*Journal of Statistical Software*

*November 2017, Volume 81, Issue 6.* *doi: 10.18637/jss.v081.i06*

# Stochastic Frontier Analysis Using SFAMB for Ox

**Jonathan Holtkamp**
University of Goettingen

**Bernhard Brümmer**
University of Goettingen

**Abstract**

**SFAMB** is a flexible econometric tool designed for the estimation of stochastic frontier models. Ox is a matrix language used in different modules, with a console version freely available to academic users. This article provides a brief introduction to the field of stochastic frontier analysis, with examples of code (input and output) as well as a technical documentation of member functions. **SFAMB** provides frontier models for both cross-sectional data and panel data (focusing on fixed effects models). Member functions can be extended depending on the needs of the user.

*Keywords*: stochastic frontier analysis, panel data, Ox.

## 1. Introduction

**SFAMB** (stochastic frontier analysis using '`Modelbase`') is a package for estimating stochastic frontier production (as well as cost, distance, and profit) functions. It provides specifications for both cross-sectional data and panel data. **SFAMB** is written in Ox (Doornik 2009) and is operated by writing small programs (scripts).

The console versions of Ox are free for research and educational purposes. **Ox Console** uses **OxEdit** to run programs, while the commercial version of the programming language, **Ox Professional**, uses the graphical user environment **OxMetrics** instead.

The structure of the paper is as follows. In the next section, we briefly introduce the econometric foundations and related literature. The first part focuses on the theory of models for cross-sectional and panel data. A subsection presents the specifications that are available in **SFAMB** and mentions some related other software. Section 3 explains the usage of the package, which includes data structure, model formulation, and model output. Furthermore, it provides a detailed list of class member functions. For illustration, we present practical examples using real world data (Section 4) which are distributed with the package. We mention some possible extensions of **SFAMB** in Section 5. Finally, a technical appendix provides a brief overview of some underlying workings (Appendix A).

# 2. Econometric methods of stochastic frontier analysis

## 2.1. Cross-sectional data

*Basic approach* – `POOLED` *model*

This section provides a brief introduction to stochastic frontier (SF) techniques. A more detailed introduction can be found in Coelli, Rao, O'Donnell, and Battese (2005) or Bogetoft and Otto (2010). More advanced material is covered in Kumbhakar and Lovell (2000). The basic problem in efficiency analysis lies in the estimation of an unobservable frontier (production, distance or cost) function from observable input and output data, together with price data when necessary. Standard estimation techniques like OLS are inappropriate in this setting as they aim to describe average relationships, which are not the focus of an efficiency model.

The basic approach was simultaneously developed by Aigner, Lovell, and Schmidt (1977) (ALS), and Meeusen and Van den Broeck (1977). The following example of a production frontier highlights its most important characteristics. The basic production frontier model is given by:

$$y_i = \alpha + \boldsymbol{\beta}^\top \boldsymbol{x}_i + v_i - u_i. \tag{1}$$

On the left hand side, $y_i$ is the output (or some transformation of the output) of observation $i$ ($i = 1, 2, \ldots, N$). On the right hand side, $\boldsymbol{x}_i$ is a $K \times 1$ vector of inputs that produces output $y_i$, and the vector $\boldsymbol{\beta}$ represents technology parameters to be estimated. The most commonly used transformation of the variables is the natural logarithm. The crucial part of this formulation is the composed error term given by $\epsilon_i = v_i - u_i$, where $v_i$ represents statistical noise and $u_i$ represents inefficiency. Both error components are assumed to be independent of each other. Estimation is possible by means of maximum likelihood estimation (MLE) where distributional assumptions on the error components are required. The noise component is a conventional two-sided error, distributed as $v_i \overset{\text{iid}}{\sim} N(0, \sigma_v^2)$. The inefficiency component is a non-negative disturbance that can be modeled using several distributions.[1] However, the truncated normal and half-normal distributions are most frequently used and implemented in **SFAMB** (see Table 2). In case of the normal-truncated normal SF model, the random variable $u_i$ is distributed as $u_i \overset{\text{iid}}{\sim} N^+(\mu, \sigma_u^2)$. If $\mu$ is set to zero, the model becomes the normal-half normal SF model.

Extensions of the basic SF approach allow us to model the location and scale of the inefficiency distribution in a more flexible way. The corresponding covariates are often labeled as $z$-variables. Alvarez, Amsler, Orea, and Schmidt (2006) offer a comprehensive discussion of this topic and the so-called "scaling property".

Another useful overview is given by Lai and Huang (2010) who summarize and categorize several well-known models.[2] The so-called KGMHLBC[3] model parameterizes $\mu$ and originally assumes the following inefficiency distribution, $u_i \sim N^+(\mu_0 + \boldsymbol{\theta}^\top \boldsymbol{z}_i, \sigma_u^2)$. If $\mu$ is set to zero

[1] This text cannot provide a full overview of all relevant models. Kumbhakar and Lovell (2000) and more recently Greene (2008) provide very detailed surveys on applied SF models.

[2] The following abbreviations used by Lai and Huang (2010) were already used by Alvarez *et al.* (2006). The abbreviation KGMHLBC was introduced by Wang and Schmidt (2002).

[3] Kumbhakar, Gosh, and McGuckin (1991); Huang and Liu (1994); Battese and Coelli (1995).

and the scale is modeled using an exponential form, it becomes the RSCFG[4] model, where $u_i \sim N^+(0, \exp(2(\delta_0 + \boldsymbol{\delta}^\top \boldsymbol{z}_i)))$. The combination of both models leads to the following form, $u_i \sim N^+(\mu_i = \mu_0 + \boldsymbol{\theta}^\top \boldsymbol{z}_i, \sigma^2_{u,i} = \exp(2(\delta_0 + \boldsymbol{\delta}^\top \boldsymbol{z}_i)))$, that according to Lai and Huang (2010) could be labeled as a generalized linear mean (GLM) model.[5]

Jondrow, Lovell, Materov, and Schmidt (1982) present a point estimator of inefficiency, given by $\mathsf{E}(u_i|\epsilon_i)$. Battese and Coelli (1988) show that if the dependent variable is in logarithms, a more appropriate estimator is the point estimator of technical efficiency, given by $TE_i = \mathsf{E}(\exp(-u_i)|\epsilon_i)$.

## 2.2. Panel data

*Unobserved heterogeneity* – `LSDV` *model*

Panel data provide additional information because each individual is observed over a certain time period, where periods are indexed with $t$ ($t = 1, 2, \ldots, T$). The respective production function model, estimated by OLS, can be written as:

$$y_{it} = \alpha_i + \boldsymbol{\beta}^\top \boldsymbol{x}_{it} + v_{it}. \tag{2}$$

This formulation includes the time dimension, a conventional two-sided error $v_{it} \overset{\text{iid}}{\sim} N(0, \sigma_v^2)$, and an individual intercept $\alpha_i$. The model's virtue originates from the identification of these $N$ time-invariant parameters. These "fixed" effects absorb unmeasured time-invariant individual attributes, and hence, the model accounts for unobserved heterogeneity. One commonly used name of this approach is "least squares with dummy variables" (`LSDV`).

Instead of estimating the dummy variables, the conventional strategy is to apply within-transformation to the dependent and independent variable(s), e.g., for an independent variable:

$$\tilde{x}_{it} = x_{it} - \bar{x}_i.$$

The observation on $x_{it}$ is transformed by subtracting the respective individual mean $\bar{x}_i$. The resulting variable $\tilde{x}_{it}$ is a deviation from the mean. This procedure eliminates the individual effects because $\tilde{\alpha}_i = \alpha_i - \alpha_i = 0$. The transformed variables are used for model estimation. After estimation, the individual effects are calculated as:

$$\hat{\alpha}_i = \bar{y}_i - \hat{\boldsymbol{\beta}}^\top \bar{\boldsymbol{x}}_i.$$

Schmidt and Sickles (1984) use the model in a frontier context. They interpret the individual with the highest intercept as 100% technically efficient and determine the inefficiency of the remaining individuals as $u_i = \max(\hat{\alpha}) - \hat{\alpha}_i$. Accordingly, efficiency scores are time-invariant and are given by $TE_i = \mathsf{E}(\exp(-u_i))$.

[4] Reifschneider and Stevenson (1991); Caudill, Ford, and Gropper (1995).

[5] Actually, they label models that include the KGMHLBC form as generalized exponential mean (GEM) models. The reason is that they refer to the exponential form of the model that has been proposed by Alvarez *et al.* (2006). Following the categorization of Lai and Huang (2010), the model implemented in **SFAMB** is a GLM model. Furthermore, note that in **SFAMB** the respective scale parameter in the `POOLED` model is (the natural logarithm of) $\sigma_{u,i}$, and not $\sigma^2_{u,i}$. While $\sigma^2_u$ is often used, the original formulation of CFG involved $\sigma_u$.

If (in)efficiency should be modeled as time-invariant or not, depends on the objective and empirical application (see Greene 2008 for a comprehensive discussion). Nevertheless, in a longer panel, the assumption of time-varying inefficiency will usually be attractive. This fact has motivated extensions of the above model as well as the development of other approaches. One famous example is the model of Battese and Coelli (1992) (BC92) that has been applied in many empirical studies. This model specifies $u_{it} = \exp(-\eta(t-T)) \times u_i$, and nests the case of persistent inefficiency (if $\eta = 0$). **SFAMB** does not support this model, but other packages do (see Table 3).

*Unobserved heterogeneity in SFA: Dummy variables –* `TFE` *model*

The approach of Schmidt and Sickles (1984) is a reinterpretation of the well-known panel data model. However, there is no differentiation between heterogeneity and inefficiency. A complete panel SF model takes both components into account:

$$y_{it} = \alpha_i + \boldsymbol{\beta}^\top \boldsymbol{x}_{it} + v_{it} - u_{it}. \tag{3}$$

This model is proposed by Greene (2005, p. 277) who labels it as the "true fixed effects [`TFE`] stochastic frontier model". Estimation of this model requires the inclusion of all $N$ dummy variables, i.e., the number of intercepts to be estimated corresponds to the number of individuals. With fixed $T$, the estimate of the error variance is inconsistent (incidental parameters problem). Furthermore, it is likely to be biased as pointed out by Chen, Schmidt, and Wang (2014). This is a relevant issue since this estimate is required for the assessment of inefficiency.

*Elimination of dummies –* `WT` *model*

Wang and Ho (2010) propose an extension to overcome the incidental parameters problem. Their model is based on deviations from means (within-transformation; `WT`):[6]

$$\tilde{y}_{it} = \boldsymbol{\beta}^\top \tilde{\boldsymbol{x}}_{it} + \tilde{v}_{it} - \tilde{u}_{it}. \tag{4}$$

This represents either a normal-truncated normal or a normal-half normal SF model where the noise component is distributed as $v_{it} \overset{\text{iid}}{\sim} N(0, \sigma_v^2)$. Let the vector of transformed $v_{it}$ be denoted by $\tilde{\boldsymbol{v}}_i = (\tilde{v}_{i1}, \ldots, \tilde{v}_{iT})^\top$. This vector has a multivariate normal distribution, i.e., $\tilde{\boldsymbol{v}}_i \sim MN(0, \Pi)$, where $\Pi$ is a $T \times T$ covariance matrix.[7]

The specification of time-varying inefficiency ($u_{it}$) is more involved. Here, the ("basic") inefficiency component is assumed to be producer-specific, but time-invariant, i.e., $u_i^* \overset{\text{iid}}{\sim} N^+(\mu, \sigma_u^2)$, where $\mu$ is equal to zero in the case of a half-normal distribution. Inefficiency varies over time by means of a scaling function:

$$u_{it} = u_i^* \times h_{it} = u_i^* \times f(\boldsymbol{\delta}^\top \boldsymbol{z}_{it}) = u_i^* \times \exp(\boldsymbol{\delta}^\top \boldsymbol{z}_{it}),$$

where $\boldsymbol{z}_{it}$ is a vector of time-varying, producer-specific covariates. The transformed inefficiency component results from the transformation of the scaling function:

$$\tilde{u}_{it} = u_i^* \times \tilde{h}_{it}.$$

[6] They also demonstrate how the model can be estimated by first-differencing.
[7] The panel may be unbalanced.

Wang and Ho (2010) present the conditional expectation of $u_{it}$ in their Equation 30; efficiency estimates are given by $TE_{it} = \mathsf{E}(\exp(-u_{it}|\tilde{\epsilon}_{it}))$. The individual effects are calculated as:

$$\hat{\alpha}_i = \bar{y}_i - \hat{\boldsymbol{\beta}}^\top \bar{\boldsymbol{x}}_i + \bar{u}_i.$$

*Consistent estimation with time-varying inefficiency –* `CFE` *model*

Consistent estimation of the fixed effects SF model given in Equation 3 is demonstrated by Chen *et al.* (2014). Their approach is also based on deviations from means (Equation 4), but the `CFE` (consistent fixed effects) model allows inefficiency to vary over individuals and time, without an auxiliary function.

The approach is characterized by two features, within-transformation and the $T-1$ deviations, as well as by the use of a more general distributional theory. Firstly, within-transformation removes the incidental parameters. Secondly, the model's likelihood function is derived from the first $T-1$ deviations, i.e., from the vector $\tilde{\boldsymbol{\epsilon}}_i^* = (\tilde{\epsilon}_{i1}, \ldots, \tilde{\epsilon}_{i,T-1})^\top$. This strategy achieves an implicit correction of the error variance.[8] The approach is based on the closed skew normal (CSN) distribution.[9]

The composed error, $\epsilon = v - u$, has a skewed distribution (to the left) due to the non-negativeness of $u$. Accordingly, the standard (half-normal) SF model has a skew normal distribution, with skewness parameter $\lambda$ and density:

$$f(\epsilon) = \frac{2}{\sigma}\,\phi\left(\frac{\epsilon}{\sigma}\right)\,\Phi\left(-\lambda\frac{\epsilon}{\sigma}\right).$$

While the skew normal distribution is a generalization of the normal distribution, it can be generalized itself by using the CSN distribution. The composed error has a CSN distribution, which is expressed by:

$$\epsilon_{it} \sim CSN_{1,1}(0, \sigma^2, -\frac{\lambda}{\sigma}, 0, 1).$$

The density of a $CSN_{p,q}$-distribution includes a $p$-dimensional pdf and a $q$-dimensional cdf of a normal distribution. The five associated parameters describe location, scale, and skewness, as well as the mean vector and covariance matrix in the cdf. With panel data, the $T$-dimensional vector $\boldsymbol{\epsilon}_i = (\epsilon_{i1}, \ldots, \epsilon_{iT})^\top$ is distributed as:

$$\boldsymbol{\epsilon}_i \sim CSN_{T,T}(0_T, \sigma^2 I_T, -\frac{\lambda}{\sigma} I_T, 0_T, I_T),$$

where $I$ is the identity matrix. Chen *et al.* (2014, p. 67) make use of the fact that the CSN distribution is "closed under linear combination", and partition the vector $\boldsymbol{\epsilon}_i$ into its mean $\bar{\epsilon}_i$ and its first $T-1$ deviations $\tilde{\boldsymbol{\epsilon}}_i^*$. The model's likelihood function is derived from $\tilde{\boldsymbol{\epsilon}}_i^*$. Therefore, it is free of incidental parameters, and the parameters to be estimated are $\boldsymbol{\beta}, \lambda$, and $\sigma^2$ – as

[8] With regards to the degrees of freedom, the correction accounts for the $N$ individuals: $df = NT - N - K = N(T-1) - K$.

[9] Chen *et al.* (2014) explain how the SF model is related to the CSN distribution and present the required properties of CSN distributed random variables. Another plain introduction to the CSN distribution in the SF context is provided by Brorsen and Kim (2013).

| | SetMethod | Example |
|---|---|---|
| SFA – cross section | POOLED | hbest1.ox |
| Least squares with dummies | LSDV | hbest2.ox |
| SFA – with dummies | TFE | hbest2.ox |
| SFA – within-transformation | WT | hbest3.ox |
| SFA – consistent fixed effects | CFE | hbest2.ox |

Table 1: Available estimators, with name of estimation method and sample files.

in the basic SF model.[10] $\bar{\epsilon}_i$ and $\tilde{\boldsymbol{\epsilon}}_i^*$ are not independent, unless $\lambda = 0$. If $\lambda = 0$ the model becomes the fixed effects model with normal error (LSDV model).

In order to assess the inefficiency, the composed error term must be recovered:

$$\epsilon_{it} = y_{it} - \hat{y}_{it} = y_{it} - \hat{\boldsymbol{\beta}}^\top \boldsymbol{x}_{it} - \hat{\alpha}_i.$$

There are two ways to calculate $\hat{\alpha}_i$. The one used here is labeled as the mean-adjusted estimate by Chen *et al.* (2014):

$$\hat{\alpha}_i^M = \bar{y}_i - \hat{\boldsymbol{\beta}}^\top \bar{\boldsymbol{x}}_i + \sqrt{\frac{2}{\pi}}\,\hat{\sigma}_u. \tag{5}$$

## 2.3. Software

**SFAMB** provides frontier models of ALS (with extensions), Schmidt and Sickles (1984), Greene (2005), Wang and Ho (2010) as well as Chen *et al.* (2014). The available estimators are listed in Table 1.

There are several other software packages that incorporate (some of) these estimators. Tables 2 and 3 provide an overview of the different specifications that are available in **SFAMB** and other software.[11]

**LIMDEP** (Econometric Software Inc. 2014) and Stata (StataCorp LP 2015) are comprehensive commercial packages that implement frontier techniques in their standard distributions. In case of Stata, there are additional third-party add-ons such as those of Wang (2012) or Belotti, Daidone, Ilardi, and Atella (2013).

Hughes (2008) has written two free packages, **sfa_hetmod** and **sfa_mod**, that can be used with **gretl** (Cottrell and Lucchetti 2014) and include variations of the standard model.

The recent package **spfrontier** (Pavlyuk 2016) is concerned with (the specific family of) spatial SF models. It is implemented in R (R Core Team 2017) and allows for various specifications.

The first program to implement frontier techniques was **Frontier** (Coelli 1996). Later, the original code was transferred to R in the package **frontier** by Coelli and Henningsen (2017). This package provides the standard model (ALS), the extension of KGMHLBC as well as the model of BC92. Its functionality is augmented by some additional options (e.g., for calculating marginal effects).

[10] The conventional parameterization is: $\lambda = \sigma_u/\sigma_v$ and $\sigma^2 = \sigma_u^2 + \sigma_v^2$.

[11] The coverage of other software may be larger as indicated here. This overview makes no claim to be complete.

| Cross-sectional data | | | | |
|---|---|---|---|---|
| POOLED | | | | |
| $u_i \sim$ | $N^+(0, \sigma_u^2)$ [ALS] | $N^+(\mu_0 + \boldsymbol{\theta}^\top \boldsymbol{z}_i, \sigma_u^2)$ [KGMHLBC] | $N^+(0, \exp(2\,\boldsymbol{\delta}^\top \boldsymbol{z}_i))$ [RSCFG] | $N^+(\mu_0 + \boldsymbol{\theta}^\top \boldsymbol{z}_i, \exp(2\,\boldsymbol{\delta}^\top \boldsymbol{z}_i))$ [GLM] |
| **SFAMB*** | ✓ | ✓ | ✓ | ✓ |
| **frontier** | ✓ | ✓ | | |
| **LIMDEP** | ✓ | ✓ | ✓ | ✓ |
| **sfa_hetmod** | | | ✓ | |
| **sfa_mod** | ✓ | | | |
| Stata | ✓ | ✓ | ✓ | ✓ |

*Here, the respective scale parameter is $\sigma_{u,i}$, and not $\sigma_{u,i}^2$.

Table 2: Available model specifications for cross-sectional data.

| Panel data | | | | | |
|---|---|---|---|---|---|
| | BC92 $u_i \sim N^+(\mu, \sigma_u^2)$ | LSDV | TFE $u_{it} \sim N^+(0, \sigma_u^2)$ | WT $u_i^* \sim N^+(0, \sigma_u^2)$ or $u_i^* \sim N^+(\mu, \sigma_u^2)$ | CFE $u_{it} \sim N^+(0, \sigma_u^2)$ |
| **SFAMB** | | ✓ | ✓ | ✓ | ✓ |
| **frontier** | ✓ | | | | |
| LIMDEP | ✓ | ✓ | ✓ | | |
| Stata | ✓ | ✓ | ✓ | ✓ | |

Table 3: Available model specifications for panel data.

Similarly, **SFAMB** offers specific member functions that can be extended by the user. To date, it is the only package including the CFE model.

# 3. Using SFAMB

## 3.1. Data organization

Ox supports different data file formats (.xls, .dta, ...) that can be directly read into an **SFAMB** object. Details can be found in printed documentation (Doornik and Ooms 2007; Doornik 2009), or online at http://www.doornik.com/ox/.

The data have to be organized in columns, i.e., one column contains one variable. The first row indicates the name. Ox interprets missing values as .NaN (Not a Number). In case of panel data, the data have to be stacked by individual ($i = 1, 2, \ldots, N$), and within individuals by time period ($t = 1, 2, \ldots, T$). The data set must include two variables that indicate the individual and the time period (the panel may be unbalanced). An example is given in Table 4.

## 3.2. Model formulation

The sequence of model formulation is sketched out in Figure 1. A new object is created in each

| id | time | y | x1 |
|---|---|---|---|
| 31 | 1 | 298384 | 24145 |
| 31 | 2 | 333522 | 27725 |
| 31 | 3 | 378768 | 38115 |
| 37 | 1 | 62473 | 3401 |
| 37 | 2 | 212442 | 12529 |
| 37 | 3 | 295142 | 16734 |
| 101 | 1 | 150037 | 10752 |
| 101 | 2 | 158909 | 10418 |
| 101 | 3 | 172744 | 10671 |

Table 4: Example data.

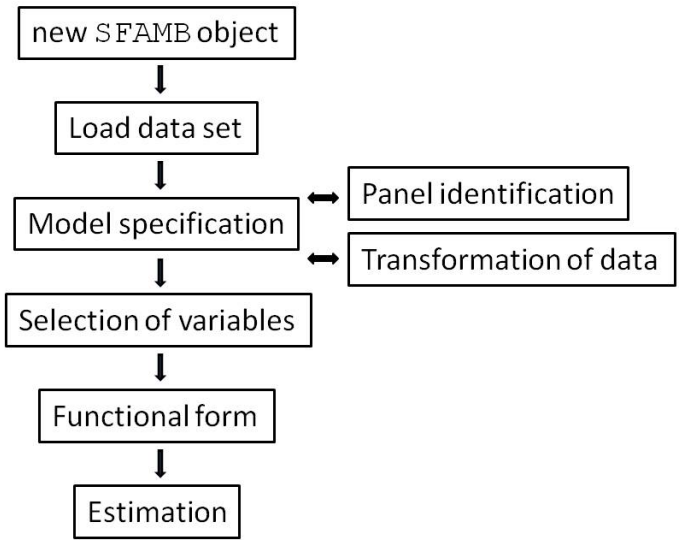


Figure 1: Model formulation.

input file (script with ending `.ox`). This object is an instance of the '`SFAMB`' class and can use the functionality of this class. The function `Load` loads the data file and creates the database. The type of model is chosen with `SetMethod`, where the respective arguments are presented in Table 1. In case of panel data, the panel structure must be specified using `Ident`. If the original data are in levels, you can use `PrepData` for transformation (the classes '`Database`' and '`Modelbase`' provide additional functions). Variables are grouped using `Select`, where variable names serve as arguments.[12]

To formulate the frontier function:

- Use `Select(Y_VAR, {".",.,.})` to select the dependent variable.
- Use `Select(X_VAR, {".",.,.})` to select the independent variable(s).

To include variables that affect the one-sided inefficiency distribution (for the `WT` model choose only one of these groups):

[12]In case of the panel models, a common constant is not identified. However, you can leave `"Constant"` in the selection because it is ignored automatically.

- Use `Select(U_VAR, {".",.,.})` to select variables that shift the individual location parameter of the distribution, $\mu_i$.
- Use `Select(Z_VAR, {".",.,.})` to select variables that affect the scale parameter of the distribution, $\sigma_{u,i}$ or $\sigma^2_{u,i}$.

`SetTranslog` can be used to choose the functional form of the frontier function. In case of the translog specification, we recommend to normalize the variables by the respective sample means. Estimation of the model is executed via `Estimate`. For more details, see the documentation of member functions in Section 3.4.

### 3.3. Model output

Besides standard results, some model-specific numbers are printed after estimation.

*Cross-sectional data*

Specific numbers for model `POOLED`:

`gamma` was defined by Battese and Corra (1977) and is given by $\gamma = \sigma_u^2/\sigma^2 = \sigma_u^2/(\sigma_v^2 + \sigma_u^2)$, where $\sigma_u^2 = \frac{1}{n}\sum_i \exp(2\,\boldsymbol{\delta}^\top \boldsymbol{z}_i)$.

`VAR(u)/VAR(total)` describes the variance decomposition of the composed error term. The share of the variance of $u$ in the total variance of the composed error is given by $\mathsf{VAR}[u]/\mathsf{VAR}[\epsilon] = [(\pi-2)/\pi]\sigma_u^2/[(\pi-2)/\pi]\sigma_u^2 + \sigma_v^2$, see Greene (2008, p. 118), where $\sigma_u^2 = \frac{1}{n}\sum_i \exp(2\,\boldsymbol{\delta}^\top \boldsymbol{z}_i)$.

`Test of one-sided err` provides a likelihood ratio test statistic for the presence of inefficiency, i.e., for the null hypothesis $H_0$: $\gamma = 0$. The critical value cannot be taken from a conventional $\chi^2$-table, see Kodde and Palm (1986).

*Panel data*

Specific numbers for model `LSDV`:

`sigma_e` describes $\sigma_v$, the square root of the corrected error variance, $\sigma_v^2 = \frac{SSR}{N(T-1)-K}$. This estimate is also used to compute the standard errors.

`AIC1 (all obs)` is given by $AIC1 = -2\ln L + 2\,(K+1)$; it uses the likelihood function $\ln L = -\frac{NT}{2}\ln(2\pi) - \frac{NT}{2}\ln(\sigma^2) - \frac{\sum_i\sum_t \tilde{v}_{it}^2}{2\sigma^2}$ with the uncorrected $\sigma^2 = \frac{SSR}{NT}$.

`AIC2` uses a different formula for the criterion, $AIC2 = \ln(\frac{SSR}{NT}) + (2\,\frac{K+N}{NT})$; that does not require the likelihood function and considers the number of individuals in the penalty term.

Specific number for models `TFE`, `WT` and `CFE`:

`lambda` given by $\lambda = \sigma_u/\sigma_v$.

### 3.4. Class member functions

These functions (user interface) together with the data members build up the '`SFAMB`' class. Some internal functions are not listed here. The interested user may consult the package's header file and source code file. Note that the class derives from the Ox '`Modelbase`' class, and hence, all underlying functions may be used.[13]

**AiHat** `AiHat();`
*Return value*
Returns the calculated individual effects $\hat{\alpha}_i$, $N \times 1$ vector.
*Description*
– Only panel data – These values can be obtained after estimation, see Section 2 for the respective formulas.

**Different functions** to extract data:
*Return value*
Different vectors or matrices.
*Description*
These functions can be used with convenient ('`Database`') functions such as `Save`, `Renew` or `savemat`.

`IDandPer();` is an $NT \times 2$ matrix with the ID of the individual (e.g., $1, 1, 1, 2, \ldots, N, N$) in the first column and the individual panel length $T_i$ in the second column. – Only panel data –

`GetLLFi();` returns the individual log-likelihood values. It is an $NT \times 1$ vector for both `POOLED` model and `LSDV` model, but an $N \times 1$ vector for the other models.

`GetResiduals();` returns the (composed) residual of the respective observation, $NT \times 1$ vector.

`GetTldata();` returns the corresponding vectors of $Y$, $X$, square, and cross terms of $X$. – Use with `SetTranslog()` –

`GetMeans();` returns the means of $Y$- and $X$-variables, $N \times (K + 1)$ matrix. – Only panel data –

`GetWithins();` returns the within-transformed $Y$- and $X$-variables, $NT \times (K + 1)$ matrix. – Only panel data –

**DropGroupIf** `DropGroupIf(const mifr);`
*No return values*
*Description*
– Only panel data – Allows the exclusion of a whole individual from the sample if the condition in one (single) period is met. Call after function `Ident`.

**mifr** is the condition that specifies the observation to be dropped, see the documentation of `selectifr`.

**Elast** `Elast(const sXname);`
*Return value*

[13] The '`Modelbase`' class derives from the '`Database`' class. Accordingly, the member functions of '`Database`' are available in **SFAMB**.

Returns the calculated output elasticities and the respective $t$-values for the specified (single) input variable (for all sample observations).
*Description*
– Use with `SetTranslog()` – Only if a translog functional form is used, $NT \times 2$ matrix. The elasticity is calculated for each observation of the sample: the output elasticity of input $k$ is given by $\partial \ln y_i / \partial \ln x_{ki}$. The $t$-values are calculated using the delta method (Greene 2012), extracting the required values from the data and covariance matrix of the parameter estimates.

`sXname` is the name of the corresponding input variable (string).

**GetResults** `GetResults(const ampar, const ameff, const avfct, const amv);`
*No return values*
*Description*
– Only `POOLED` model – This function can be used to store the results of the estimation procedure for further use. All four arguments should be addresses of variables.

`mpar` consists of a *Npar* $\times$ 3 matrix, where *Npar* is the number of parameters in the model. The first column contains the coefficient estimates, the second column the standard errors, and the third column the appropriate probabilities.

`eff` consists of a *Nobs* $\times$ 3 matrix, where *Nobs* is the number of total observations. The first column holds the point estimate for technical efficiency, the second and third columns contain the upper and lower bound of the $(1 - \alpha)$ confidence interval.

`fct` holds some likelihood function values (OLS and ML), as well as some information on the correct variance decomposition of the composed error term.

`v` variance-covariance-matrix.

**Ident** `Ident(const vID, const vPer);`
*No return values*
*Description*
– Only panel data – Identifies the structure of the panel.

`vID` is an $NT \times 1$ vector holding the identifier (integer) of the individual.

`vPer` is an $NT \times 1$ vector holding the identifier (integer) of the time period.

**Ineff** `Ineff();`
*Return value*
Returns point estimates of technical inefficiency, $NT \times 1$ vector.
*Description*
These predictions are given by the conditional expectation of $u$, see Section 2 for details.

**PrepData** `PrepData(const mSel, iNorm);`
*Return value*
Returns logarithms of the specified variables, either normalized or not.
*Description*
This function expects your data in levels and can do two things: It takes logarithms of your specified variables (if `iNorm = 0`) or it normalizes your data (by the sample mean if `iNorm = 1`), before taking logarithms. The transformed variable should receive a new name.

`mSel` is a $NT \times k$ matrix holding the respective $Y$- and $X$-variables.

`iNorm` is an integer: 0=no normalization; 1=normalization;

**SetConfidenceLevel** `SetConfidenceLevel(const alpha);`

*No return values*

*Description*

– Only `POOLED` model – This function expects a double constant, indicating the error probability for the construction of confidence bounds (default 0.05).

**SetPrintDetails** `SetPrintDetails(const bool);`

*No return values*

*Description*

– Not for `LSDV` model – Prints starting values, warnings and elapsed time if `bool` $\neq 0$.

**SetRobustStdErr** `SetRobustStdErr(const bool);`

*No return values*

*Description*

– Only `POOLED` model – By default, robust standard errors are used for the cross-sectional model. Use `FALSE` to switch off this setting.

**SetStart** `SetStart(const vStart);`

*No return values*

*Description*

This function expects a column vector of appropriate size $(K + 2)$, containing starting values for the maximum likelihood iteration. If the function is not called at all, OLS values are used in conjunction with a grid search for the SFA-specific parameter(s) $\sigma^2 = \sigma_v^2 + \sigma_u^2$. If only $(K)$ technology parameters are given, the grid search is also applied.

**SetTranslog** `SetTranslog(const iTl);`

*No return values*

*Description*

This function expects an integer to control the construction of additional regressors from the selected $X$-variables.

- A value of zero indicates no further terms to be added, e.g., for a log-linear model, this corresponds to the Cobb-Douglas form.
- A value of one indicates that all square and cross terms of all independent variables should be constructed, e.g., for a log-linear model, this corresponds to the full translog form.
- An integer value of $k > 1$ indicates that the square and cross terms should be constructed for only the first $k$ independent variables (useful when the regressor matrix contains dummy variables).

**TE** `TE();`

*Return value*

Returns point estimates of technical efficiency, $NT \times 1$ vector.

*Description*

These predictions are given by the conditional expectation of $\exp(-u)$, see Section 2 for details.

**TEint** `TEint(const dAlpha);`

*Return value*

Returns point estimates of technical efficiency as well as lower and upper bounds.

*Description*

– Only `POOLED` model – This function expects a double constant, indicating the error probability for the construction of confidence bounds (default 0.05); for details see Horrace and Schmidt (1996), for an application Brümmer (2001). It returns an $NT \times 3$ matrix structured as (point estimate – lower bound – upper bound).

**TestGraphicAnalysis** `TestGraphicAnalysis();`

*No return values*

*Description*

Only useful in conjunction with the free Ox package **GnuDraw** (Bos 2014), which is an `Ox` interface to **gnuplot** (**gnuplot** Team 2015). This function draws two (or three) graphs: A histogram of the efficiency point estimates and a respective boxplot. It displays an additional graph in case of the `POOLED` model, depicting the interval estimates of technical efficiency at the specified significance.

# 4. Examples

## 4.1. Example: `hbest1.ox`

The first example is a generalized linear mean (GLM) model, where $u_i \sim N^+(\mu, \sigma_{u,i} = \exp(\delta_0 + \boldsymbol{\delta}^\top \boldsymbol{z}_i))$. The original data are in levels and are transformed using member function `PrepData` to accommodate the translog functional form. The data are a subset of FAO/USDA data prepared by Fuglie (2012), including Sub-Saharan African countries and South Africa.

General usage and details of the Ox language are explained in Doornik and Ooms (2007). The sample file `hbest1.ox` is presented below. At the beginning of each program some header files are linked in:

```
#include <oxstd.h>
#include <packages/gnudraw/gnudraw.h>
#import  <packages/sfamb/sfamb>
```

The first file, the so-called standard header, ensures that all standard library functions can be used. The second line includes the header file of **GnuDraw** (Bos 2014), an `Ox` interface to **gnuplot** (**gnuplot** Team 2015). If it is not installed or you do not want to use this package, delete this line. However, graphics output will then be disabled in the free **Ox Console** version (in the commercial **OxMetrics** version, graphics would still be available). Alternatively, you can comment it out via `//`:

```
// #include <packages/gnudraw/gnudraw.h>
```

The third line imports the (compiled) source code of the package (you may also use `#include <packages/sfamb/sfamb.ox>`). Each Ox program is executed by the `main()` function that contains the main loop of Ox.

```
main(){
  ...
}
```

The next steps outlined follow the structure of Figure 1. A new object of class '`Sfa`' has to be declared.

```
decl fob = new Sfa();
```

The data are loaded with a call to the member function `Load`. The argument of `SetMethod` chooses the respective estimator (see Table 1). Here, the model for cross-sectional data is specified. The function `SetConstant` creates a constant (intercept).

```
fob.Load("USDAafrica.xls");
fob.SetMethod(POOLED);
fob.SetConstant();
```

Data are either used directly or prepared within the code. Here, the output variable, five input variables, and a time variable are transformed where logarithms of the mean-normalized inputs (output) are used.[14] New names are assigned to the prepared variables. These names are used for further instructions. The function `Info` is useful here because it prints summary statistics, thereby allowing the transformed data to be checked. The program always stops at an `exit` function (that is why it is commented out here).

```
decl inorm = 1;
fob.Renew(fob.PrepData(fob.GetVar("output"), inorm), "lny");
fob.Renew(fob.PrepData(fob.GetVar("labour"), inorm), "lnlab");
fob.Renew(fob.PrepData(fob.GetVar("land"), inorm), "lnland");
fob.Renew(fob.PrepData(fob.GetVar("machinery"), inorm), "lnmac");
fob.Renew(fob.PrepData(fob.GetVar("fertilizer"), inorm), "lnfert");
fob.Renew(fob.GetVar("time") - meanc(fob.GetVar("time")), "trend");
// fob.Info(); exit(1);
```

Selection of variables is carried out by `Select` where `Y_VAR` is the selection of the dependent variable and `X_VAR` is the selection of the regressors. The function uses the new variable names defined above (if your data file already includes transformed variables, you would use the names from within the file). The intercept (`"Constant"`) is available because `SetConstant` is called above. Within the `Select` function there are arrays with three elements (variable name, start lag, end lag). Here, the lags are set to zero. Note that there must not be a comma before the closing curly brace of `Select`.

[14] Normalization of inputs (and output): $\ln\left(\frac{x_{jit}}{\bar{x}_j}\right)$; normalization of time trend: $t - \bar{t}$.
`PrepData` is a member function of this package (see Section 3.4). Both of the other functions are member functions of the '`Database`' class (see Doornik and Ooms 2007).

```
fob.Select(Y_VAR, {"lny", 0, 0});
fob.Select(X_VAR, {
  "Constant", 0, 0,
  "lnlab", 0, 0,
  "lnland", 0, 0,
  "lnmac", 0, 0,
  "lnfert", 0, 0,
  "trend", 0, 0
});
```

The above selections define the production frontier. Additional covariates associated with the underlying inefficiency distribution can be introduced (`POOLED` and `WT` model). Covariates used to model the location parameter of the distribution are selected in the group `U_VAR`. Here, only `"Constant"` is selected, meaning that $\mu \neq 0$, but additional variables could be included.

```
fob.Select(U_VAR, {
  "Constant", 0, 0
});
```

Likewise, covariates used to model the scale of the distribution are selected in the group `Z_VAR`, i.e., these variables parameterize $\sigma_{u,i}$ (in case of the `WT` model, it is $\sigma^2_{u,i}$).

```
fob.Select(Z_VAR, {
  "Constant", 0, 0,
  "lnlab", 0, 0,
  "lnland", 0, 0,
  "lnmac", 0, 0,
  "lnfert", 0, 0
});
```

The next three lines allow for different adjustments. `SetSelSample` is required and can be used to choose a subset of the data (here: full sample). `SetPrintSfa` ensures that estimation output is printed. `MaxControl` is an optional function that allows for documentation and adjustments of the maximization procedure.

```
fob.SetSelSample(-1, 1, -1, 1);
fob.SetPrintSfa(TRUE);
MaxControl(1000, 10, TRUE);
```

The functional form of the production frontier is chosen by `SetTranslog` where the options are either Cobb-Douglas or translog. Here, a translog form is specified. Estimation of the model is invoked via `Estimate`.

```
fob.SetTranslog(1);
fob.Estimate();
```

A number of results can be obtained after estimation. In the SF context, the efficiency scores ($TE_i$) are of particular interest. Here, the point estimates are extracted, together with the

lower and upper bounds of a 95% confidence band. The respective function is `TEint`. The function `Ineff` extracts the point estimates of inefficiency, $\mathsf{E}(u_i|\epsilon_i)$. These results are labeled and appended to the object using `Renew`. The original database together with the transformed variables and results is saved to file via `Save`.

```
fob.Renew(fob.TEint(0.05), {"TE", "lower", "upper"});
fob.Renew(fob.Ineff(), {"jlms"});
fob.Save("out.xls");
```

There is a graphical functionality involving the package **GnuDraw** that allows for a visual assessment of the efficiency scores. The function `TestGraphicAnalysis` displays the graphics presented in Figure 2. The confidence band can be changed with `SetConfidenceLevel`, where an error probability of 0.05 is the default.

```
fob.SetConfidenceLevel(0.05);
fob.TestGraphicAnalysis();
```

The output of this program appears as follows (omitting information on the maximization procedure). Some general information:

```
SFAMB package version 1.1, object created on 19-02-2014
Constructing Squares and Cross-Products...done.
-Pooled model-

---- SFAMB ----
The estimation sample is:  1 - 2400
The dependent variable is: lny
The dataset is: USDAafrica.xls
```

The transformed variables facilitate the interpretation of the estimated coefficients of the translog functional form. Thus, the first order coefficients listed below can be interpreted as output elasticities at the sample mean. These estimates are positive and meet the requirement of monotonicity – except for the machinery input whose (insignificant) estimate violates the regularity condition. The parameter associated with `trend` indicates the estimated average rate of technical change per year.

| | Coefficient | Std.Error | robust-SE | t-value | t-prob |
|---|---|---|---|---|---|
| Constant | 0.418510 | 0.01734 | 0.01604 | 26.1 | 0.000 |
| lnlab | 0.128543 | 0.01338 | 0.01105 | 11.6 | 0.000 |
| lnland | 0.747665 | 0.01552 | 0.01301 | 57.5 | 0.000 |
| lnmac | -0.0103601 | 0.009487 | 0.008851 | -1.17 | 0.242 |
| lnfert | 0.0753082 | 0.006573 | 0.006243 | 12.1 | 0.000 |
| trend | 0.0104214 | 0.0007006 | 0.0006763 | 15.4 | 0.000 |

Furthermore, the output shows the coefficients of the squared and cross terms that can be used to calculate the individual output elasticities.

| | Coefficient | Std.Error | robust-SE | t-value | t-prob |
|---|---|---|---|---|---|
| .5*lnlab^2 | -0.0555284 | 0.02432 | 0.02387 | -2.33 | 0.020 |

```
.5*lnland^2          -0.170593    0.02547    0.02843    -6.00    0.000
.5*lnmac^2           -0.0152333   0.005151   0.004632   -3.29    0.001
.5*lnfert^2           0.0611977   0.003107   0.003063    20.0    0.000
.5*trend^2         0.000420193 6.481e-005 6.132e-005    6.85    0.000
lnlab*lnland          0.189011    0.02492    0.02557     7.39    0.000
lnlab*lnmac          -0.125612    0.008138   0.007344   -17.1    0.000
lnlab*lnfert         -0.0294981   0.006109   0.005248   -5.62    0.000
lnlab*trend       -0.000443318  0.0007217  0.0006231  -0.711    0.477
lnland*lnmac          0.137893    0.008829   0.008381    16.5    0.000
lnland*lnfert        -0.0633867   0.006748   0.006383   -9.93    0.000
lnland*trend      -0.000495235  0.0007838  0.0007483  -0.662    0.508
lnmac*lnfert         -0.0135746   0.002997   0.002857   -4.75    0.000
lnmac*trend           0.000810357 0.0002892  0.0002743   2.95    0.003
lnfert*trend          0.000898471 0.0002366  0.0002062   4.36    0.000
```

After the technology parameters, the estimates of $\sigma_v$ and $\sigma_u$ are listed in the form of their natural logarithms. The next line refers to the noise component.

```
                  Coefficient  Std.Error  robust-SE  t-value  t-prob
ln{\sigma_v}         -2.64681     0.1459     0.1361    -19.4   0.000
```

Since $\ln(\sigma_u)$ is parameterized using covariates, there are several related estimates. The order of coefficients corresponds to the specification $\ln(\sigma_u) = \delta_0 + \sum_{l=1}^{4} \delta_l \times z_l$ where $l = 1$ (labor), 2 (land), 3 (machinery), 4 (fertilizer); and the $z$'s are in logarithms. Higher use of $z_l$ is associated with a lower level of inefficiency (or higher technical efficiency) if the estimated parameter has a negative sign.

```
                  Coefficient  Std.Error  robust-SE  t-value  t-prob
Constant             -1.04439    0.04104    0.04790    -21.8   0.000
lnlab                 0.232702   0.04300    0.05044     4.61   0.000
lnland               -0.146200   0.04176    0.05050    -2.90   0.004
lnmac             -0.00976576    0.01491    0.01671   -0.584   0.559
lnfert               -0.0149142   0.01372    0.01647   -0.905   0.365
```

Here, the inefficiency distribution is supposed to have a non-zero mean, $u_i \sim N^+(\mu = \mu_0, \sigma^2_{u,i})$, i.e., the location parameter is a constant ($\mu_0$) common to all individuals. Additional covariates could be introduced. The omission of `U_VAR` in the model specification leads to $\mu = 0$, and hence, results in the normal half-normal model. Note that this estimate (here, the third `Constant`) is always the last `Constant` term in the list (if a truncated-normal is specified).

```
                  Coefficient  Std.Error  robust-SE  t-value  t-prob
Constant             0.454144    0.02926    0.03249     14.0   0.000
```

Some additional information is provided, for details see Section 3.3.

```
log-likelihood    -458.928611
no. of observations      2400  no. of parameters          28
AIC.T              973.857222  AIC               0.405773842
```

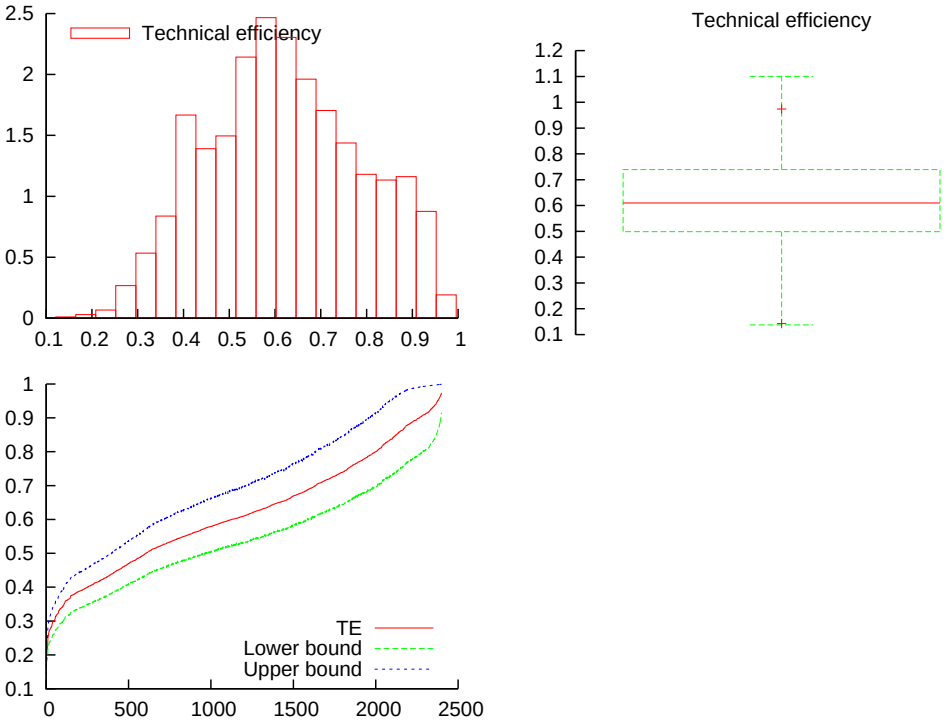


Figure 2: TE scores of the `POOLED` model.

```
mean(lny)              -1.14273  var(lny)               2.98932
\gamma:                  0.9618  VAR(u)/VAR(total)      0.9016
Test of one-sided err   172.93  mixed Chi^2 !!
with p-value        6.246e-035
```

Finally, the graph created by the function `TestGraphicAnalysis` is depicted in Figure 2.

## 4.2. Example: `hbest2.ox`

In this example, the `CFE` model of Chen *et al.* (2014) is specified using, again, the data set `USDAafrica.xls` and a translog functional form. You can immediately switch to the `LSDV` or `TFE` model, respectively, by changing the argument of `SetMethod`. A large part of this example corresponds to the code of the previous subsection. However, as panel data are involved here some things are different.

```
#include <oxstd.h>
#include <packages/gnudraw/gnudraw.h>
#import  <packages/sfamb/sfamb>

main(){
  decl fob = new Sfa();
  fob.Load("USDAafrica.xls");
  fob.SetMethod(CFE);
  fob.SetConstant();
```

CFE is the estimator selected. Here, the function SetConstant does not create a constant because it is not required. However, this line can be kept for convenience. The function Ident identifies the panel structure of the data. The required information includes the variable names of the individuals ("ID") and the period ("time").

```
  fob.Ident(fob.GetVar("ID"), fob.GetVar("time"));
```

Data transformation and model specification correspond to the previous example. Note that neither U_VAR nor Z_VAR are available here.

```
  decl inorm = 1;
  fob.Renew(fob.PrepData(fob.GetVar("output"), inorm), "lny");
  fob.Renew(fob.PrepData(fob.GetVar("labour"), inorm), "lnlab");
  fob.Renew(fob.PrepData(fob.GetVar("land"), inorm), "lnland");
  fob.Renew(fob.PrepData(fob.GetVar("machinery"), inorm), "lnmac");
  fob.Renew(fob.PrepData(fob.GetVar("fertilizer"), inorm), "lnfert");
  fob.Renew(fob.GetVar("time") - meanc(fob.GetVar("time")), "trend");

  fob.Select(Y_VAR, {"lny", 0, 0});

  fob.Select(X_VAR, {
    "Constant", 0, 0,
    "lnlab", 0, 0,
    "lnland", 0, 0,
    "lnmac", 0, 0,
    "lnfert", 0, 0,
    "trend", 0, 0});

  fob.SetSelSample(-1, 1, -1, 1);
  fob.SetPrintSfa(TRUE);
  MaxControl(1000, 10, TRUE);
  fob.SetTranslog(1);
  fob.Estimate();
```

For this model, there is no calculation of the confidence bounds involved. The efficiency scores can be extracted as point estimates using function TE.

```
  fob.Renew(fob.TE(), {"TE"});
  fob.Renew(fob.Ineff(), {"jlms"});
  fob.TestGraphicAnalysis();

  delete fob;
}
```

The output of this program appears as follows. Additional information on the panel structure is printed.

```
SFAMB package version 1.1, object created on 19-02-2014
        #groups:   #periods(max):  avg.T-i:
```

```
         48.000       50.000       50.000
Constructing Squares and Cross-Products...done.
-CFE model-

---- SFAMB ----
The estimation sample is:  1 - 2400
The dependent variable is: lny
The dataset is: USDAafrica.xls
```

A common intercept is not identified, and hence, there is no `Constant`.

```
                 Coefficient  Std.Error  t-value  t-prob
lnlab              0.00883652    0.03048    0.290   0.772
lnland               0.677192    0.02304     29.4   0.000
lnmac                0.106177   0.009083     11.7   0.000
lnfert              0.0837343   0.007086     11.8   0.000
trend               0.00920993  0.0006800     13.5   0.000
.5*lnlab^2            0.138565    0.02083     6.65   0.000
.5*lnland^2           0.177254    0.02047     8.66   0.000
.5*lnmac^2           0.0121082   0.003350     3.61   0.000
.5*lnfert^2          0.0245012   0.002852     8.59   0.000
.5*trend^2         0.000407978 3.744e-005     10.9   0.000
lnlab*lnland         -0.138300    0.02024    -6.83   0.000
lnlab*lnmac         -0.0247345   0.007611    -3.25   0.001
lnlab*lnfert         0.00218990   0.005678    0.386   0.700
lnlab*trend       -0.000134440  0.0005109   -0.263   0.792
lnland*lnmac         0.0243333   0.008190     2.97   0.003
lnland*lnfert       -0.0319551   0.006178    -5.17   0.000
lnland*trend       0.000212194  0.0004843    0.438   0.661
lnmac*lnfert         0.00379000   0.001959     1.93   0.053
lnmac*trend        0.000346355  0.0001844     1.88   0.060
lnfert*trend      -0.000171510  0.0001308    -1.31   0.190
```

This model is restricted to the normal half-normal case. Here, the estimates of (the natural logarithms of) $\sigma_v^2$ and $\sigma_u^2$ are given.

```
ln{\sigma_v^2}       -4.94563     0.1464    -33.8   0.000
ln{\sigma_u^2}       -3.44008     0.1125    -30.6   0.000

log-likelihood     1476.81739
no. of observations      2400  no. of parameters          22
AIC.T             -2909.63478  AIC               -1.21234782
mean(lny)        7.55183e-018  var(lny)             0.127523
lambda                  2.123
```

The function `TestGraphicAnalysis` is used to create the graph depicted in Figure 3.

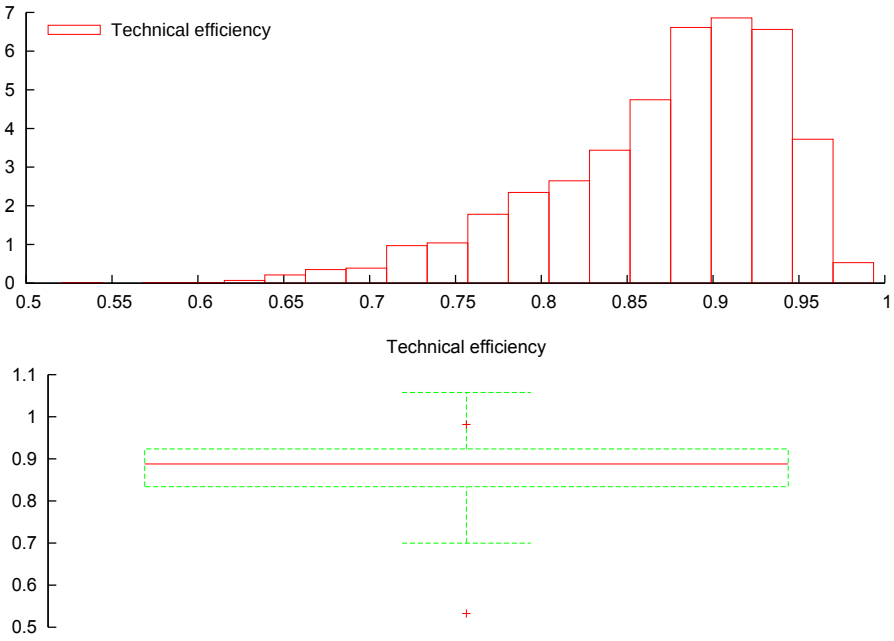


Figure 3: TE scores of the `CFE` model.

### 4.3. Example: Member functions `SetTranslog` and `Elast`

The member function `SetTranslog` allows for convenient specification of a translog functional form. In the following excerpt, we refer to the current instance of the class as `fob`. Suppose your selection of regressors looks like this:

```
fob.Select(X_VAR, {
  "Constant", 0, 0,
  "lnx1", 0, 0,
  "lnx2", 0, 0,
  "lnx3", 0, 0,
  "trend", 0, 0});
```

The default specification is Cobb-Douglas, i.e., `SetTranslog(0)`, changing the argument to 1 invokes construction of the respective square and cross terms of `X_VAR`. In general notation:

$$\ln y_i = \beta_0 + \sum_{j=1}^{K} \beta_j \ln x_{ji} + \frac{1}{2} \sum_{j=1}^{K} \sum_{l=1}^{K} \beta_{jl} \ln x_{ji} \ln x_{li}.$$

If your selection includes dummies, the variables should be ordered like this:

```
fob.Select(X_VAR, {
  "Constant", 0, 0,
  "lnx1", 0, 0,
  "lnx2", 0, 0,
  "lnx3", 0, 0,
  "trend", 0, 0,
```

```
  "dummy1", 0, 0,
  "dummy2", 0, 0});
```

Specification of a translog form is then possible by means of `SetTranslog(4)` because only the first four regressors are used (`"Constant"` is ignored automatically).

After estimation, the member function `Elast` can be used to calculate the output elasticity ($\epsilon_{ji}$) of each input for each observation:

$$\epsilon_{ji} = \beta_j + \sum_{l=1}^{K} \beta_{jl} \ln x_{li}.$$

The following example illustrates one possible way the function may be used. Here, results are plotted as histograms (see Figure 4). Note that indexing starts at 0 in Ox (`Elast` returns an $NT \times 2$ matrix but only the first column is considered here). The first three arguments of `DrawDensity` are the most important here: area (panel) index, variable, label. See the documentations of Ox or **GnuDraw** for a full description.

```
decl vEps1 = fob.Elast("lnx1");
decl vEps2 = fob.Elast("lnx2");
decl vEps3 = fob.Elast("lnx3");
decl vEpst = fob.Elast("trend");

DrawDensity(0, vEps1[][0]', {"eps1"}, 1, 1, 0, 0, 0, 0, 1, 0, 1);
DrawDensity(1, vEps2[][0]', {"eps2"}, 1, 1, 0, 0, 0, 0, 1, 0, 1);
DrawDensity(2, vEps3[][0]', {"eps3"}, 1, 1, 0, 0, 0, 0, 1, 0, 1);
DrawDensity(3, vEpst[][0]', {"epst"}, 1, 1, 0, 0, 0, 0, 1, 0, 1);
ShowDrawWindow();
```

## 5. Future developments

The basic version of **SFAMB** dates back to the mid 1990s where the capability was restricted to cross-sectional data. As the package now allows for panel data and the literature on SF methods is considerably broader and still growing, there is scope for potential extensions. Some related possibilities are mentioned here.

In the model framework of Chen *et al.* (2014) there are two ways to calculate the individual effects. As an alternative to Equation 5, the individual "*between estimator of* $\alpha_i$" can be used. It could be implemented as an optional function, involving a second maximization. Its availability would allow us to compare results and check the consequences for TE scores.

While the current focus of panel methods is on fixed effects estimation, a more comprehensive supplement might involve random effects models. The most recent SF approach using the CSN distribution is presented by Colombi, Kumbhakar, Martini, and Vittadini (2014). Its specification is similar to Equation 3, but the time-invariant part is further decomposed into two residuals (persistent inefficiency and time-invariant unobserved heterogeneity). Filippini and Greene (2016) introduce computational simplifications and label the model as the "*Generalized True Random Effects SF model*".

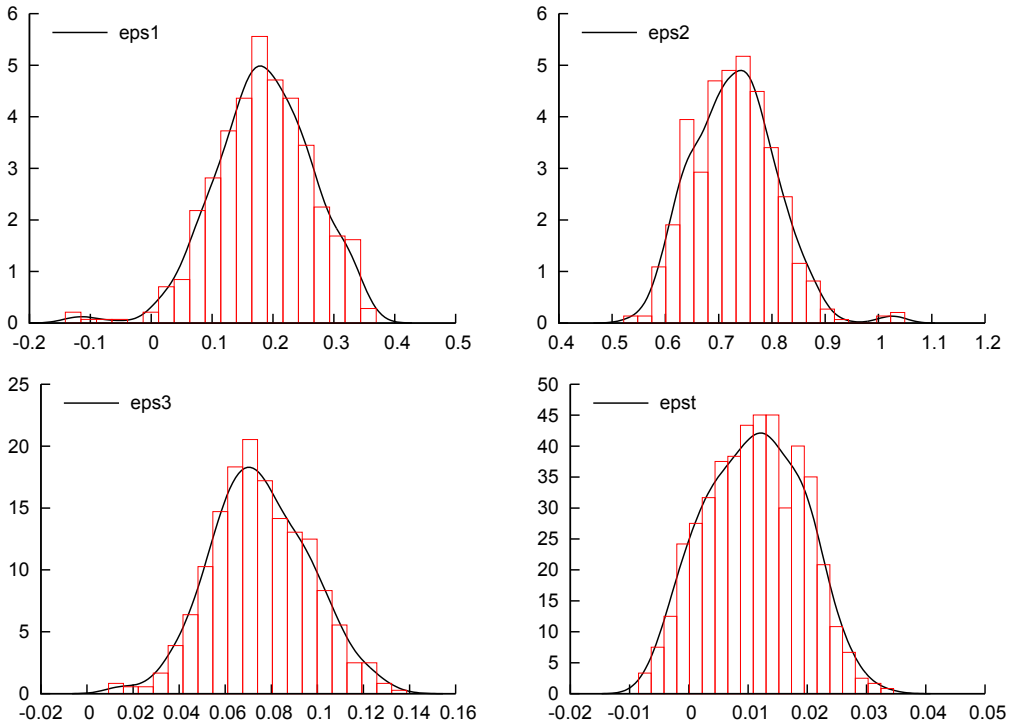


Figure 4: Histograms of calculated elasticities (by observation).

## Acknowledgments

All code is pure Ox code. However, code of the `WT` model is partially adapted from Stata code by Hung-Jen Wang. We thank Hung-Jen Wang for providing data to check our `CFE` code. Additional thanks are due for the referees for offering helpful comments that improved the quality of the manuscript.

## References

Aigner D, Lovell CAK, Schmidt P (1977). "Formulation and Estimation of Stochastic Frontier Production Function Models." *Journal of Applied Econometrics*, **6**(1), 21–37. doi:10.1016/0304-4076(77)90052-5.

Alvarez A, Amsler C, Orea L, Schmidt P (2006). "Interpreting and Testing the Scaling Property in Models Where Inefficiency Depends on Firm Characteristics." *Journal of Productivity Analysis*, **25**(3), 201–212. doi:10.1007/s11123-006-7639-3.

Battese GE, Coelli TJ (1988). "Prediction of Firm-Level Technical Efficiencies with a Generalized Frontier Production Function and Panel Data." *Journal of Econometrics*, **38**(3), 387–399. doi:10.1016/0304-4076(88)90053-x.

Battese GE, Coelli TJ (1992). "Frontier Production Functions, Technical Efficiency and Panel Data: With Application to Paddy Farmers in India." *Journal of Productivity Analysis*, **3**(1–2), 153–169. doi:10.1007/bf00158774.

Battese GE, Coelli TJ (1995). "A Model for Technical Inefficiency Effects in a Stochastic Frontier Production Function for Panel Data." *Empirical Economics*, **20**(2), 325–332. doi:10.1007/bf01205442.

Battese GE, Corra GS (1977). "Estimation of a Production Function Model: With Application to the Pastoral Zone of Eastern Australia." *Australian Journal of Agricultural Economics*, **21**(3), 169–179. doi:10.1111/j.1467-8489.1977.tb00204.x.

Belotti F, Daidone S, Ilardi G, Atella V (2013). "Stochastic Frontier Analysis Using Stata." *Stata Journal*, **13**(4), 719–758. URL http://www.stata-journal.com/article.html?article=st0315.

Bogetoft P, Otto L (2010). *Benchmarking with DEA, SFA, and R*, volume 157 of *International Series in Operations Research & Management Science*. Springer-Verlag.

Bos CS (2014). ***GnuDraw*** *– An Ox Package for Creating* ***gnuplot*** *Graphics*. URL http://personal.vu.nl/c.s.bos/software/gnudraw.html.

Brorsen BW, Kim T (2013). "Data Aggregation in Stochastic Frontier Models: The Closed Skew Normal Distribution." *Journal of Productivity Analysis*, **39**(1), 27–34. doi:10.1007/s11123-012-0274-2.

Brümmer B (2001). "Estimating Confidence Intervals for Technical Efficiency: The Case of Private Farms in Slovenia." *European Review of Agricultural Economics*, **28**(3), 285–306. doi:10.1093/erae/28.3.285.

Caudill SB, Ford JM, Gropper DM (1995). "Frontier Estimation and Firm-Specific Inefficiency Measures in the Presence of Heteroscedasticity." *Journal of Business & Economic Statistics*, **13**(1), 105–111. doi:10.2307/1392525.

Chen YY, Schmidt P, Wang HJ (2014). "Consistent Estimation of the Fixed Effects Stochastic Frontier Model." *Journal of Econometrics*, **181**(2), 65–76. doi:10.1016/j.jeconom.2013.05.009.

Coelli TJ (1996). *A Guide to* ***FRONTIER 4.1****: A Computer Program for Stochastic Frontier Production and Cost Function Estimation*. CEPA Working Papers, University of New England, URL http://www.uq.edu.au/economics/cepa/frontier.php.

Coelli TJ, Henningsen A (2017). ***frontier****: Stochastic Frontier Analysis*. R package version 1.1-2, URL https://CRAN.R-Project.org/package=frontier.

Coelli TJ, Rao PDS, O'Donnell CJ, Battese GE (2005). *An Introduction to Efficiency and Productivity Analysis*. Springer-Verlag. doi:10.1007/978-1-4615-5493-6.

Colombi R, Kumbhakar SC, Martini G, Vittadini G (2014). "Closed-Skew Normality in Stochastic Frontiers with Individual Effects and Long/Short-Run Efficiency." *Journal of Productivity Analysis*, **42**(2), 123–136. doi:10.1007/s11123-014-0386-y.

Cottrell A, Lucchetti R (2014). ***gretl*** *User's Guide – Gnu Regression, Econometrics and Time-Series Library*. URL http://gretl.sourceforge.net/.

Doornik JA (2009). *An Object-Oriented Matrix Language Ox 6*. Timberlake Consultants Press, London.

Doornik JA, Ooms M (2007). *Introduction to Ox: An Object-Oriented Matrix Language*. Timberlake Consultants Press, London. Available at http://www.doornik.com/ox/OxIntro.pdf.

Econometric Software Inc (2014). ***LIMDEP***, *Version 10.0*. ESI, New York. URL http://www.limdep.com/.

Filippini M, Greene WH (2016). "Persistent and Transient Productive Inefficiency: A Maximum Simulated Likelihood Approach." *Journal of Productivity Analysis*, **45**(2), 187–196. doi:10.1007/s11123-015-0446-y.

Fuglie KO (2012). "Productivity Growth and Technology Capital in the Global Agricultural Economy." In KO Fuglie, SL Wang, VE Ball (eds.), *Productivity Growth in Agriculture: An International Perspective*. CABI.

**gnuplot** Team (2015). ***gnuplot 5.0*** – *An Interactive Plotting Program*. URL http://sourceforge.net/projects/gnuplot.

Greene WH (2005). "Reconsidering Heterogeneity in Panel Data Estimators of the Stochastic Frontier Model." *Journal of Econometrics*, **126**(2), 269–303. doi:10.1016/j.jeconom.2004.05.003.

Greene WH (2008). "The Econometric Approach to Efficiency Analysis." In HO Fried, CAK Lovell, SS Schmidt (eds.), *The Measurement of Productive Efficiency and Productivity Growth*. Oxford University Press.

Greene WH (2012). *Econometric Analysis*. 7th edition. Pearson International Edition.

Horrace WC, Schmidt P (1996). "Confidence Statements for Efficiency Estimates from Stochastic Frontier Models." *Journal of Productivity Analysis*, **7**(2–3), 257–282. doi:10.1007/bf00157044.

Huang CJ, Liu JT (1994). "Estimation of a Non-Neutral Stochastic Frontier Production Function." *Journal of Productivity Analysis*, **5**(2), 171–180. doi:10.1007/bf01073853.

Hughes G (2008). ***sfa_hetmod*** *and* ***sfa_mod***. User-Contributed Function Packages for **gretl**. URL https://gretlwiki.econ.univpm.it/wiki/index.php/List_of_available_user-contributed_function_packages.

Jondrow J, Lovell CAK, Materov IS, Schmidt P (1982). "On the Estimation of Technical Inefficiency in the Stochastic Frontier Production Function Model." *Journal of Econometrics*, **19**(2–3), 233–238. doi:10.1016/0304-4076(82)90004-5.

Kodde DA, Palm FC (1986). "Wald Criteria for Jointly Testing Equality and Inequality Restrictions." *Econometrica*, **54**(5), 1243–1248. doi:10.2307/1912331.

Kotz S, Balakrishnan N, Johnson NL (2000). *Continuous Multivariate Distributions: Models and Applications*, volume 1. John Wiley & Sons. doi:10.1002/0471722065.

Kumbhakar SC, Gosh S, McGuckin JT (1991). "A Generalized Production Frontier Approach for Estimating Determinants of Inefficiency in U.S. Dairy Farms." *Journal of Business & Economic Statistics*, **9**(3), 279–286. doi:10.2307/1391292.

Kumbhakar SC, Lovell CAK (2000). *Stochastic Frontier Analysis*. Cambridge University Press, Cambridge.

Lai H, Huang CJ (2010). "Likelihood Ratio Tests for Model Selection of Stochastic Frontier Models." *Journal of Productivity Analysis*, **34**(1), 3–13. doi:10.1007/s11123-009-0160-8.

Meeusen W, Van den Broeck J (1977). "Efficiency Estimation from Cobb-Douglas Production Functions with Composed Error." *International Economic Review*, **18**(2), 435–444. doi:10.2307/2525757.

Pavlyuk D (2016). ***spfrontier**: Spatial Stochastic Frontier Models Estimation*. R package version 0.2.3, URL https://CRAN.R-project.org/package=spfrontier.

Piessens R, de Doncker-Kapenga E, Überhuber CW, Kahaner DK (1983). ***QUADPACK**, A Subroutine Package for Automatic Integration*. Springer-Verlag.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Reifschneider D, Stevenson R (1991). "Systematic Departures from the Frontier: A Framework for the Analysis of Firm Inefficiency." *International Economic Review*, **32**(3), 715–723. doi:10.2307/2527115.

Schmidt P, Sickles RC (1984). "Production Frontiers and Panel Data." *Journal of Business & Economic Statistics*, **2**(4), 367–374. doi:10.2307/1391278.

StataCorp LP (2015). *Stata, Version 14*. College Station. URL http://www.stata.com/.

Wang HJ (2012). *Manual of Hung-Jen Wang's Stata Codes*. URL http://homepage.ntu.edu.tw/~wangh.

Wang HJ, Ho CW (2010). "Estimating Fixed-Effect Panel Stochastic Frontier Models by Model Transformation." *Journal of Econometrics*, **157**(2), 286–296. doi:10.1016/j.jeconom.2009.12.006.

Wang HJ, Schmidt P (2002). "One-Step and Two-Step Estimation of the Effects of Exogenous Variables on Technical Efficiency Levels." *Journal of Productivity Analysis*, **18**(2), 129–144. doi:10.1023/a:1016565719882.

White H (1980). "A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity." *Econometrica*, **48**(4), 817–838. doi:10.2307/1912934.

# A. Technical appendix

## A.1. Starting values

OLS estimates are used as starting values for the technology parameters $\boldsymbol{\beta}$, and a grid search is applied to find an appropriate value for $\sigma^2 = \sigma_v^2 + \sigma_u^2$. Battese and Corra (1977, p. 173) point out that the OLS estimates of $\boldsymbol{\beta}$ are unbiased, except for the common constant $\beta_0$. They show that $\beta_0$ can be corrected as:

$$\hat{\beta}_0 = \hat{\beta}_0^{\text{OLS}} + \sqrt{\frac{2}{\pi}}\,\hat{\sigma}_u.$$

Furthermore, they define $\gamma = \sigma_u^2/(\sigma_v^2 + \sigma_u^2)$, and suggest to try different initial values. The grid search evaluates the likelihood function over a range of values ($\gamma$ $[0.05, 0.98]$) and chooses the parameters associated with the highest likelihood value. Within this procedure, $\sigma_v^2$ and $\sigma_u^2$ are parameterized as:[15]

$$\sigma_v^2 = \sigma^2 \times (1-\gamma) = \left(\sigma_v^2 + \sigma_u^2\right) \times \left(\frac{\sigma_v^2 + \sigma_u^2}{\sigma_v^2 + \sigma_u^2} - \frac{\sigma_u^2}{\sigma_v^2 + \sigma_u^2}\right),$$
$$\sigma_u^2 = \sigma^2 \times \gamma = \left(\sigma_v^2 + \sigma_u^2\right) \times \frac{\sigma_u^2}{\sigma_v^2 + \sigma_u^2}.$$

The search for values for $\sigma_v^2$ and $\sigma_u^2$ involves partitioning the variance of the composed error term. Aigner *et al.* (1977) show that $\mathsf{VAR}(\epsilon) = \sigma_v^2 + ((\pi - 2)/\pi)\,\sigma_u^2$, which can be expressed as:

$$\mathsf{VAR}(\epsilon) = \sigma^2(1-\gamma) + \frac{\pi - 2}{\pi}\sigma^2\gamma = \sigma^2\left(1 - \gamma\left(1 - \frac{\pi - 2}{\pi}\right)\right).$$

Using the variance of the OLS residuals ($m_2$) as an estimate for $\mathsf{VAR}(\epsilon)$:

$$m_2 = \sigma^2\left(1 - \gamma\frac{2}{\pi}\right) \quad \longleftrightarrow \quad \sigma^2 = \frac{m_2}{1 - \gamma\frac{2}{\pi}}.$$

The grid search is called within member function `DoEstimation`, and runs over $\gamma$ values with step length 0.01; it passes the starting values for $\boldsymbol{\beta}$ and $\sigma^2$ back to `DoEstimation`.

## A.2. Estimation and standard errors

After the starting values are obtained from the OLS regression and grid search, the log-likelihood function is maximized using the BFGS (Broyden-Fletcher-Goldfarb-Shanno) algorithm. The respective Ox routine is named `MaxBFGS` and documented in Doornik (2009) and online at `http://www.doornik.com/ox/`. The log-likelihood function can be found as function `fSfa` in the source code.[16] Analytical first derivatives are used in the case of the `POOLED`

[15] In case of the pooled model, the parameters are $\sigma_v$ and $\sigma_u$. In case of the panel models, there is no constant ($\beta_0$) to be corrected. If the model includes additional covariates ($z$-variables, related to inefficiency), a vector of zeros is used for the respective parameters. Zeros are also used as starting values for the individual effects in the `TFE` model.

[16] Member functions of the source code are structured by `case`(s) where `default = POOLED`, `1 = WT`, `2 = LSDV`, `3 = CFE`, `4 = TFE`.

model, while the remaining models employ numerical first derivatives based on the Ox routine `Num1Derivative` (finite difference approximation).

The `CFE` model's "within-likelihood function" includes a $T$-dimensional cdf. In their Appendix C, Chen *et al.* (2014) show how the $T$-dimensional integral is reduced to a one-dimensional integral, referring to Kotz, Balakrishnan, and Johnson (2000) (`fKotzetal` is the respective function). The numerical quadrature is executed by function `QNG` which is part of the Fortran package **QuadPack** (Piessens, de Doncker-Kapenga, Überhuber, and Kahaner 1983) and linked in via `#include <quadpack.h>`.

Standard errors are obtained from `m_mCovar` (covariance matrix). This data member is produced by function `Num2Derivative` (called within member function `Covar`) that uses a finite difference approximation (Doornik 2009). Estimation output of the cross-sectional model returns robust standard errors (by default) which are obtained by the method of White (1980).

## A.3. Log-likelihood functions

In this section, $\phi$ and $\Phi$ denote the pdf and cdf of a standard normal distribution, respectively; $\lambda = \sigma_u/\sigma_v$ and $\sigma^2 = \sigma_u^2 + \sigma_v^2$.

Kumbhakar and Lovell (2000) present the log-likelihood functions of the normal-truncated normal model ($u_i \overset{\text{iid}}{\sim} N^+(\mu, \sigma_u^2)$) and the normal-half normal model ($\mu = 0$) for cross-sectional data. The log-likelihood function of the `POOLED` model for one observation, with $\epsilon_i = y_i - \boldsymbol{\beta}^\top \boldsymbol{x}_i$, is given by:

$$\ln L_i = \ln \frac{1}{\sqrt{2\pi}} - \frac{1}{2} \ln \sigma^2 - \frac{(\epsilon_i + \mu)^2}{2\sigma^2} + \ln \Phi\left(\frac{\mu}{\sigma\lambda} - \frac{\lambda\epsilon_i}{\sigma}\right) - \ln \Phi\left(\mu \frac{\sqrt{1+\lambda^2}}{\sigma\lambda}\right). \tag{6}$$

The log-likelihood function of the `TFE` model (Greene 2005) corresponds to Equation 6, but with $\mu = 0$ and $\epsilon_{it} = y_{it} - \boldsymbol{\beta}^\top \boldsymbol{x}_{it} - \alpha_i$ in the place of $\epsilon_i$.

The log-likelihood function of the `WT` model (Wang and Ho 2010) for one individual, with $\tilde{\epsilon}_{it} = \tilde{y}_{it} - \boldsymbol{\beta}^\top \tilde{\boldsymbol{x}}_{it}$, and $\tilde{\boldsymbol{\epsilon}}_i = (\tilde{\epsilon}_{i1}, \ldots, \tilde{\epsilon}_{iT})^\top$:

$$\begin{aligned} \ln L_i = -\frac{1}{2}(T-1)\ln(2\pi) - \frac{1}{2}(T-1)\ln(\sigma_v^2) - \frac{1}{2}\tilde{\boldsymbol{\epsilon}}_i^\top \Pi^- \tilde{\boldsymbol{\epsilon}}_i \\ + \frac{1}{2}\left(\frac{\mu_{**}^2}{\sigma_{**}^2} - \frac{\mu^2}{\sigma_u^2}\right) + \ln\left(\sigma_{**}\Phi\left(\frac{\mu_{**}}{\sigma_{**}}\right)\right) - \ln\left(\sigma_u \Phi\left(\frac{\mu}{\sigma_u}\right)\right), \end{aligned}$$

where

$$\begin{aligned} \mu_{**} &= \frac{\mu/\sigma_u^2 - \tilde{\boldsymbol{\epsilon}}_i^\top \Pi^- \tilde{\boldsymbol{h}}_i}{\tilde{\boldsymbol{h}}_i^\top \Pi^- \tilde{\boldsymbol{h}}_i + 1/\sigma_u^2}, \\ \sigma_{**}^2 &= \frac{1}{\tilde{\boldsymbol{h}}_i^\top \Pi^- \tilde{\boldsymbol{h}}_i + 1/\sigma_u^2}. \end{aligned}$$

Log-likelihood function of the `CFE` model (Chen *et al.* 2014), with $\tilde{\epsilon}_{it} = \tilde{y}_{it} - \boldsymbol{\beta}^\top \tilde{\boldsymbol{x}}_{it}$, $\tilde{\boldsymbol{\epsilon}}_i =$

$(\tilde{\epsilon}_{i1}, \ldots, \tilde{\epsilon}_{iT})^\top$, and $\tilde{\boldsymbol{\epsilon}}_i^* = (\tilde{\epsilon}_{i1}, \ldots, \tilde{\epsilon}_{i,T-1})^\top$:

$$\ln L_W = \text{constant} + \sum_i \left[ \ln \phi_{T-1} \left( \tilde{\boldsymbol{\epsilon}}_i^*; 0, \sigma^2 (I_{T-1} - \frac{1}{T} E_{T-1}) \right) \right] + \sum_i \left[ \ln \Phi_T \left( -\frac{\lambda}{\sigma} \tilde{\boldsymbol{\epsilon}}_i; 0_T, I_T + \frac{\lambda^2}{T} E_T \right) \right],$$

where $I_n$ is an $n \times n$ identity matrix and $E_n$ is an $n \times n$ matrix of ones; or alternatively:

$$\ln L_W = -NT \ \ln \Phi(0) + \sum_i \left[ \left( -\frac{(T-1)}{2} \right) \ln (2\pi) - \frac{1}{2} \ln |\Sigma| - \left( \frac{1}{2} \right) \tilde{\boldsymbol{\epsilon}}_i^{*\top} \Sigma^- \tilde{\boldsymbol{\epsilon}}_i^* \right] + \sum_i \ln \left[ \int_{-\infty}^{\infty} \phi(u_0) \prod_{t=1}^{T} \Phi \left( -\frac{\lambda}{\sigma} \tilde{\epsilon}_{it} - \frac{\lambda}{\sqrt{T}} U_0 \right) du_0 \right].$$

**Affiliation:**

Jonathan Holtkamp
Department of Agricultural Economics and Rural Development
University of Goettingen
D-37073 Goettingen, Germany
E-mail: jonathan-holtkamp@web.de

Bernhard Brümmer
Department of Agricultural Economics and Rural Development
& Centre of Biodiversity and Sustainable Land Use
University of Goettingen
D-37073 Goettingen, Germany
E-mail: bbruemm@gwdg.de
URL: http://www.uni-goettingen.de/en/19255.html

*Journal of Statistical Software* http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
November 2017, Volume 81, Issue 6 *Submitted:* 2014-05-04
doi:10.18637/jss.v081.i06 *Accepted:* 2016-08-07