 1$ productivity remains unchanged, and $ML < 1$ signals productivity decline. Following Chung *et al.* (1997) the index can be decomposed into efficiency change and technical change, which have the same interpretation that their previous Malmquist counterparts. The change in technical efficiency defines now as $MLTEC = (1+\beta^{1,1}_{\text{CRS}})/(1+\beta^{2,2}_{\text{CRS}})$, while the *frontier-shift* effect corresponding to technical change are $MLTC_1 = (1+\beta^{2,2}_{\text{CRS}})/(1+\beta^{2,1}_{\text{CRS}})$ and $MLTC_2 = (1+\beta^{1,2}_{\text{CRS}})/(1+\beta^{1,1}_{\text{CRS}})$. Again, if $MLTEC > 1$ or $MLTC > 1$ productivity change responds to both technical efficiency gains and technical change improvements (technical progress); while $MLTEC < 1$ or $MLTC < 1$ imply lower productivity with greater inefficiency and technical regress. Finally, unitary values signal that the technical efficiency and the reference frontier remain unchanged. Therefore the decomposition of productivity change defines as $ML_1 = MLTEC \times MLTC_1 = (1+\beta^{1,1}_{\text{CRS}})/(1+\beta^{2,2}_{\text{CRS}}) \times (1+\beta^{2,2}_{\text{CRS}})/(1+\beta^{2,1}_{\text{CRS}})$–and similarly for ML_2. Given a sequence a of years, the toolbox calculates the year-to-year variation of ML_1 and the geometric mean of both indices: $ML = MLTEC \times MLTC = (1+\beta^{1,1}_{\text{CRS}})/(1+\beta^{2,2}_{\text{CRS}}) \times ((1+\beta^{2,2}_{\text{CRS}})/(1+\beta^{2,1}_{\text{CRS}}) \times ((1+\beta^{1,2}_{\text{CRS}})/(1+\beta^{1,1}_{\text{CRS}}))^{(1/2)}$.

The Malmquist-Luenberger indices are computed in MATLAB calling the `deamalmluen(X, Y, Yu,...)`. The parameters `X`, `Y` and `Yu` must be 3D arrays, with the third dimension corresponding to different time periods. By default, Malmquist-Luenberger indices are computed on a year-to-year basis taking the previous year as reference period. The aggregate Malmquist-Luenberger index taking the first period as the base period can be computed by setting the `fixbaset` parameter to `1`. By default, Malmquist indices are computed as the geometric mean of M_1 and M_2. However, for simplicity, the indices can be computed taking the first period as the base for technical change, M_1, by setting the parameter `geomean` to `0`.

```
> X1 = ones(5,1);
> Y1 = [8; 5.5; 2; 2; 4];
> Yu1 = [1; 3; 2; 4; 1];

> X = X0;
> X(:, :, 2) = X1;
> Y = Y0;
> Y(:, :, 2) = Y1;
> Yu = Yu0;
> Yu(:, :, 2) = Yu1;

> ml = deamalmluen(X, Y, Yu);
> deadisp(ml);
```

```
_______________________________
Data Envelopment Analysis (DEA)

DMUs: 5
Inputs: 1     Outputs: 1     Undesirable: 1
Model: directional-malmquist-luenberger
Orientation: ddf (Directional distance function)
```

```
Returns to scale: crs (Constant)

Malmquist-Luenberger:
Base period is previous period

--------------------------------
 DMU|      ML|   MLTEC|    MLTC|
--------------------------------
   1|  1.3702|  1.0000|  1.3702|
   2|  1.1000|  0.9625|  1.1429|
   3|  1.1260|  1.0272|  1.0962|
   4|  0.9162|  0.8264|  1.1087|
   5|  1.2792|  0.9545|  1.3401|
--------------------------------
ML: Malmquist-Luenberger. MLTEC: Technical Efficiency Change.
MLTC: Technical Change.
```

8. Bootstrapping DEA estimators

The efficiency scores and productivity indices calculated in the previous sections can be considered as estimates as their values are subject to uncertainty due to sampling variation. Simar and Wilson (1998) introduce bootstrap methods that, based on resampling, provide estimation bias, confidence intervals and allow hypotheses testing. This toolbox implements the algorithms presented by these authors following Bogetoft and Otto (2011), who also discuss other non-parametric and asymptotic tests. We implement the tests that allow determining the significance of the models covered by the toolbox: The assumption about returns to scale, i.e., whether constant and variable returns to scale scores are significantly different–Simar and Wilson (2002), and the existence of productivity change over time, i.e., whether the Malmquist index and its components are different from unity, Simar and Wilson (1999).

The algorithm implemented in the toolbox is the following:

1. Selection of B independent bootstrap samples–drawn from the original dataset with replacements;

2. Calculate an initial estimate for the efficiency score of each DMU with respect to each bootstrapped sample and smooth their distribution by perturbating them with a random noise generated from from a kernel density function with scale given with bandwith h;

3. Correct the original estimates for the mean and variance of the smoothed values;

4. Obtain a second set of bootstrapped samples generating inefficient DMUs, inside the DEA attainable set and conditional on the original input–or output–mix;

5. Repeat the process, estimate the efficiency scores for each original DMU with respect to that second set, so as to obtain a set of B bootstrap estimates; and finally,

6. Based on this distribution calculate the threshold values that truncate it according to the pre-determined significance value α, so as to determine the confidence intervals for the efficiency score of each DMU.

In addition, the bootstrapped scores can be used to obtain an estimate of the bias of the true efficiency value, and thereby a bias-corrected estimator. [11]

The radial bootstrap model can be computed in MATLAB using the `deaboot(X, Y, ...)` function with the `orient` parameter set to the desired orientation (`io` for input oriented or `oo` for output oriented). The returns to scale assumption can be specified by setting the `rts` parameter to `crs` (constant returns to scale; default) or `vrs` (variable returns to scale). The number of bootstrap replication are set in the `nreps` parameter (which defaults to 200), and the significance level in the `alpha` parameter (0.05).

```
> load 'deadataFLS'
> rng(1234567); % Set seed for reproducibility

> io_b = deaboot(X, Y, 'orient', 'io', 'nreps', 200, 'alpha', 0.05);
deadisp(io_b);
```

```
_______________________________
Data Envelopment Analysis (DEA)

DMUs: 11
Inputs: 2     Outputs: 1
Model: radial-bootstrap
Orientation: io (Input oriented)
Returns to scale: crs (Constant)
Bootstrap replications: 200
Significance level: 0.05

-----------------------------------------
 DMU|     eff| effboot|  effCI1|  effCI2|
-----------------------------------------
   1|  1.0000|  0.8245|  0.7024|  0.9853|
   2|  0.6223|  0.5606|  0.4949|  0.6150|
   3|  0.8199|  0.7123|  0.6289|  0.8064|
   4|  1.0000|  0.8864|  0.7825|  0.9841|
   5|  0.3104|  0.2793|  0.2468|  0.3063|
   6|  0.5556|  0.4846|  0.4083|  0.5527|
   7|  1.0000|  0.8473|  0.7345|  0.9814|
   8|  0.7577|  0.6794|  0.5989|  0.7513|
   9|  0.8201|  0.7008|  0.6024|  0.8159|
  10|  0.5000|  0.4460|  0.3924|  0.4936|
  11|  1.0000|  0.8461|  0.7024|  0.9888|
-----------------------------------------
```

[11] A similar procedure is implemented to determine the significance of the Malmquist indices–see Simar and Wilson (1999) for the specific algorithm.

The returns to scale test is compute by calling the `deatestrts(X, Y, ...)` function with the `orient` parameter set to the desired orientation (`io` for input oriented or `oo` for output oriented). The number of bootstrap replication are set in the `nreps` parameter (which defaults to 200), and the significance level in the `alpha` parameter (0.05). Results of the test can be displayed on screen by setting the parameter `disp` to `1`.

```
> rng(1234567); % Set seed for reproducibility
> deatestrts(X, Y, 'orient', 'io', 'nreps', 200, 'alpha', 0.05, 'disp', 1);
```

```
_______________________________
DEA Test of RTS

H0: Globally CRS
H1: VRS

Bootstrap replications: 200
Significance level: 0.05

S statistic:  0.8318
Critical value:  0.7302
p-value:  0.2900
```

To compute bootstrapped Malmquist indices in MATLAB one calls the `deamalmboot(X, Y, ...)` function with the `orient` parameter set to input orientation (`io`). The parameters `X` and `Y` must be 3D arrays, with the third dimension corresponding to different time periods. The number of bootstrap replication are set in the `nreps` parameter (which defaults to 200), and the significance level in the `alpha` parameter (0.05). Again, by default, Malmquist indices are computed as the geometric mean of M_1 and M_2. However, for simplicity, the indices can be computed taking the first period as the base for technical change, M_1, by setting the parameter `geomean` to `0`.

```
> X = [2; 3; 5; 4; 4];
> X(:, :, 2) = [1; 2; 4; 3; 4];

> Y = [1; 4; 6; 3; 5];
> Y(:, :, 2) = [1; 4; 6; 3; 3];

> rng(1234567); % Set seed for reproducibility
> malmquist = deamalmboot(X, Y, 'orient', 'io', 'nreps', 200, 'alpha', 0.05);
deadisp(malmquist)
```

```
_______________________________
Data Envelopment Analysis (DEA)

DMUs: 5
Inputs: 1     Outputs: 1
```

```
Model: radial-malmquist-bootstrap
Orientation: io (Input oriented)
Returns to scale: crs (Constant)
Bootstrap replications: 200
Significance level: 0.05

Base period is previous period

-----------------------------------------
 DMU|       M|   Mboot|   McLow|   McUpp|
-----------------------------------------
   1|  2.0000|  1.7355|  1.3193|  2.1003|
   2|  1.5000|  1.5191|  1.2958|  1.7594|
   3|  1.2500|  1.3943|  1.2383|  1.5831|
   4|  1.3333|  1.4445|  1.2798|  1.6690|
   5|  0.6000|  0.8266|  0.7780|  0.8941|
-----------------------------------------
```

9. Advanced options, displaying and exporting results

9.1. Specifying DMU names

When using the functions to compute DEA models, by default observations are numbered from 1 to n. However, custom names can be assigned to DMUs by adding the parameter `names` and a cell string with the desired names when calling the functions. Using data from the example in section 6:

```
> X0 = ones(5,1);
> Y0 = [7; 5; 1; 3; 4];
> Yu0 = [2; 5; 3; 3; 2];
> names = {'A','B','C','D','E'}';

> undesirable = deaund(X0, Y0, Yu0, 'names', names);
> deadisp(undesirable);
```

```
_______________________________
Data Envelopment Analysis (DEA)

DMUs: 5
Inputs: 1     Outputs: 1     Undesirable: 1
Model: directional-undesirable
Orientation: ddf (Directional distance function)
Returns to scale: crs (Constant)

-------------------------------------------------------------------
```

```
 DMU|       X|       Y|      Yu|    Beta|  slackX|  slackY| slackYu|
--------------------------------------------------------------------
   A|  1.0000|  7.0000|  2.0000|  0.0000|  0.0000|  0.0000|  0.0000|
   B|  1.0000|  5.0000|  5.0000|  0.4000|  0.0000|  0.0000|  1.0000|
   C|  1.0000|  1.0000|  3.0000|  0.8261|  0.7391|  0.0000|  0.0000|
   D|  1.0000|  3.0000|  3.0000|  0.5556|  0.3333|  0.0000|  0.0000|
   E|  1.0000|  4.0000|  2.0000|  0.2727|  0.2727|  0.0000|  0.0000|
--------------------------------------------------------------------
```

Another approach is to change the field `names` of the returned `deaout` structure before displaying or exporting the results.

9.2. Changing the reference set

Sometimes we need to evaluate some DMUs with respect to a reference set that differs from the observed input and output data. By default, all functions solve all models using the data corresponding to `X`, `Y` and `Yu` –the observed DMUs, which also represents the reference set. However, we can change the data corresponding to the DMU to be evaluated with the `Xeval`, `Yeval` and `Yueval` optional parameters.[12]

As an example, using data from section 3, if we want to evaluate the first DMU with respect to a reference set including all the other DMUs but not itself:

```
> load 'deadataFLS'
> Xref = X(2:end, :);
> Yref = Y(2:end, :);
> Xeval = X(1, :);
> Yeval = Y(1, :);
> io1 = dea(Xref, Yref, 'orient', 'io', 'Xeval', Xeval, 'Yeval', Yeval);
> disp(io1.eff);
```

```
1.1303
```

9.3. Custom display

When calling the `deadisp(out, dispstr)` function after computing a DEA model, appropriate information depending on the estimated model will be displayed on the screen. This setting can be changed to display the desired information by specifying in the `deadisp` function the string `dispstr` (display string) as a second parameter.

The default `dispstr` after using the `dea` function is `names/X/Y/eff/slack.X/slack.Y`. Each of the fields to be displayed in the output table must be separated with a `/` including the names corresponding to the field names of the `deaout` structure. The available fields are presented in Table 1.

As an example, we can display DMU names, the efficiency measure and the efficient X's and Y's of the input-oriented model of Section 3 using `names/eff/Xeff/Yeff` in `dispstr`:

[12]This is the procedure used internally by the toolbox to compute super-efficiency models and the productivity indices.

Common fields	
Field	**Data**
names	DMU names
X	Inputs
Y	Outputs
eff	Efficiency measure
slack.X	Inputs slacks
slack.Y	Outputs slacks
lambda	Computed λ's
Xeff	Efficient X's
Yeff	Efficient Y's
exitflag	Exit flag of the optimization
Scale efficiency models	
eff.crs	CRS efficiency
eff.vrs	VRS efficiency
eff.scale	Scale efficiency
Malmquist index	
eff.M	Malmquist index
eff.MTEC	Technical efficiency change
eff.MTC	Technical change
Allocative efficiency model	
Xprice	Inputs' prices
Yprice	Outputs' prices
eff.C	Cost efficiency
eff.R	Revenue efficiency
eff.P	Profit efficiency
eff.A	Allocative efficiency
eff.T	Technical efficiency
Undesirable outputs model	
Yu	Undesirable outputs
slack.Yu	Undesirable outputs' slacks
Yueff	Efficient Yu's
Malmquist-Luenberger index	
eff.ML	Malmquist-Luenberger index
eff.MLTEC	Technical efficiency change
eff.MLTC	Technical change

Table 1: Fields of the `deaout` structure available for the `dispstr` string

```
> load 'deadataFLS'
> io = dea(X, Y, 'orient', 'io');
> deadisp(io, 'names/eff/Xeff/Yeff');
```

```
_______________________________
Data Envelopment Analysis (DEA)

DMUs: 11
Inputs: 2     Outputs: 1
Model: radial
Orientation: io (Input oriented)
Returns to scale: crs (Constant)

-----------------------------------------
 DMU|   Theta|   Xeff1|   Xeff2|    Yeff|
-----------------------------------------
   1|  1.0000|  5.0000| 13.0000| 12.0000|
   2|  0.6223|  9.9566|  7.4675| 14.0000|
   3|  0.8199| 13.1177| 21.3163| 25.0000|
   4|  1.0000| 17.0000| 15.0000| 26.0000|
   5|  0.3104|  5.5867|  4.3452|  8.0000|
   6|  0.5556|  8.3333|  3.3333|  9.0000|
   7|  1.0000| 25.0000| 10.0000| 27.0000|
   8|  0.7577| 20.4571| 16.6687| 30.0000|
   9|  0.8201| 28.7037| 11.4815| 31.0000|
  10|  0.5000| 21.0000| 12.5000| 26.5000|
  11|  1.0000|  5.0000| 13.0000| 12.0000|
-----------------------------------------
```

9.4. Exporting results

Results of the computed DEA models can be easily exported to diverse file formats for posterior analysis and sharing. First the `deaout` structure should be converted to a MATLAB `table` data type using the `dea2table(out, dispstr)` function using the desired `dispstr`.[13].

```
> load 'deadataFLS'
> io = dea(X, Y, 'orient', 'io');
> T = dea2table(io);
```

Then, the table can be exported using the MATLAB function `writetable`.[14]

```
> writetable(T, 'ioresults.csv');
```

[13] If the `dispstr` parameter is omitted, the default one is used.

[14] See the official MATLAB documentation for this function: http://www.mathworks.com/help/matlab/ref/writetable.html

10. Conclusions

The new **Data Envelopment Analysis Toolbox** covers a wide variety of models calculating efficiency and productivity measures in an organized environment for MATLAB. The models implemented correspond to the classic radially oriented, the directional model and the additive formulation. Both constant and variable returns to scale technical efficiency measures are calculated, which allows the calculation of scale efficiency. The economic performance of firms in terms of technical and allocative criteria is also presented, along with efficiency models including undesirable outputs. Productivity indices are also implemented, both the standard Malmquist index based on radial efficiency, and the Malmquist-Luenberger defined in terms of the directional distance function. Finally, statistical analyses and hypotheses testing using bootstrapping techniques are also available

We show how to organize the data, use the available functions and interpret results. To illustrate the toolbox we solve several examples that are presented in reference DEA texts and handbooks. This positions the new toolbox as a valid self-contained package for these evaluating techniques in MATLAB.

Since the code is freely available in an open source repository on GitHub, under the GNU General Public License version 3, users will benefit from the collaboration and review of the community, and can check the code to learn how DEA optimizing programs are translated into suitable code.[15]

Acknowledgments

We are grateful to Juan Aparicio for useful comments and suggestions. This research was supported by the Spanish Ministry for Economy and Competitiveness under grant MTM2013-43903-P.

References

Andersen P, Petersen NC (1993). “A Procedure for Ranking Efficient Units in Data Envelopment Analysis.” *Management Science*, **39**(10), 1261–1264.

Aparicio J, Pastor JT, Zofio JL (2013). “On the inconsistency of the Malmquist-Luenberger index.” *European Journal of Operational Research*, **229**(3), 738 – 742.

Aparicio J, Pastor JT, Zofio JL (2015). “How to properly decompose economic efficiency using technical and allocative criteria with non-homothetic *DEA* technologies.” *European Journal of Operational Research*, **240**(3), 882 – 891.

Banker R, Charnes A, Cooper W (1984). “Some Models for Estimating Technical and Scale Inefficiencies in Data Envelopment Analysis.” *Management Science*, **30**(9), 1078–1092.

Barr RS (2004). *Handbook on Data Envelopment Analysis*, chapter Dea Software Tools and Technology, pp. 539–566. Springer US, Boston, MA.

[15] The address of the repository is `https://github.com/javierbarbero/DEAMATLAB`

Bogetoft P, Otto L (2011). *Benchmarking with DEA, SFA, and R*. Springer-Verlag New York, New York: Springer.

Briec W (1997). "A Graph-Type Extension of Farrell Technical Efficiency Measure." *Journal of Productivity Analysis*, **8**(1), 95–110. ISSN 1573-0441.

Caves DW, Christensen LR, Diewert WE (1982). "The Economic Theory of Index Numbers and the Measurement of Input, Output, and Productivity." *Econometrica*, **50**(6), 1393–1414.

Chambers RG, Chung Y, Färe R (1996). "Benefit and Distance Functions." *Journal of Economic Theory*, **70**(2), 407 – 419.

Chambers RG, Chung Y, Färe R (1998). "Profit, Directional Distance Functions, and Nerlovian Efficiency." *Journal of Optimization Theory and Applications*, **98**(2), 351–364.

Charnes A, Cooper W, Rhodes E (1978). "Measuring the efficiency of decision making units." *European Journal of Operational Research*, **2**(6), 429 – 444.

Chung Y, Färe R, Grosskopf S (1997). "Productivity and Undesirable Outputs: A Directional Distance Function Approach." *Journal of Environmental Management*, **51**(3), 229 – 240.

Cooper W, Pastor JT, Aparicio J, Borras F (2011). "Decomposing profit inefficiency in DEA through the weighted additive model." *European Journal of Operational Research*, **212**(2), 411–416.

Cooper WW, Seiford LM, Tone K (2007). *Data Envelopment Analysis: A Comprehensive Text with Models, Applications, References and DEA-Solver Software*. NJ: Princeton University Press., Princeton.

Daraio C, Simar L (2016). "Efficiency and benchmarking with directional distances: a data-driven approach." *Journal of the Operations Research Society*. Forthcoming.

Du J, Liang L, Zhu J (2010). "A slacks-based measure of super-efficiency in data envelopment analysis: A comment." *European Journal of Operational Research*, **204**(3), 694 – 697.

Econometric Software I (2009). *LIMDEP*. Plainview, NY. URL http://www.limdep.com/.

Emrouznejad A, Cabanda E (eds.) (2014). *Managing Service Productivity: Using Frontier Efficiency Methodologies and Multicriteria Decision Making for Improving Service Performance*. Springer-Verlag, Berlin.

Färe R, Grosskopf S, Norris M, Zhang Z (1994). "Productivity Growth, Technical Progress, and Efficiency Change in Industrialized Countries." *American Economic Review*, **84**(1), 66–83.

Farrell M (1957). "The Measurement of Productive Efficiency." *Journal of the Royal Statistical Society, Series A, General*, **120**(3), 253–81.

Fried HO, Lovell CAK, Schmidt SD (1993). *The Measurement of Productivity Efficiency. Techniques and Applications*. Oxford University Press, New York.

Hollingsworth B (2004). "Non Parametric Efficiency Measurement." *The Economic Journal*, **114**(496), F307–F311.

Ji Y, Lee C (2010). "Data envelopment analysis." *Stata Journal*, **10**(2), 267–280(14).

Lovell CK, Pastor JT (1995). "Units invariant and translation invariant {DEA} models." *Operations Research Letters*, **18**(3), 147 – 151.

Sheel H (2000). *EMS: Efficiency Measurement System User's Manual*. University of Dortmund, Dortmund.

Sherman H, Zhu J (2006). *Service Productivity Management: Improving Service Productivity using Data Envelopment Analysis (DEA)*. Springer, New York.

Simar L, Wilson PW (1998). "Sensitivity Analysis of Efficiency Scores: How to Bootstrap in Nonparametric Frontier Models." *Management Science*, **44**(1), 49–61.

Simar L, Wilson PW (1999). "Estimating and bootstrapping Malmquist indices." *European Journal of Operational Research*, **115**(3), 459–471.

Simar L, Wilson PW (2002). "Non-parametric tests of returns to scale." *European Journal of Operational Research*, **139**(1), 115 – 132.

StataCorp (2015). *Stata Statistical Software: Release 14*. StataCorp LP, College Station, TX. URL http://www.stata.com.

The MathWorks, Inc (2015). *MATLAB — The Language of Technical Computing, Version R2015a (8.5)*. Natick, Massachusetts. URL http://www.mathworks.com/products/matlab/.

Wilson PW (2008). "FEAR: A software package for frontier efficiency analysis with R." *Socio-Economic Planning Sciences*, **42**(4), 247–254.

Zofío JL (2007). "Malmquist productivity index decompositions: a unifying framework." *Applied Economics*, **39**(18), 2371–2387.

Zofío JL, Pastor JT, Aparicio J (2013). "The directional profit efficiency measure: on why profit inefficiency is either technical or allocative." *Journal of Productivity Analysis*, **40**(3), 257–266.