# Production Function Estimation in R: The prodest Package

**Gabriele Rovigatti**
University of Chicago
Booth School of Business

**Abstract**

This paper presents the R package **prodest** for production function estimation using the control function approach. Focusing on the Value Added PF, it provides functions to estimate two–steps models presented by Olley and Pakes (1996) and Levinsohn and Petrin (2003), as well as their correction proposed by Ackerberg et al. (2015). The system GMM framework proposed by Wooldridge (2009) is also implemented and tested in a series of Monte Carlo exercises. The **prodest** package features the DGP used by Ackerberg et al. (2015) and allows for the simulation of datasets according to various measurement errors / random shock variances. I illustrate the package use with an application to a popular firm–level dataset.

*Keywords*: production functions, productivity, prodest, R software .

# 1. Introduction

The correct estimation of the total factor productivity (TFP) is a fundamental issue in applied economics and the main topic of several seminal papers. On the one hand, when subject to positive productivity shocks, firms respond by both expanding their level of output and demanding more input; negative shocks, on the other hand, lead to a decline in output and demand for input. The positive correlation between the observable input levels and the unobservable productivity shocks is a major source of bias in OLS when estimating the total factor productivity. Various methods have been proposed to tackle such simultaneity issue and, according to their approaches, is possible to group them in three families: Fixed Effects (FE), Instrumental Variables (IV) and Control Function (CF). In the latter group, Olley and Pakes (1996) - OP henceforth - are the first to propose a two-step procedure aimed at overcoming the endogeneity: they use the investment level to proxy productivity. Their approach has been refined by Levinsohn and Petrin (2003) - LP - and Ackerberg et al. (2015) - ACF. Wooldridge (2009) proposes a novel estimation setting that shows i) how to obtain LP estimator within a system GMM econometric framework, which can be estimated in a single step, and ii) the appropriate moment conditions.

A crucial assumption underlies both the dynamic profit maximization problem faced by the firm at each time $t$ and all the above models: the idiosyncratic shock to productivity at time $t$ (i.e., $\xi_t$) does not affect the choice of the level of state variables, which is taken at $t - b$[1], but only that of free variables. Therefore, $\xi_t$ is uncorrelated to the contemporaneous value of the state and to all the lagged values of the free and state variables; all of these are valid instruments for parameter identification. These, in turn, can be used in GMM–type estimation settings.

**prodest** is a brand new package that implements - for the first time on R - all the main models proposed in the literature so far, in a user-friendly and intuitive way. Dealing with Value Added models,[2] it allows users to perform (i) TFP estimation following OP, LP, ACF and Wooldridge methods, and (ii) simulation of production data according to the Data Generating Process firstly proposed by ACF. The package is written using `S4` classes, and provides several methods such as `coef()`, `summary()` and `show()`, alongside custom methods (`omega` and `FSres`) to extract and analyze the results.

The remainder of the paper is structured as follows: in Section 2 we review the control function approaches to TFP estimation in the literature, list their weaknesses and provide a general overview of the state of the art in empirical applications; Section 3 presents **prodest**, its main features, structure and functions; in Section 4 we present practical examples of the package usage, along with comparative results of the various models implemented on real and simulated data; Section 5 concludes.

# 2. Control function approach[3]

In this section we provide a brief but complete overview of the most common techniques for production function estimation using control function approach. For the remainder of the

[1] where $b > 0$ can take different values depending on state variable dynamics.
[2] Extensions to the Gross Output case are planned.
[3] This section is taken from Mollisi and Rovigatti (2017), currently under review at *The Stata Journal*.

paper, consider a Cobb-Douglas technology for firm $i$ at time $t$ :

$$y_{it} = \alpha + \mathbf{w}_{it}\beta + \mathbf{x}_{it}\gamma + \omega_{it} + \varepsilon_{it} \tag{1}$$

where $y_{it}$ is the log gross output , $\mathbf{w}_{it}$ is a $1 \times J$ vector of log free variables and $\mathbf{x}_{it}$ is a $1 \times K$ vector of log state variables. The random component $\omega_{it}$ is the unobservable productivity or technical efficiency and $\varepsilon_{it}$ is an idiosyncratic output shock distributed as white noise. We assume with OP and LP that productivity evolves according to a first-order Markov process:

$$\omega_{it} = E(\omega_{it}\,|\Omega_{it-1}) + \xi_{it} = E(\omega_{it}\,|\omega_{it-1}) + \xi_{it} = g(\omega_{it-1}) + \xi_{it} \tag{2}$$

where $\Omega_{it-1}$ is the information set at $t-1$ and $\xi_{it}$ is the productivity shock, assumed to be uncorrelated with productivity $\omega_t$ and with state variables $\mathbf{x}_{it}$.

## 2.1. Olley-Pakes method

OP were the first to propose a consistent two-step estimation procedure for (1). Their key idea is to exploit firm investment levels as a proxy variable for $\omega_{it}$. They prove their estimates of productivity to be consistent under several assumptions on top of those mentioned above:

A.1 $i_{it} = f(\mathbf{x}_{it}, \omega_{it})$ is the investment policy function, invertible in $\omega_{it}$. Moreover, $i_{it}$ is monotonically increasing in $\omega_{it}$;

A.2 The state variables - typically capital - evolve according to the investment policy function $i_{it}$ which is decided at time $t-1$;

A.3 The free variables $\mathbf{w}_{it}$ - typically labor inputs - are non-dynamic, in the sense that their choice at $t$ does not impact future profits, and are chosen at time $t$ after the firm productivity shock realizes.

Hence, given A.1 and A.2, the investment $i_{it}$ is orthogonal to the state variable in $t$ such that $E[i_{it}|\mathbf{x}_{it}] = 0$ and can be inverted, yielding the following proxy for productivity:

$$\omega_{it} = f^{-1}(i_{it}, \mathbf{x}_{it}) = h(i_{it}, \mathbf{x}_{it}) \tag{3}$$

which is an unknown function of observable variables. Plugging (3) in (1), we obtain:

$$\begin{aligned} y_{it} &= \alpha + \mathbf{w}_{it}\beta + \mathbf{x}_{it}\gamma + h(i_{it}, \mathbf{x}_{it}) + \varepsilon_{it} = \\ &= \mathbf{w}_{it}\beta + \Phi_{it}(i_{it}, \mathbf{x}_{it}) + \varepsilon_{it} \end{aligned} \tag{4}$$

where we define $\Phi_{it}(i_{it}, \mathbf{x}_{it}) = \mathbf{x}_{it}\gamma + h(i_{it}, \mathbf{x}_{it}) = \mathbf{x}_{it}\gamma + \omega_{it}$. Equation (4) is a partially linear model identified only in the free variable vector, $\mathbf{w}_{it}$ and can be non parametrically estimated approximating $\Phi_{it}(i_{it}, \mathbf{x}_{it})$ by a $n^{th}$ order polynomial $\hat{\Phi}$ or by a local linear regression (First Stage). This yields a consistent estimate of the free variables' parameters, $\hat{\beta}$. Using (2), then, it becomes possible to estimate $\gamma$ by rewriting the model for $y_{it} - \mathbf{w}_{it}\hat{\beta}$ conditional on $\mathbf{x}_{it}$:

$$\begin{aligned} y_{it} - \mathbf{w}_{it}\hat{\beta} &= \alpha_0 + \mathbf{x}_{it}\gamma + \omega_{it} + \varepsilon_{it} = \\ &= \alpha_0 + \mathbf{x}_{it}\gamma + E[\omega_{it}|\,\omega_{it-1}] + \xi_{it} + \varepsilon_{it} = \\ &= \alpha_0 + \mathbf{x}_{it}\gamma + g(\omega_{it-1}) + e_{it} \end{aligned} \tag{5}$$

where $e_{it} = \xi_{it} + \varepsilon_{it}$. Being $\hat{\omega}_{it} = \hat{\Phi}_{it} - \mathbf{x}_{it}\gamma$ equation (5) becomes:

$$y_{it} - \mathbf{w}_{it}\hat{\beta} = \alpha_0 + \mathbf{x}_{it}\gamma + g(\hat{\Phi}_{it-1} - \mathbf{x}_{it-1}\gamma) + e_{it} \tag{6}$$

where the function $g(.)$ can be left unspecified and estimated non parametrically. Alternatively, if we assume $g(\cdot)$ to follow a random walk we can restate equation (6) as:

$$y_{it} - \mathbf{w}_{it}\hat{\beta} = \alpha_0 + (\mathbf{x}_{it} - \mathbf{x}_{it-1})\,\gamma + \hat{\Phi}_{it-1} + e_{it} \tag{7}$$

and

$$e_{it} = y_{it} - \mathbf{w}_{it}\hat{\beta} - \alpha_0 - \mathbf{x}_{it}\gamma^* - g(\hat{\Phi}_{it-1} - \mathbf{x}_{it-1}\gamma) \tag{8}$$

at the true $\gamma^*$ value.
Equation (7) suggests an immediate approach to the estimation. In fact, residuals $e_{it}$ can be used to build a GMM estimator exploiting the moment conditions $E[e_{it}x_{it}^k]$=0, $\forall k$ (Second Stage)[4]. The $\gamma*$ vector is the vector of parameters which minimizes the criterion function:

$$\gamma* = \text{argmax}\left\{\sum_k \left(\sum_i \sum_t e_{it}x_{it}^k\right)^2\right\} \tag{9}$$

In their seminal paper OP discuss potential selection bias due to the non-randomness in plants dropping out the sample. More specifically, less productive firms could be forced out of the market exactly due to their low level of productivity, thus leaving only the most productive firms in the sample. They assume that a firm continues to operate provided that its productivity level exceeds the lower bound, i.e. $\chi_{it} = 1 \iff \omega_{it} \geq \underline{\omega}_{it}$, where $\chi_{it}$ is a survival binary variable and the $\underline{\omega}_{it}$ is industry-specific (see Hopenayn (1992) and Melitz (2003)). Hence, they propose a third step in estimation in order to account for that: model (6) is expressed conditionally not only on the state variable, but also on $\chi_{it}$ - i.e. productivity is a function of its past values and of the survival indicator variable:

$$y_{it} - \mathbf{w}_{it}\hat{\beta} = \alpha_0 + \mathbf{x}_{it}\gamma + E[\omega_{it}|\,\omega_{it-1},\,\chi_{it}] + e_{it} \tag{10}$$

The bias correction proposed by OP consists in adding to (7) an estimate of the unconditional probability of remaining active in the market, i.e. $\hat{Pr}_{it} \equiv Pr\{\chi_{it+1} = 1|\mathbf{x}_{it}\}$. Thus:

$$y_{it} - \mathbf{w}_{it}\hat{\beta} = \alpha_0 + \mathbf{x}_{it}\gamma + g(\hat{\Phi}_{it-1} - \mathbf{x}_{it-1}\gamma,\,\hat{Pr}_{it-1}) + e_{it} \tag{11}$$

[4]Alternatively, estimation of second stage can be carried out on Eq. (7) using non lineal least squares since $e_{it}$ is a combination of pure errors.

where $\hat{Pr}_{it-1}$ is the fitted surviving probability - typically estimated through a discrete choice model on a polynomial of the state variable vector $\mathbf{x}_{it}$ and the investment.

## 2.2. Levinsohn-Petrin method

OP approach has a major drawback in empirical applications which limits its range of applications: real firm- or plant-level data have many zeros in investment preventing, in practice, the estimation. This is due to common industrial practices which violate the monotonicity assumption A.1: investments are not decided at each point in time, but accumulated for few years before being made all at once. LP propose to overcome this issue by exploiting intermediate input levels as a proxy variable for $\omega_{it}$. As in the OP case, LP methodology is based on some assumptions:

B.1 Firms observe their productivity shock and adjust their optimal level of intermediate inputs - materials - according to the demand function $m(\omega_{it}, \mathbf{x}_{it})$;

B.2 $m_{it} = f(\mathbf{x}_{it}, \omega_{it})$ is the intermediate input function, invertible in $\omega_{it}$. Moreover, $m_{it}$ is monotonically increasing in $\omega_{it}$;

B.3 The state variables - typically capital - evolve according to the investment policy function $i()$ which is decided at time $t-1$;

B.4 The free variables $\mathbf{w}_{it}$ - typically labor inputs - are non-dynamic, in the sense that their choice at $t$ does not impact future profits, and are chosen in $t$ after the firm productivity shock realizes.

Under the set of assumptions B.1-B.4, intermediate input demand is orthogonal to the set of state variables in $t$ such that $E[m_{it}|\mathbf{x}_{it}] = 0$ and $m_{it}$ can be inverted, yielding the following technical efficiency proxy:

$$\omega_{it} = h(m_{it}, \mathbf{x}_{it}) \tag{12}$$

which is an unknown function of observable variables. Plugging (12) in (1) and distinguishing the intermediate input variable from the free variables we obtain:

$$\begin{aligned} y_{it} &= \alpha + \mathbf{w}_{it}\beta + \mathbf{x}_{it}\gamma + \delta m_{it} + h(m_{it}, \mathbf{x}_{it}) + e_{it} = \\ &= \mathbf{w}_{it}\beta + \Phi_{it}(m_{it}, \mathbf{x}_{it}) + e_{it} \end{aligned} \tag{13}$$

where $e_{it} = \xi_{it} + \varepsilon_{it}$.

Equation (2.2) is a partially linear model identified only in the free variable vector but not in the proxy variable, $m_{it}$. Similar to OP, equation (12) can be non-parametrically estimated approximating $\Phi_{it}(m_{it}, \mathbf{x}_{it})$ by a $n^{th}$ order polynomial or by local linear regression (First Stage). At the true values $[\gamma*, \delta^*]$ we can define the residual function $e_{it}$ like:

$$e_{it} = y_{it} - \mathbf{w}_{it}\hat{\beta} - \mathbf{x}_{it}\gamma^* - g\left(\hat{\Phi}_{it-1}\left(\delta*\right) - \mathbf{x}_{it-1}\gamma\right) \tag{14}$$

However, $e_{it}$ is no longer a combination of pure errors. The intermediate input variable is correlated with the error term given firms' response to the technology efficiency shock $\xi_{it}$. Thus, non-linear least squares would provide inconsistent estimates and relying on a GMM estimator is mandatory. The GMM estimator might be constructed by exploiting the residuals $e_{it}$ and the set of moment conditions $E[e_{it}z_{it}^k]=0$, $\forall k$, where $k$ is the index of the instrument vector $\mathbf{z} = [\mathbf{x}_{it}, \mathbf{m}_{it-1}]$

$$[\gamma*, \delta^*] = \text{argmax} \left\{ \sum_k \left( \sum_i \sum_t e_{it} z_{it}^k \right)^2 \right\} \tag{15}$$

consistently estimates the set of paramenters $[\gamma, \delta]^\intercal$.

### 2.3. Ackerberg, Caves and Frazer correction

Both OP and LP assume that firms are able to instantly adjust some inputs at no cost when subject to productivity shocks. However, ACF and Bond and Soderbom (2005) remark that the labor coefficient can be consistently estimated in the first stage only if the free variables show variability independently from the proxy variable. If this is not the case, their coefficients would be perfectly collinear in the first-stage estimation and hence would not be identifiable. In particular, in the LP setting labor and intermediate inputs are assumed to be allocated simultaneously at $t$. This implies that labor and materials are both chosen as a function of productivity and state variables $\mathbf{x}_{it}$:

$$\begin{aligned} m_{it} &= m(\omega_{it}, \mathbf{x}_{it}) \\ l_{it} &= l\,(\omega_{it}, \mathbf{x}_{it}) \end{aligned} \tag{16}$$

Using the monotonicity condition (B.2) ACF provide the following results:

$$l_{it} = l\,[h(m_{it}, \mathbf{x}_{it}), \mathbf{x}_{it}] \tag{17}$$

Hence, a collinearity issue arises in estimating the first stage, where the labor appears both as a free variable and in the non-parametric polynomial approximation $\hat{\Phi}_{it}$. In the same fashion the collinearity issue affects the OP estimator. ACF propose an alternative approach based on the following assumptions:

C.1 $p_{it} = p_{it}(\mathbf{x}_{it}, l_{it}, \omega_{it})$ is the proxy variable policy function, invertible in $\omega_{it}$. Moreover, $p_{it}$ is monotonically increasing in $\omega_{it}$;

C.2 The state variables are decided at time $t - b$;

C.3 The labor input, $l_{it}$, is chosen at time $t - \zeta$, where $0 < \zeta < 1$. The free variables, $\mathbf{w}_{it}$, are chosen at time $t$ when the firm productivity shock is realized.

Under the set of assumptions C.1-C.3 the first stage estimation is meant to remove the shock $\varepsilon_{it}$ from the the output $y_{it}$. In particular the policy function can be inverted and plugged in equation (1) yielding:

$$y_{it} = \Phi_{it}(p_{it}, \mathbf{x}_{it}, \mathbf{w}_{it}, l_{it}) + \varepsilon_{it} \tag{18}$$

where $\Phi_{it}(p_{it}, \mathbf{x}_{it}, \mathbf{w}_{it}, l_{it}) = \mathbf{x}_{it}\gamma + \mathbf{w}_{it}\beta + \mu l_{it} + h(p_{it}, \mathbf{x}_{it}, \mathbf{w}_{it}, l_{it})$. Once $\hat{\Phi}_{it}$ is recovered, for any candidate vector $(\gamma^*, \beta^*, \mu^*)$, it is possible to obtain the residuals

$$\hat{\omega}_{it} = \hat{\Phi}_{it} - \mathbf{x}_{it}\gamma^* - \mathbf{w}_{it}\beta^* - \mu^* l_{it} \tag{19}$$

and, exploiting the Markov chain assumption $\omega_{it} = E(\omega_{it}\,|\omega_{it-1}) + \xi_{it} = g(\omega_{it-1}) + \xi_{it}$, obtain the residuals $\xi_{it}$. These, combined with the set of moment conditions $E[\xi_{it} z_{it}^k]$=0, $\forall k$, where $k$ is the index of the instrument vector $\mathbf{z} = [\mathbf{x}_{it}, \mathbf{m}_{it-1}, l_{it-1}]$, lead to the GMM criterion function (Second stage):

$$[\gamma^*, \beta^*, \mu^*] = \text{argmax} \left\{ \sum_k \left( \sum_i \sum_t \xi_{it} z_{it}^k \right)^2 \right\} \tag{20}$$

## 2.4. Wooldridge

Wooldridge (2009) proposes to address the OP/LP problems by replacing the two-step estimation procedure with a generalized method of moments (GMM) setup as in Wooldridge (1996). In particular, he shows how to write the relevant moment restrictions in terms of two equations: these have the same dependent variable ($y_{it}$) but are characterized by a different set of instruments. Such approach has useful features with respect to previously proposed estimation routines:

- it overcomes the potential identification issue highlighted by ACF in the first stage;
- robust standard errors are easily obtained, accounting for both serial correlation and/or heteroskedasticity[5].

In Equation (2.2) $\Phi(x_{it}, m_{it}) \equiv \alpha + x_{it}\gamma + h(x_{it}, m_{it})$. The estimation of $(\beta, \gamma)$ is addressed in the first stage by OP/LP under the assumption that

$$E(\varepsilon_{it}|\omega_{it-1}, w_{it}, x_{it}, m_{it}, w_{it-1}, x_{it-1}, m_{it-1}, ..., w_{i1}, x_{i1}, m_{i1}) = 0 \tag{21}$$

without imposing any functional form on $h(.,.)$. The second stage assumption exploits the markovian nature of productivity and the assumed uncorrelation between productivity shocks and current values of state variables and past realizations of free variables and intermediate inputs. Following LP and rewriting Eq. (2) it states:

$$E(\omega_{it}|x_{it}, w_{it-1}, x_{it-1}, m_{it-1}, ..., w_{i1}, x_{i1}, m_{i1}) = E(\omega_{it}\,|\omega_{it-1}) = f[h(x_{it-1}, m_{it-1})] \tag{22}$$

where, as for $h(.,.)$, no functional form is imposed for $f(.)$. Assumptions (21) and (22) directly lead to the formulation of the two key functions to identify $(\beta, \gamma)$:

$$y_{it} = \alpha + \mathbf{w}_{it}\beta + \mathbf{x}_{it}\gamma + h(\mathbf{x}_{it}, \mathbf{m}_{it}) + v_{it} \tag{23}$$

[5]LP and OP recommend instead to bootstrap the standard errors of their estimators, as usual in two-step estimation procedures.

$$y_{it} = \alpha + \mathbf{w}_{it}\beta + \mathbf{x}_{it}\gamma + f[h(\mathbf{x}_{it-1}, \mathbf{m}_{it-1})] + \eta_{it} \tag{24}$$

where $\eta_{it} = \xi_{it} + v_{it}$.

In the estimation the approach followed is to deal with the unknown functional forms using $\mathrm{n}^{th}$ order polynomials in $\mathbf{x}_{it}$ and $\mathbf{m}_{it}$[6], where the limiting case with $x_{it}$ and $m_{it}$ (i.e. $n = 1$) entering linearly should always be allowed. In particular, if we assume that

$$h(\mathbf{x}_{it},\, \mathbf{m}_{it}) = \lambda_0 + k(\mathbf{x}_{it}, \mathbf{m}_{it})\lambda_1 \tag{25}$$

it implies $f(\omega_{it}) = \delta_0 + \delta_1[k(x_{it}, m_{it})\lambda_1] + \delta_2[k(x_{it}, m_{it})\lambda_1]^2 + ... + \delta_G[k(x_{it}, m_{it})\lambda_1]^G$.

For sake of simplicity, consider the case with $G = 1$ and $\delta_1 = 1$[7]: a simple substitution in Eqs. (23)-(24) yields

$$y_{it} = \zeta + \mathbf{w}_{it}\beta + \mathbf{x}_{it}\gamma + k(\mathbf{x}_{it}, \mathbf{m}_{it})\lambda_1 + v_{it} \tag{26}$$

$$y_{it} = \theta + \mathbf{w}_{it}\beta + \mathbf{x}_{it}\gamma + k(\mathbf{x}_{it-1}, \mathbf{m}_{it-1})\lambda_1 + \eta_{it} \tag{27}$$

The choice of instruments for both Eq. (26) and (27) is straightforward and reflects the orthogonality conditions listed above: in particular, we define $\mathbf{z}_{it1} = (1, \mathbf{x}_{it}, \mathbf{w}_{it}, k(\mathbf{x}_{it}, \mathbf{m}_{it}))$, $\mathbf{z}_{it2} = (1, \mathbf{x}_{it}, \mathbf{w}_{it-1}, k(\mathbf{x}_{it-1}, \mathbf{m}_{it-1}))$ and $\mathbf{Z}_{it} = \begin{pmatrix} \mathbf{z}_{it1} \\ \mathbf{z}_{it2} \end{pmatrix}$.

For each $t > 1$ the usual GMM with IV setup applies and the moment conditions are derived from the residual functions

$$\mathbf{r}_{it}(\theta) = \begin{pmatrix} r_{it1}(\theta) \\ r_{it2}(\theta) \end{pmatrix} = \begin{pmatrix} y_{it} - \zeta - \mathbf{w}_{it}\beta - \mathbf{x}_{it}\gamma - k(\mathbf{x}_{it}, \mathbf{m}_{it})\lambda_1 \\ y_{it} - \theta - \mathbf{w}_{it}\beta - \mathbf{x}_{it}\gamma - k(\mathbf{x}_{it-1}, \mathbf{m}_{it-1})\lambda_1 \end{pmatrix} \tag{28}$$

and $E[\mathbf{Z}'_{it}\mathbf{r}_{it}(\theta)] = 0$.

[6] LP suggest to use third-degree polynomials. However, the higher the degree the better the result.

[7] This is the case whose estimation is implemented in `prodest`.

# 3. The R package prodest

The R package **prodest** offers an integrated environment to deal with production function approach methods in R. The implementation is straightforward and follows from the steps and the models outlined in the previous section. In particular, it includes functions for i) estimation, and ii) simulation of production function data along with a subset of a widely used dataset on Chilean production.

The general nomenclature of estimation functions is "`prodest...()`" , where methods are *OP*, *LP*, *ACF* and *WRDG*. These stand for (Olley and Pakes 1996; Levinsohn and Petrin 2003; Ackerberg et al. 2015) and Wooldridge (2009), respectively. As remarked in the previous sections, all of these - apart from Wooldridge, a one-step system of equations - are two-step models.

The function `panelSim()` is a Data Generating Process (DGP) that creates data suitable for Monte Carlo simulations. It is directly taken from the DGP presented by Ackerberg et al. (2015)

## 3.1. Specification

All **prodest** estimation functions accept at least 6 objects - either vectors, dataframes or matrices:

- `Y`: vector/matrix/dataframe of log value added output. Every element of it corresponds to the $y_{it}$ in (4), (2.2), (18) and (26);
- `fX`: the vector/matrix/dataframe of log free variables. These often account for the labor input at each point in time - the $\mathbf{w}_{it}$ in all above models;
- `sX`: the vector/matrix/dataframe of log state variables. These often account for the capital input at each point in time - the $\mathbf{x}_{it}$ in all specifications;
- `pX`: the vector/matrix/dataframe of log proxy variables. In OP these account for the amount of investment (corresponding to $i_{it}$ in 4), whereas, following the ACF's and WRDG's critique to the LP approach, the material input are generally a better choice for the polynomial. The suitability of the proxy variable in these models strongly depends on the type of firm, sector and country of the analysis;
- `idvar`: the vector/matrix/dataframe identifying individual panels. More specifically, it must contain a unique identifier for the plant / firm, either in string or (preferred) numeric format.
- `timevar`: the vector/matrix/dataframe identifying time: it should contain the date (year, quarter, etc.) in numeric format.

There is a number of optional inputs, mostly model-specific. `cX` is a vector/matrix/dataframe of control variables to be used in the estimation, by default `cX = NULL`; `seed` accepts a numeric element and sets the seed for the estimation - i.e., it ensures results' replicability - and by default `seed = 123456`. The two-step models, then, accept further parameters related to bootstrap repetitions, second stage optimization and parallelization.

In particular, `opt` accepts a string element taking values *'optim'*, *'DEoptim'* or *'solnp'*. These refer to optimization procedures used for the second stage in the GMM framework: `optim` (package **stats**) performs general-purpose optimization based on Nelder-Mead algorithm; `DEoptim` - package **DEoptim** by (Mullen et al. 2011; Ardia et al. 2011) - performs the optimization by the Differential Evolution algorithmPrice *et al.* (2006); `solnp` - package **Rsolnp** by Ghalanos and Theussl (2015) - implements the solver firstly proposed by Ye (1988).

`cluster` option - `cluster = NULL` by default - accepts objects of class `SOCKcluster` or `cluster` (e.g., using `makeCluster()` in **parallel** by Tierney and R Core team). If specified, the bootstrap repetitions for the second stage standard errors are performed in parallel. `theta0` is a numeric vector/matrix that can be specified in order to provide starting points in the optimization of the second stage. Its use is recommended for methods like ACF, which are very sensitive and whose value function appears to show several local maxima - see section 4 for a general discussion of the issue. By default `theta0 = NULL` and, in this case, the starting points are the first-stage results plus a noise drawn from a standard normal ($\mu_0 = 0, \sigma^2 = 0.01$).

### 3.2. Estimation

`prodestOP()` and `prodestLP()` proceed with a straightforward OLS estimation of Eq. (4) and (2.2) with a $2^{nd}$ order polynomial in state and proxy variables. The first stage regression yields consistent estimates of $\beta$. Then, using the fitted residuals and the moment conditions in (9) or (15) they proceed with the GMM estimation of the second stage, which yields $\hat{\gamma}$. Estimates of standard errors are computed by *cluster-Bootstrapping* at individual level the second stage and collecting the estimates.

`prodestACF()` builds on a similar framework, but the first stage regression in (18) does not yield $\hat{\beta}$, which is instead estimated jointly with $\hat{\gamma}$ in the second stage. It is done exploiting the moment conditions in (20) in a GMM setting. Standard errors are computed by *cluster-Bootstrapping* at individual level the second stage and collecting the estimates.

The Wooldridge estimator is implemented in two versions by **prodest**, with functions `prodestWRDG()` and `prodestWRDG_GMM()`. The first is a classical two-step system GMM framework: it proceeds by estimating the system with an initial weighting matrix and assuming that the moment equations are independent and identically distributed, with covariances between two moment equations set to zero - first step. It uses the `optim` algorithm to perform the estimation. Then, with the estimates of the first step it is possible to compute the optimal weighting matrix $W^{\star} = \sigma_{rs} Z'Z$ (see Wooldridge (2002)) and estimate the second-step parameters with the same econometric model. Standard errors are computed by *cluster-Bootstrapping* at the individual level and collecting the estimates.
`prodestWRDG_GMM()` implements a slightly different version of the same two-step estimator, building a matrix of regressors $X$ for the system of equation aimed at avoiding collinearity

issues in the following way:

$$\mathbf{X} = \begin{bmatrix} X1_1^1 & \cdots & X1_1^k & X1_1^{k+1} & \cdots & X1_1^{r1} & 0 & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & 0 & 0 & 0 \\ X1_N^1 & \cdots & X1_N^k & X1_N^{k+1} & \cdots & X1_N^{r1} & 0 & 0 & 0 \\ X2_1^1 & \cdots & X2_1^k & 0 & 0 & 0 & X2_1^{k+1} & \cdots & X2_1^{r2} \\ \vdots & \ddots & \vdots & 0 & 0 & 0 & \vdots & \ddots & \vdots \\ X2_N^1 & \cdots & X2_N^k & 0 & 0 & 0 & X2_N^{k+1} & \cdots & X2_N^{r2} \end{bmatrix} \tag{29}$$

where $k$ is the number of common regressors - the constant $\alpha$, the free and state variables in equations (23) and (23) - $r1$ the number of regressors in the first equation and $r2$ the regressors in the second. Then, the estimation is performed in two steps relying on the fact that the solution to the GMM for over-identified linear models reads $\hat{\beta}^0 = ((X'Z)W^0(Z'X))^{-1}(X'Z)W^0(Z'Y)$ - first step with unadjusted and independent weighting matrix $W^0$. Using $\hat{\beta}^0$, define the optimal weighting matrix $W^\star = \sigma_{rs}Z'Z$ and the parameters are estimated in the second step like $\hat{\beta}^\star = ((X'Z)W^\star(Z'X))^{-1}(X'Z)W^\star(Z'Y)$, with variance-covariance matrix $Var(\hat{\beta}^\star) = \frac{1}{N}((X'Z)W^\star(Z'X))^{-1}$.[8]

All **prodest** estimation functions return objects of the `S3` class `prod`. These are lists of length 3 with elements `Model`, `Data` and `Estimates`. In turn, these feature several subelements:

- `Model`: `$method`, a string with the method - OP, LP, etc. - `$boot.repetitions`, the number of boostrap repetitions used to compute the standard errors, `$elapsed.time`, showing the estimation time, `$theta0`, the vector of optimization starting points, `$opt`, a string with the optimizer, `$seed`, `$opt.outcome`, the output from the optimizer, `$FSbetas`, the vector of first stage estimated parameters.
- `Data`: the output, free, state, proxy and control variables used in the estimation are stored in `$Y`, `$free`, `$state`, `$proxy` and `$control`, respectively. `$idvar` and `$timevar` store the panel and time variable, while the `$FSresiduals` vector contains the vector of first-stage residuals.
- `Estimates`: `$pars`, the vector of estimated parameters, and `$std.errors`, the vector of parameters' standard errors.

### 3.3. Simulation

`panelSim()` is a function implementing the data generating process proposed by Ackerberg et al. (2015). They propose three different DGPs, clustered according to their characteristics: the DGP1 is simulated with serially correlated wages and the labor input is set at time $t - b$, the DGP2 with an optimization error in labor and the DGP3 is simulated with both sources of heterogeneity.

`panelSim()` accepts up to 9 optional inputs, namely `N`, which is the number of firms, `T`, the time spanned by the simulation, $\alpha_l$ and $\alpha_k$, the productivity parameters of the free and the state variables, respectively, `DGP`, which controls for the type of DGP, $\rho$ is the AR(1) coefficient

[8] see (Wooldridge 2001, 2002)

of $\omega$ in equation (2), $\sigma_\epsilon$ and $\sigma_\omega$, the standard deviations of $\epsilon$ and of the productivity shocks, and $\rho_{lnw}$, which is the AR(1) coefficient of log(wages). Option `seed` sets the seed as the simulation starts.

`panelSim()` returns a `data.frame` with the `idvar`, the `timevar`, the output `Y`, the free - `fX` - and state - `sX` - variables. In addition, it produces 4 proxy variables (`pX1-pX4`), simulated with a measurement error drawn from distributions with standard deviation of 0, 0.1, 0.2 and 0.5, respectively. `timevar` will amount to the 10% of `T`, in order to deliver only observations simulated in equilibrium.

### 3.4. Methods

**prodest** features several post-estimation methods aimed at handling `prod` objects. Among them, `show()`, `coef()`, `FSres()` and `summary`, aimed at extracting parts of the stored results; see `help("prod")`. `omega()` generates the residuals of the second stage, namely, the estimates of the log productivity term.

## 4. Comparative results

### 4.1. Real Data

In table (1) we report the results of all models implemented in **prodest** on a sectoral subset of the well-known and broadly used dataset of Chilean firms 1986-1996.

Table 1: Chilean dataset - sectoral estimation

| | OP | LP | Acf | Wrdg | Wrdg.GMM |
|---|---|---|---|---|---|
| $\beta_{skil}$ | 0.314 | 0.199 | 0.646 | 0.247 | 0.247 |
| | (0.034) | (0.025) | (0.159) | (0.031) | (0.017) |
| $\beta_{unskil}$ | 0.256 | 0.169 | 0.644 | 0.217 | 0.217 |
| | (0.036) | (0.022) | (0.209) | (0.03) | (0.015) |
| $\beta_k$ | 0.168 | 0.117 | 0.251 | 0.058 | 0.058 |
| | (0.028) | (0.041) | (0.035) | (0.092) | (0.031) |
| N | 2544 | 2544 | 2544 | 2544 | 1944 |

Note: productivity estimation on a sector-specific sample of Chilean firms 1986-1996. Free variables are *skilled* and *unskilled* quantities of labor input, the state variable is capital $k$ and the proxy variables is investment - OP - or material inputs - LP, ACF, WRDG. Column titles indicate the estimated models. Standard errors are bootstrapped - for all but *Wrdg.GMM* models and reported in parenthesis.

As an illustration of how **prodest** works, the code producing table (1) reads

```
R> data(chilean)
R> OP <- prodestOP(Y, fX, sX, pX = d\$log_investment, idvar, timevar, R = 20)
```

```
R> LP <- prodestLP(Y, fX, sX, pX, idvar, timevar, R = 20)
R> ACF <- prodestACF(Y, fX, sX, pX, idvar, timevar, R = 20, theta0 = (c(.5,.5,.5)))
R> WRDG <- prodestWRDG(Y, fX, sX, pX, idvar, timevar, R = 5)
R> WRDG.GMM <- prodestWRDG_GMM(Y, fX, sX, pX, idvar, timevar, R = 5)
R> printProd(list(OP,LP,ACF,WRDG,WRDG.GMM), modnames = c('OP','LP','Acf','Wrdg','Wrdg.GMM'
+ parnames = c('$\\beta_{skil}$','$\\beta_{unskil}$','$\\beta_{k}$'))

\begin{tabular}{ccccccccccc}\hline\hline
 & & OP & & LP & & Acf & & Wrdg & & Wrdg.GMM \\\hline
 $\beta_{skil}$ & & 0.314 & & 0.199 & & 0.646 & & 0.247 & & 0.247 \\
 & & (0.034) & & (0.025) & & (0.159) & & (0.031) & & (0.017) \\
 &  &  &  &  &  \\
 $\beta_{unskil}$ & & 0.256 & & 0.169 & & 0.644 & & 0.217 & & 0.217 \\
 & & (0.036) & & (0.022) & & (0.209) & & (0.03) & & (0.015) \\
 &  &  &  &  &  \\
 $\beta_{k}$ & & 0.168 & & 0.117 & & 0.251 & & 0.058 & & 0.058 \\
 & & (0.028) & & (0.041) & & (0.035) & & (0.092) & & (0.031) \\
 &  &  &  &  &  \\
 &  &  &  &  &  \\
N & & 2544 & & 2544 & & 2544 & & 2544 & & 1944 \\\hline\hline
\end{tabular}
```

### 4.2. Monte Carlo simulations

Using `panelSim()` it is possible to run Monte Carlo-type estimations on a plethora of simulated datasets. In table (2) we report a replica of Ackerberg et al. (2015)'s Table I using `prodestACF()` - columns 1-4 - and `prodestLP()` - columns 5-8 - on 12 different simulated datasets (DGP1-3, each featuring 4 different levels of measurement error). Results replicate fairly well those presented in the original paper.

### 4.3. Parallel

Parallel computing is extremely useful in **prodest** applications, mostly when dealing with a high number of bootstrap repetitions. In table (3) we report the results, in terms of estimates, number of bootstrap repetitions and computing time, of the ACF model run on a simulated dataset (`DGP = 2`, `N = 1000`, `T = 100`) with 100 and 1000 bootstrap repetitions and estimated plainly - columns (1) and (4) - or parallelized over 2 - columns (2) and (5) - and 3 - columns (3) and (6) - cores. See *Computational Details* below for a full reference to the hardware used.

As expected, the computational time decreases with the number of cores employed, and it is particularly relevant when the number of bootstrap repetitions increases: with 1000 repetitions, using 3 cores halves the computational time ($\approx 6$ to $\approx 3$ mins).

Table 2: ACF and LP - Monte Carlo Simulations

| | ACF | | | | LP | | | |
|---|---|---|---|---|---|---|---|---|
| | $\beta_l$ | | $\beta_k$ | | $\beta_l$ | | $\beta_k$ | |
| Meas. Error | Coeff. | St. Dev. | Coeff. | St. Dev. | Coeff. | St. Dev. | Coeff. | St. Dev. |
| *DGP1 - Serially Correlated Wages and Labor Set at Time $t-b$* | | | | | | | | |
| 0.0 | 0.599 | 0.009 | 0.401 | 0.015 | -0.004 | 0.005 | 1.099 | 0.030 |
| 0.1 | 0.600 | 0.011 | 0.425 | 0.016 | 0.679 | 0.009 | 0.365 | 0.012 |
| 0.2 | 0.621 | 0.012 | 0.406 | 0.015 | 0.790 | 0.007 | 0.242 | 0.010 |
| 0.5 | 0.668 | 0.015 | 0.357 | 0.017 | 0.876 | 0.005 | 0.195 | 0.160 |
| *DGP2 - Optimization Error in Labor* | | | | | | | | |
| 0.0 | 0.606 | 0.054 | 0.395 | 0.054 | 0.600 | 0.003 | 0.401 | 0.013 |
| 0.1 | 0.610 | 0.030 | 0.406 | 0.033 | 0.755 | 0.004 | 0.257 | 0.009 |
| 0.2 | 0.617 | 0.021 | 0.404 | 0.024 | 0.809 | 0.004 | 0.205 | 0.010 |
| 0.5 | 0.633 | 0.019 | 0.389 | 0.022 | 0.864 | 0.003 | 0.446 | 0.238 |
| *DGP3 - Optimization Error in Labor and Serially Correlated Wages and Labor Set at Time $t-b$ (DGP1 plus DGP2)* | | | | | | | | |
| 0.0 | 0.593 | 0.005 | 0.409 | 0.014 | 0.473 | 0.003 | 0.576 | 0.017 |
| 0.1 | 0.602 | 0.037 | 0.425 | 0.041 | 0.635 | 0.005 | 0.417 | 0.012 |
| 0.2 | 0.610 | 0.042 | 0.424 | 0.041 | 0.702 | 0.005 | 0.348 | 0.012 |
| 0.5 | 0.621 | 0.023 | 0.415 | 0.027 | 0.778 | 0.005 | 1.320 | 0.191 |

*Notes:* 1000 replications. True values of parameters are $\beta_l = 0.6$ and $\beta_k = 0.4$. Standard deviations have been calculated among 1000 replications. $\rho$ is set at .7 and we used **optim** (optimizer: 'BFGS').
DGP1 - Serially Correlated Wages and Labor Set at Time $t-b$
DGP2 - Optimization Error in Labor
DGP3 - Optimization Error in Labor and Serially Correlated Wages and Labor Set at Time $t-b$ (DGP1 plus DGP2)

Table 3: ACF estimation in parallel - various cores

| | Acf | Acf-par2 | Acf-par3 | Acf | Acf-par2 | Acf-par3 |
|---|---|---|---|---|---|---|
| $\beta_{free}$ | 0.617 | 0.617 | 0.617 | 0.617 | 0.617 | 0.617 |
| | (0.007) | (0.007) | (0.007) | (0.009) | (0.009) | (0.009) |
| $\beta_{state}$ | 0.386 | 0.386 | 0.386 | 0.386 | 0.386 | 0.386 |
| | (0.014) | (0.014) | (0.014) | (0.016) | (0.016) | (0.016) |
| Time | 35.61 | 28.97 | 26.59 | 6.19 | 3.43 | 3.23 |
| BootRep | 100 | 100 | 100 | 1000 | 1000 | 1000 |
| N | 10000 | 10000 | 10000 | 10000 | 10000 | 10000 |

Notes: all models are estimated on a DGP1, N=1000, T=100 simulated dataset. Columns (1)-(3) show models estimated with 100 bootstrap repetitions for the standard errors, while (4)-(6) employ 1000 repetitions. Estimates are obtained using 1 (plain), 2, or 3 cores in parallel.

## 5. Conclusions

This article introduced the R package **prodest** for production function estimation using the control function approach and for simulating production data. It is the first package that implements the models proposed by Olley and Pakes (1996), Levinsohn and Petrin (2003), Ackerberg et al. (2015) and Wooldridge (2009); as such, it allows practitioners to perform their applied research on TFP within the R environment.

After providing a brief theoretical background of the models implemented, I introduced the specifications and illustrated the package usage and various features. In order to do that, I performed several estimations on real and simulated data, showing how the performance can improve with the use of parallel computing, even with a relatively small number of cores.

## Computational Details

All results in this paper were obtained using R 3.3.2 (R Core Team 2017) with the packages: **prodest** version 1.0.1 Rovigatti (2017), **Rsolnp** version 1.16 Ghalanos and Theussl (2015), **DEoptim** version 2.2.4 Ardia et al. (2011), **plyr** version 1.8.4 Wickham (2011), **parallel** version 3.3.2 R Core Team (2017), **Matrix** version 1.2.7.1 Bates and Maechler (2016). Computations were performed on a Intel® Core CPU i7–6500U 2.59Ghz processor.

R itself and all packages used are available from CRAN at http://CRAN.R-project.org/.

## Acknowledgments

Special thanks go to Francesco Decarolis who supervised the whole project, providing expertise and support. Also, I thank Daniel Ackerberg, Federico Belotti, Sara Calligaris, Leopoldo Catania, Vincenzo Mollisi, Andrea Pozzi, Lorenzo Rovigatti and Federico Tullio for comments that improved the package and the paper. Any errors are my own.

## References

Ackerberg, D. and Caves, K. and Frazer, G. (2015). "Identification Properties of Recent Production Function Estimators." *Econometrica*, **83**(6), 2411–2451.

Ardia, D. and Mullen, K. and Peterson, B. G. and Ulrich, J. (2016). "'DEoptim': Differential Evolution in 'R'." R package version 2.2-4, URL https://cran.r-project.org/package=DEoptim.

Bates, D. and Maechler M. (2016). ***Matrix**: Sparse and Dense Matrix Classes and Methods*. R package version 1.2-7.1, URL https://CRAN.R-project.org/package=Matrix.

Blundell, R. and Bond, S. (1998). "Initial conditions and moment restrictions in dynamic panel data models." *Journal of Econometrics*, **87**, 115–143.

Bond, S. and Soderbom, M. (2005). "Adjustment Costs and the Identification of Cobb Douglas Production Functions." *Working Paper*.

Ghalanos A. and Theussl S. (2015). ***Rsolnp**: General Non–Linear Optimization using Augmented Lagrange Multiplier Method*. R package version 1.16, URL https://cran.r-project.org/package=Rsolnp.

Hopenayn, H. A. (1992). "Entry, Exit and Firm Dynamics in Long Run Equilibrium." *Econometrica*, **60**(5), 1127–1150.

Levinsohn, J. and Petrin, A. (2003). "Estimating Production Functions Using Inputs to Control for Unobservables." *The Review of Economic Studies*, **70**(2), 317–341.

Melitz, M. J. (2003). "The Impact of Trade on Intra-Industry Reallocations and Aggregate Industry Productivity." *Econometrica*, **71**(6), 1695–1725.

Mollisi, V. and Rovigatti, G. (2017). "Theory and Practice of TFP Estimation: the Control Function Approach Using Stata." submitted to *The Stata Journal*.

Mullen, K. and Ardia, D. and Gil, D. and Windower, D. and Cline, J. "DEoptim: An R Package for Global Optimization by Differential Evolution." *Journal of Statistical Software*, **40**(6).

Olley, S. G. and Pakes, A. (1996). "The Dynamics of Productivity in the Telecommunications Equipment Industry." *Econometrica*, **64**(6), 1263–1297.

Price, K. V. and Storn, R. M. and Lampinen, J. A. (2006). "Differential Evolution - A Practical Approach to Global Optimization." *Berlin Heidelberg: Springer-Verlag*, ISBN 3540209506.

Rovigatti, G. (2017). "Production Function Estimation in R: The **prodest** package." R package version 0.1.1

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/.

Ye, Y. (1988). "Interior algorithms for linear, quadratic, and linearly constrained non linear programming." *Stanford University, PhD Thesis*.

Wickham, H. (2011). "The Split-Apply-Combine Strategy for Data Analysis." *Journal of Statistical Software*, **40**(1), 1–29. URL http://www.jstatsoft.org/v40/i01/.

Wooldridge, J. M. (1996). "Estimating systems of equations with different instruments for different equations." *Journal of Econometrics*, **74**, 387–405.

Wooldridge, J. M. (2001). "Applications of generalized method of moments estimation." *The Journal of Economic Perspectives*, **15**(4), 87–100.

Wooldridge, J. M. (2002). "Econometric analysis of cross section and panel data." *MIT press*.

Wooldridge, J. M. (2009). "On estimating firm-level production functions using proxy variables to control for unobservables." *Economics Letters*, **104**, 112–114.

**Affiliation:**

Gabriele Rovigatti
Booth School of Business
University of Chicago
5807 S Woodlawn Ave
Chicago, IL 60637, USA
E-mail: gabriele.rovigatti@gmail.com
URL: https://sites.google.com/view/gabrielerovigatti/home