DEPARTMENT OF FOOD AND RESOURCE ECONOMICS
UNIVERSITY OF COPENHAGEN

IFRO Working Paper

Panel Data Specifications in Nonparametric Kernel Regression: An Application to Production Functions

Tomasz Czekaj
Arne Henningsen

2013 / 5

IFRO Working Paper 2013 / 5

Panel Data Specifications in Nonparametric Kernel Regression:
An Application to Production Functions

Authors: Tomasz Czekaj, Arne Henningsen

Department of Food and Resource Economics (IFRO)
University of Copenhagen
Rolighedsvej 25
DK 1958 Frederiksberg DENMARK
www.ifro.ku.dk

Panel Data Specifications in Nonparametric Kernel Regression:

An Application to Production Functions

TOMASZ CZEKAJ and ARNE HENNINGSEN

Department of Food and Resource Economics, University of Copenhagen,
Rolighedsvej 25, 1958 Frederiksberg C, Denmark
e-mail: tcz@ifro.ku.dk, arne@ifro.ku.dk

March 2013

Abstract

We discuss nonparametric regression models for panel data. A fully nonparametric panel data specification that uses the time variable and the individual identifier as additional (categorical) explanatory variables is considered to be the most suitable. We use this estimator and conventional parametric panel data estimators to analyse the production technology of Polish crop farms. The results of our nonparametric kernel regressions generally differ from the estimates of the parametric models but they only slightly depend on the choice of the kernel functions. Based on economic reasoning, we found the estimates of the fully nonparametric panel data model to be more reliable.

Keywords: nonparametric kernel regression, panel data, choice of the kernel, kernels for categorical variables, production function
JEL codes: C14, C23, D24, Q12

1. Introduction

When analysing economic phenomena within a regression framework, economic theory very rarely provides distinct information regarding the functional form of a relationship between a dependent variable and its covariates. Simple specifications of regression functions—such as models that are linear in parameters—are most widely applied in empirical applications, because this simplifies both the econometric estimation as well as the economic interpretation of the estimated regression parameters. Nevertheless, the *a priori* assumption regarding the functional form of the regression function involves the risk of parametric misspecification, which could result in incorrect economic conclusions and recommendations. Nonetheless, in practice it is common that no formal test is conducted to detect a possible misspecification of the functional form (e.g. Ramsey's (1969) Regression Specification Error Test (RESET) or Utts' (1982) Rainbow test).

In recent years the rapidly growing literature on nonparametric econometric methods has offered a remedy for the problems related to the parametric misspecification of econometric regression models. Nonparametric regression techniques do not obligate the researcher to assume and specify a functional form for the relationship between the explanatory variables and the dependent variable. Thus, the functional form is determined by the data rather than by the researcher's arbitrary decision.

Nonparametric regression methods are most often applied to cross-sectional data, while they are seldom applied to panel data sets. However, the popularity of semiparametric and nonparametric regression methods for panel data has recently increased (e.g. Porter, 1996; Lin and Carroll, 2000; Wang, 2003; Henderson and Ullah, 2005; Su and Ullah, 2007; Henderson, Carroll and Li, 2008),[1] but which approach is the most suitable to account for the panel structure in nonparametric panel data models is still an open question. In order to answer this question, we discuss different panel data specifications which may be applied within nonparametric regression. We apply fully nonparametric specifications by using the time variable and the individual identifier as additional (categorical) explanatory variables in the nonparametric kernel regression framework for mixed data types that was proposed by Racine and Li (2004). Next we confront the result of the nonparametric regression models that utilised the fully nonparametric panel data specification with the results of parametric regression that use the conventional, therefore fully parametric panel data specification.

Furthermore, we investigate the use of different kernel functions in nonparametric kernel regression, where we particularly focus on the kernels for the categorical explanatory

[1] The most recent literature review on nonparametric and semiparametric panel data econometric methods can be found in Su and Ullah (2010). Although semiparametric models rely on less assumptions regarding the functional form than fully parametric models, they are still susceptible to the same misspecification problems as their parametric counterparts. Therefore, we will focus on fully nonparametric methods.

variables that are used to account for the panel structure, i.e. individual heterogeneity and time heterogeneity. Previous studies have found that the choice of the kernel is not as important as the choice of the bandwidths—for both continuous regressors (e.g. Silverman, 1986; Taylor, 1989) and categorical regressors (Racine and Li, 2004). We highlight the fact that bandwidths are not always directly comparable between different kernel types and we show how bandwidths for categorical variables can be made comparable. This allows us to compare the permissible individual heterogeneity and time heterogeneity across fully nonparametric panel data estimations with different kernels for the (categorical) individual and time variables.

As an empirical example, we estimate a firm-level production function of Polish crop farms based on a balanced panel data set of 342 farms in the years 2004–2010, which gives 2,394 observations in total. The concept of a production function is frequently used to investigate, e.g., the optimal firm size, the substitutability between production inputs, or the productivity and efficiency of individual firms. These issues have significant policy implications and therefore it is crucial to obtain consistent estimates. The nonparametric estimation of the production function with a fully nonparametric panel data specification avoids incorrect conclusions due to a misspecified functional form or a misspecified parametric panel data specification. In our specific empirical example, we focus on the returns to scale and the optimal firm size in order to contribute to the on-going policy debate about the structural change in the Polish farm sector and the numerous policy interventions that affect this restructuring.

The paper is organized as follows. Section 2 discusses different specifications of panel data regression models. Section 3 describes our empirical application, section 4 delivers and discusses the results of the conducted analyses, and section 5 concludes.

2. Specifications of panel data models

The increasing availability of panel data sets has resulted in rapid theoretical development of panel data regression methods during recent decades. The advantages of panel data over conventional cross-sectional and time-series datasets are unquestionable (see Hsiao, 2003, for detailed discussion on the advantages and challenges of panel data analysis). A vast literature on panel data analysis exists and a profound description of panel data methods can be found in many econometrics textbooks (e.g. Wooldridge, 2002; Arellano, 2003; Hsiao, 2003; Baltagi, 2005). Up to now, methodological contributions to panel data analysis, as well as empirical applications with panel data, have focused on parametric regression approaches.

A panel data regression model can be generally specified as:

$$y_{it} = f(x_{it}, i, t) + \epsilon_{it}, \tag{1}$$

where y_{it} is the observed dependent variable, $f(.)$ is the unknown regression function, x_{it} is a set of explanatory variables, ϵ_{it} is an idiosyncratic error term with $E[\epsilon_{it}|x_{it}, i, t] = 0$, whereas $i = 1, \ldots, N$ indicates the individual, and $t = 1, \ldots, T$ indicates the time. Regression models for panel data usually differ in the specification (simplification) of the unknown regression function $f(x_{it}, i, t)$.

2.1. Fully parametric panel data specification

In parametric panel data models, it is often assumed that the individual effects are additive and separable:[2]

$$f(x_{it}, i, t) = f^*(x_{it}, t) + \mu_i, \tag{2}$$

where μ_i is an individual-specific effect and $f^*(.)$ is the remaining part of the unknown regression function that is assumed to be the same for all individuals i. These panel data models are most often estimated as (one-way) "fixed effects" models, where the individual effects are usually eliminated either by the so-called "within transformation" or by first-differencing:

$$\tilde{y}_{it} \equiv y_{it} - \bar{y}_i = f^*(x_{it}, t) - \frac{1}{N}\sum_i f^*(x_{it}, t) + \tilde{\epsilon}_{it} \tag{3}$$

$$\Delta y_{it} \equiv y_{it} - y_{i,t-1} = f^*(x_{it}, t) - f^*(x_{i,t-1}, t-1) + \Delta\epsilon_{it}, \tag{4}$$

where $\bar{y}_i = (1/T)\sum_t y_{it}$, $\tilde{\epsilon}_{it} = \epsilon_{it} - \bar{\epsilon}_i$ with $\bar{\epsilon}_i = (1/T)\sum_t \epsilon_{it}$, and Δ is the first-difference operator.

In parametric panel data models, it is often assumed that $f^*(.)$ is linear in x_{it} and t[3] so that the regression models (3) and (4) simplify to:

$$\tilde{y}_{it} = f^*(\tilde{x}_{it}, t) + \tilde{\epsilon}_{it} \tag{5}$$

$$\Delta y_{it} = f^*(\Delta x_{it}, \Delta t) + \Delta\epsilon_{it}, \tag{6}$$

respectively, where $\tilde{x}_{it} = x_{it} - \bar{x}_i$ with $\bar{x}_i = (1/T)\sum_t x_{it}$. These models can be estimated by standard linear regression methods such as ordinary least squares (OLS).

[2] In this context, separability means that the individual effects are separable from (the effects of) the explanatory variables x_{it} and from (the effects of) time t and vice versa, i.e. $\partial f(.)/\partial x_{it}$ and $\partial f(.)/\partial t$ do not depend on i and $\partial f(.)/\partial i$ does not depend on x_{it} or t.

[3] This does not necessarily rule out nonlinear relationships between the dependent variable and the explanatory variables and/or time. For instance, variable y_{it} and vector x_{it} may be nonlinear transformations (e.g. logarithms) of the original dependent variable and the original explanatory variables, respectively. Furthermore, vector x_{it} may include more than one transformation of each original explanatory variable (e.g. linear and quadratic), one or more transformations of the time variable t, and interaction terms among and between the (original and transformed) explanatory variables and (original or transformed) time variables.

Sometimes, it is assumed that, as well as the individual effects, the time effects are also additive and separable:[4]

$$f(x_{it}, i, t) = f^{**}(x_{it}) + \mu_i + \nu_t, \tag{7}$$

where ν_t is a time-specific effect and $f^{**}(.)$ is the remaining part of the unknown regression function that is assumed to be the same for all individuals i and over all time periods t. In this case, the "within transformation" can be applied twice to eliminate both the individual effects and the time effects:

$$\tilde{\tilde{y}}_{it} \equiv y_{it} - \bar{y}_i - \bar{y}_t = f^{**}(x_{it}) - \frac{1}{N}\sum_i f^{**}(x_{it}) - \frac{1}{T}\sum_t f^{**}(x_{it}) - \bar{\mu} - \bar{\nu} + \tilde{\tilde{\epsilon}}_{it}, \tag{8}$$

where $\tilde{\tilde{\epsilon}}_{it} = \epsilon_{it} - \bar{\epsilon}_i - \bar{\epsilon}_t$ with $\bar{\epsilon}_t = (1/N)\sum_i \epsilon_{it}$.

If $f^{**}(.)$ is linear in x_{it},[5] regression model (8) simplifies to:

$$\tilde{\tilde{y}}_{it} = f^{**}(\tilde{\tilde{x}}_{it}) - \bar{\mu} - \bar{\nu} + \tilde{\tilde{\epsilon}}_{it}, \tag{9}$$

where $\tilde{\tilde{x}}_{it} = x_{it} - \bar{x}_i - \bar{x}_t$ with $\bar{x}_t = (1/N)\sum_i x_{it}$. This model can also be estimated by standard linear regression methods such as ordinary least squares (OLS), where the average individual and time effects, i.e. $\bar{\mu}$ and $\bar{\nu}$, are absorbed by the intercept of the regression function $f^{**}(.)$.

2.2. Nonparametric models with parametric panel data specifications

Since economic theory is very rarely informative regarding the functional form, the *a priori* selection of a functional form of $f(.)$, $f^*(.)$, or $f^{**}(.)$ involves the risk of specifying a functional form that is not similar to the "true" functional relationship between x_{it} and y_{it}. This misspecification can lead to inconsistent parameter estimates, and hence to incorrect inference regarding the investigated phenomena. A nonparametric estimation of the regression function avoids the risk of a misspecified functional form. However, the model specifications in (5), (6), and (9) are based on the linearity of $f^*(.)$ and $f^{**}(.)$ so that they are invalid in a nonparametric framework.

Li and Stengos (1996) introduced an estimator for the partially linear (semiparametric) regression of panel data models with fixed individual effects (2). This estimator relies on first-differencing and uses kernel instrumental variable regression with lagged explanatory variables as instruments for the estimation of the nonparametric part of the regression equation. However, Baltagi and Li (2002) showed that the estimator of Li and Stengos (1996) suffers from the 'curse of dimensionality' and that it can be used to estimate the

[4] In this context, separability means that the individual and time effects are separable from each other and from (the effects of) the explanatory variables, i.e. $\partial f(.)/\partial x_{it}$ do not depend on i or t, $\partial f(.)/\partial i$ does not depend on x_{it} or t, and $\partial f(.)/\partial t$ does not depend on x_{it} or i.

[5] This does not necessarily rule out nonlinear relationships between the dependent variable and the explanatory variables (see footnote 3).

function $F^*(x_{i,t}, x_{i,t-1}, t) \equiv f^*(x_{i,t}, t) - f^*(x_{i,t-1}, t-1)$ in (4) but it cannot be used to estimate the original unknown function $f^*(x_{i,t}, t)$, which is of primary interest in nonparametric regression analysis. Therefore, Baltagi and Li (2002) proposed to estimate semiparametric panel data models with fixed individual effects (2) in the first-differenced form (6), where the nonparametric part is modelled using a 'series estimation' method (e.g. a power series[6] or a series of spline functions). Ai and Li (2008) also used 'series estimation' to address the same problem as Baltagi and Li (2002). However, while Baltagi and Li (2002) use first-differencing, Ai and Li (2008) use the "within" transformation and estimate model (2) in the form (3).[7] Alternative solutions to the problems of Li and Stengos's (1996) estimator are the kernel marginal integration method of Linton and Nielsen (1995) and the backfitting method of Buja, Hastie and Tibshirani (1989). These methods are compared in Opsomer and Ruppert (1997).

By applying a Taylor-series approximation to the unknown (true) regression function $f^*(.)$, Ullah and Roy (1998) proposed a method for the nonparametric estimation of one-way individual fixed effects models (2) that uses a conventional "within" transformation (5) but uses the original regressors x_{it} rather than $\tilde{x}_{it}$ in the kernel (weighting) functions. However, Lee and Mukherjee (2008) showed that nonparametric estimators that use the conventional "within" transformation or first differencing (e.g. the estimators of Li and Stengos (1996) and Ullah and Roy (1998)) are biased and that the bias does not decrease with sample size. As a solution to this problem, the authors proposed a local-linear kernel estimator for models with fixed individual effects (2) or fixed individual and time effects (7) that uses the locally weighted averages of the regressors for the "within" transformation.[8] Lee and Mukherjee (2008) provided the asymptotic properties of this estimator and showed that it is asymptotically unbiased and consistent.

Henderson and Ullah (2005) proposed a generalisation of the method of Ullah and Roy (1998). This local-linear weighted least squares estimator uses information from the covariance matrix of the disturbance vector to obtain a consistent estimator for random effects models. However, in most empirical economic applications, the "random effects" estimator is found to be inconsistent, because the individual (or time) effects are correlated with the regressors (Henderson, Carroll and Li, 2008).

Based on the marginal nonparametric kernel regression method for panel data introduced by Wang (2003), Henderson, Carroll and Li (2008) proposed an iterative procedure for nonparametric kernel estimations of panel data models with fixed individual effects (2). This method uses the profile likelihood method to estimate the differenced form of the

[6] However, since power series estimators are sensitive to outliers, their use is rather limited.
[7] Ai and Li (2008) provide an in-depth discussion of nonparametric 'series estimation' approaches.
[8] Since there is no averaging in first-difference transformations, the method of Lee and Mukherjee (2008) cannot be applied to the first-difference estimator.

unknown regression function as specified in (4).[9] The approaches of Wang (2003) and Henderson, Carroll and Li (2008) are extensively discussed in Li and Racine (2007).

In summary, all the methods discussed above use a parametric panel data specification—either (2) or (7)—and some kind of data transformation, which is often a generalisation of the conventional panel data transformations.

2.3. Fully nonparametric panel data specification

All nonparametric panel data estimators discussed in the previous section assume that the individual effects (and/or time effects) are additive and separable. These parametric assumptions about the panel data structure contradict the idea of nonparametric regression and hence make the use of nonparametric regression questionable (Racine, 2009). Therefore, Altonji and Matzkin (2005), Evdokimov (2010), and Hoderlein and White (2012) proposed nonparametric estimation procedures of nonseparable panel data models. Although these methods do not rely on the restrictive assumption of additive and separable individual effects (and/or time effects), they still rely on first-differencing. However, the first differencing results in a considerable loss of observations (degrees of freedom), particularly when the time dimension of the panel data set is short or when the panel is highly unbalanced, both of which are not uncommon in many panel data sets. The reduction in the number of observations is a substantial drawback in nonparametric regression, because these methods require a large number of observations. Moreover, to our knowledge, these estimators are not available in any software package, which currently limits their use in applied research.

In our analysis, we use a different specification of a fully nonparametric and nonseparable panel data model that has been suggested by Henderson and Simar (2005), Racine (2008), and Gyimah-Brempong and Racine (2010). They estimate equation (1) as a fully nonparametric two-ways effects panel data model with time (t) and individual ID (i) as categorical explanatory variables using the nonparametric regression method proposed by Li and Racine (2004) and Racine and Li (2004) that can handle both continuous and categorical explanatory variables. This estimator does not require any data transformation (e.g. "within" transformation or first differencing) so that it does not suffer from a loss of observations. Furthermore, this approach does not assume additivity or separability of the individual or time effects. This means that the level of the dependent variable ("intercept") and also the marginal effects of the explanatory variables on the dependent variable ("slopes") may differ between time periods and between individuals and the time effects may depend on the individual, while the individual effects may vary over time.

[9] Henderson, Carroll and Li (2008) use a slightly different differencing than usual, namely $y_{it} - y_{i,1} = f^*(x_{it}, t) - f^*(x_{i,1}, 1)$, i.e. subtracting the variables from the first time period rather than subtracting the variables from the previous time period, but they claim that the asymptotic properties are similar to the standard first-differencing approach.

Hence, this estimator does not imply any restrictions on the most general specification of panel data models, i.e. function $f(.)$ in (1). Furthermore, the bandwidths of the explanatory variables can be selected using data driven cross-validation methods. Thus, the overall shape of the relationship between the dependent variable and the covariates x_{it}, the individual i, and time t is entirely determined by the data. Therefore, we call it fully nonparametric panel data specification.

The only arbitrary choice to be made by the researcher is the choice of the kernel function. However, it has been shown (e.g. in Silverman, 1986; Taylor, 1989; Racine and Li, 2004) that this choice is less important than the selection of the smoothing parameters (bandwidths). Initially, nonparametric kernel regression methods could only include continuous explanatory variables and they usually used the Gaussian kernel or the Epanechnikov kernel for these variables. In this study, we use a more recent local-linear regression method proposed by Li and Racine (2004) and Racine and Li (2004) that can also include kernels for categorical explanatory variables using generalised product kernels. In this approach the kernel suggested by Aitchison and Aitken (1976) or the kernel suggested by Li and Racine (2004) can be used for unordered categorical variables, while the kernel suggested by Wang and van Ryzin (1981) or the kernel suggested by Racine and Li (2004) can be used for ordered categorical variables.[10] The statistical significance levels of the explanatory variables can be obtained by bootstrapping using the methods proposed by Racine (1997) and Racine, Hart and Li (2006).

3. Empirical application

We use the production function framework to empirically illustrate the use of the parametric and nonparametric panel data estimators. However, our considerations apply to panel data regression in general.

We use a balanced panel data set from the Polish Farm Accountancy Data Network (FADN), which consists of 342 farms specialised in crop production during the period 2004 to 2010. Hence, our data set includes 2,394 observations in total. The dependent variable of the production function is the farms' output, which is measured as the value of the total agricultural production. Four explanatory variables (inputs) are used in the regression analyses: labour (L), land (A), intermediate inputs (V), and capital (K).[11]

We take the logarithms of all variables. This approach has three advantages. First, the log-transformed values are more equally distributed, which is particularly desirable when fixed bandwidths in local-linear kernel regression are used. Second, the unknown true relationship between the input quantities and the output quantity is likely much closer to a log-linear relationship (Cobb-Douglas technology) than to a linear relationship (linear

[10] The mathematical specifications of these kernels are given in Appendix B.

[11] This data set is extensively discussed in Czekaj and Henningsen (2012). Therefore, we only describe it briefly here.

technology) so that the use of logarithmic quantities of the inputs and the output allows for larger bandwidths, which in turn increases the precision of the local-linear estimates, because they are based on a larger number of observations. Third, the estimated gradients of nonparametric regression can be directly interpreted as partial output elasticities in the production function framework.

4. Results

All estimations and calculations were conducted within the statistical software environment "R" (R Development Core Team, 2012) using the add-on package "plm" (Croissant and Millo, 2008) for estimation and testing of parametric panel data models, the add-on package "lmtest" (Zeileis and Hothorn, 2002) for further parametric specification tests, and the add-on package "np" (Hayfield and Racine, 2008) for nonparametric regression and nonparametric specification tests.[12]

4.1. Fully parametric panel data estimations

We start with traditional parametric estimations of a "pooled" model (i.e. ignoring the panel structure of the data set) and the panel data models that are defined in equations (5), (6), and (9), where we estimated specifications (5) and (9) additionally with the "random effects" estimator. We use the two functional forms that are most widely applied in econometric production analysis, i.e. the log-linear (Cobb-Douglas) and the log-quadratic (Translog) functional forms. The estimation results are summarised in Table 1. The panel data specification generally had a greater effect on the estimation results (e.g. the partial output elasticities and the elasticities of scale at the sample mean) than the specification of the functional form. The most salient result is that the partial output elasticities of land and intermediate inputs are around 0.15 and 0.8, respectively, if individual heterogeneity is ignored, while these elasticities are around 0.5 and 0.4, respectively, if the models allow for additive and separable fixed individual effects. The estimates of the random effects models take a middle ground between these two groups of estimators. Moreover, the first-difference models indicate decreasing returns to scale, while all other estimates indicate increasing returns to scale.

We apply statistical tests in order to check, which model gives us the most reliable results. Standard parametric tests indicate that the log-quadratic (Translog) two-ways fixed effects model is the most suitable (least unsuitable) parametric model specification, because it only suffers from serial correlation, while all other specifications are rejected by at least two tests.[13]

[12] The R commands used for this analysis are available in Appendix C.

[13] Poolability tests and the Lagrange Multiplier tests of Breusch and Pagan for individual, time, and two-ways effects indicate that individual effects and time effects are significant (tested separately as well as jointly); Hausman (1978) tests indicate that all four random effects estimators are inconsistent; a

Table 1: Results of fully parametric estimations of panel data models

Model	form	ID	Year	Labour	Land	Int. inp.	Capital	Scale
Pool	CD	—	—	0.052**	0.121***	0.815***	0.080***	1.068
1WWI	CD	0.236***	—	0.065*	0.453***	0.472***	0.052**	1.043
1WWT	CD	—	0.064***	0.054**	0.128***	0.808***	0.080**	1.070
1WRI	CD	0.223***	—	0.067**	0.347***	0.581***	0.081***	1.076
1WRT	CD	—	0.057***	0.054**	0.128***	0.808***	0.080***	1.070
FD	CD	> 0	—	0.044	0.512***	0.339***	0.017	0.911
2WW	CD	0.243***	0.072***	0.064*	0.507***	0.448***	0.047**	1.066
2WR	CD	0.229***	0.071***	0.068**	0.388***	0.548***	0.076***	1.080
Pool	TL	—	—	0.050**	0.161***	0.754***	0.088***	1.053
1WWI	TL	0.239***	—	0.060+	0.506***	0.413***	0.054**	1.030
1WWT	TL	—	0.064***	0.051**	0.169***	0.746***	0.089***	1.055
1WRI	TL	0.223***	—	0.063*	0.393***	0.518***	0.091***	1.065
1WRT	TL	—	0.058***	0.051**	0.169***	0.746***	0.089***	1.055
FD	TL	> 0	—	0.073+	0.522***	0.294***	0.001	0.890
2WW	TL	0.244***	0.072***	0.050+	0.562***	0.390***	0.058*	1.060
2WR	TL	0.231***	0.071***	0.058*	0.436***	0.486***	0.091***	1.071

Notes: model "1WWI" denotes the one-way within specification with fixed individual effects (5), model "1WWT" denotes the one-way within specification with fixed time effects, "FD" denotes the first-difference specification (6), "2WW" denotes the two-ways within specification (9), "1WRI" indicates the one-way specification with random individual effects, and "2WR" indicates the two-way random effects estimator. "CD" denotes to the log-linear (Cobb-Douglas) functional form and "TL" denotes to the log-quadratic (Translog) functional form. The numbers in the columns "ID" and "Year" indicate the standard deviations of the individual effects and time effects, respectively. The numbers in the following columns indicate the partial output elasticities of the inputs and the elasticity of scale evaluated at the mean values of the explanatory variables. The asterisks indicate the significance levels of the explanatory variables (not the significance levels of the corresponding standard deviations or elasticities), where $^+$ indicates $P < 0.10$, * indicates $P < 0.05$, ** indicates $P < 0.01$, and *** indicates $P < 0.001$.

Moreover, we use two variations of the bootstrap nonparametric consistent model specification test described in Hsiao, Li and Racine (2007) to test the functional forms of the parametric models. One specification uses a local-constant and the other specification uses a local-linear kernel estimation as the nonparametric counterpart to the tested parametric specification. The Cobb-Douglas functional form is not rejected by the local-linear version of the test, but it is rejected at the 10% significance level by the local-constant version. We found that for our data set only the two-ways fixed effects model with the Translog functional form is accepted by the nonparametric consistent model specification

Wald test rejects the log-linear (Cobb-Douglas) specification in favour of the log-quadratic (Translog) specification; and the regression error specification test (RESET) proposed by Ramsey (1969) rejects the linearity of the log-linear specification but accepts the log-quadratic (Translog) specification with a P-value of 0.112. We also estimated models in the first-difference specification (6). To investigate whether the first-difference estimator or the "within" estimator is more suitable for our data set, we conducted tests for serial correlation that have been suggested by Wooldridge (2002). We found serial correlation in both models. Therefore, we cannot conclude which of these two estimators is more efficient. Detailed test results are given in Tables A1 to A6 in Appendix A.

test (see Table A6 in Appendix A), which is also the most suitable model according to the parametric tests.

4.2. Fully nonparametric panel data estimations

We estimate the most general panel data specification (1) by nonparametric kernel regression, where we use the ID of the individual farm (i) and the time (t) as additional (categorical) explanatory variables using the estimator proposed by Li and Racine (2004) and Racine and Li (2004), because it can account for both continuous and categorical explanatory variables. This estimator allows the effect of one explanatory variable to depend on the values of all other explanatory variables, e.g. the effect of a continuous explanatory variable may differ between individuals and between time periods.

In the nonparametric kernel regressions, we can either use the Epanechnikov kernel or the Gaussian kernel for the continuous explanatory variables. The categorical regressors may be modelled either as ordered or unordered categorical variables. For ordered categorical variables, we can either use the Wang and van Ryzin (1981) kernel or the Li and Racine (2004) kernel for ordered categorical variables. For unordered categorical variables, we can either use the Aitchison and Aitken (1976) kernel or the Li and Racine (2004) kernel for unordered categorical variables. While the ID of the individual farm is clearly an unordered categorical variable, time can either be considered as an ordered categorical variable or as an unordered categorical variable (Czekaj and Henningsen, 2012).

We make the frequently used assumption that the bandwidths (smoothing parameters) for the explanatory variables can differ between regressors but are constant over the domain of each regressor. The bandwidths (smoothing parameters) of the kernel functions are chosen according to the expected Kullback-Leibler cross-validation criterion (Hurvich, Simonoff and Tsai, 1998). Hence, the researcher's only arbitrary decision is the choice of the kernel. In order to increase the validity and objectivity of our results, we estimated our fully nonparametric panel data model with all combinations of the different kernels and different specifications of the time variable. As the ID variable and time were not always statistically significant (at the 5% significance level), we estimated two additional models: one model without a time variable (but with an ID variable) and another model without a time or ID variable. The latter specification is effectively a "pooled" model that completely ignores the panel structure of our data set.

The cross-validated (optimal) bandwidth for the ID variable clearly depends on the kernel functions that are used for the categorical explanatory variables (i.e. ID and time). If the Li and Racine (2004) kernel for unordered categorical variables is used for the ID variable, the bandwidth of this variable is either 0.006 (if the Wang and van Ryzin (1981) kernel is used for unordered categorical variables) or 0.003 (otherwise). If the Aitchison and Aitken (1976) kernel is used for the ID variable, the bandwidth of this variable is either around 0.68 (if the Wang and van Ryzin (1981) kernel is used for unordered categorical variables) or 0.52 (otherwise) (Table 2). While the bandwidth of the Li and Racine (2004) kernel for unordered categorical variables directly indicates the weight of the other levels (farms) relative to the current farm (all other variables being equal), this is not the case for the Aitchison and Aitken (1976) kernel. For this kernel, the weight of the other levels (farms) relative to the current level (all other variables being equal) is $\lambda/((c-1)(1-\lambda))$, where c indicates the total number of levels (= 342 farms in our application) and λ indicates the bandwidth (see Appendix B.1). In our application, the relative weights based on the Aitchison and Aitken (1976) kernel are virtually identical to the relative weights based on the Li and Racine (2004) kernel for unordered categorical variables: 0.006 if the Wang and van Ryzin (1981) kernel is used for unordered categorical variables and 0.003 otherwise. As Aitchison and Aitken (1976) kernel and the Li and Racine (2004) kernel for unordered categorical variables are basically the same kernel function but just use different measures (normalisations) of the smoothing parameter (bandwidth), the cross-validation resulted in virtually identical relative weights so that the estimation results do not depend on the choice of the kernel for unordered categorical variables. The low weights of the other farms relative to the current farm allows for considerable variations in the production technology between the individual farms (Table 3). According to the bootstrap significance test proposed by Racine (1997) and Racine, Hart and Li (2006), the ID variable has a statistically significant effect, at least at the 10% significance level (Table 3).

In all model specifications, i.e. no matter which kernel functions are used, the bandwidth of the time variable is at its upper boundary (Table 2), which is one for the Wang and van Ryzin (1981) kernel and the Li and Racine (2004) kernels and $(c-1)/c$ for the Aitchison and Aitken (1976) kernel, where c indicates the total number of levels (= 7 years in our application) (see Appendix B.2). In case of the two Li and Racine (2004) kernels and the Aitchison and Aitken (1976) kernel, the effect of time is smoothed out, because the current year always has the same weight as the other years so that there is no variation over time (all other variables being equal) (Table 3). In contrast, the Wang and van Ryzin (1981) kernel cannot smooth out time, because the current year always has at least twice the weight as the other years (see Appendix B.2). Therefore, the models with the Wang and van Ryzin (1981) kernel for the time variable still allow for variation over time even though the bandwidth is one (Tables 2 and 3). The bootstrap significance test proposed by Racine (1997) and Racine, Hart and Li (2006) even indicates that the time variable has a statistically significant effect if the Wang and van Ryzin (1981) kernel is used for the time variable (Table 3).[14] As the models with the Wang and van Ryzin (1981) kernel

[14] This test also indicates that time has a highly statistically significant effect in the model with the Gaussian kernel for continuous variables, the Aitchison and Aitken (1976) kernel for unordered categorical variables, and the Li and Racine (2004) kernel for ordered categorical variables, although the effect of time is smoothed out in this model. This seems to be strange and needs further investigation in the future.

are the only models that allow for variation over time (all other variables being equal), these models have a considerably larger cross-validated bandwidth of the ID variable, probably because the removal of the variation over time leaves less variation that could be attributed to the ID variable. One could argue that the Wang and van Ryzin (1981) kernel is generally unsuitable, because it does not allow for a variable to be smoothed out even if it is completely irrelevant. However, as time has a statistically significant effect in all of our models with the Wang and van Ryzin (1981) kernel for the time variable, it might also be argued that the Wang and van Ryzin (1981) kernel can be particularly useful in specific situations, because it can prevent statistically significant variables from being smoothed out.

In all model specifications except for the two "pooled" estimations, the cross-validation suggests using rather large bandwidths for the (logarithmic) input quantities (Table 2). This means that the gradient with respect to a continuous explanatory variable ($\partial f(.)/\partial x_{jit}$) does not depend on the value of this variable (x_{jit}). However, the gradients are not necessarily the same for all observations (as in the Cobb-Douglas functional form), because the gradients may depend on the values of other explanatory variables (including the ID variable and time). This is indeed the case, as the estimated gradients vary considerably across the observations. According to the bootstrap significance test proposed by Racine (1997) and Racine, Hart and Li (2006), all continuous regressors (logarithmic input quantities) have a statistically significant effect on the dependent variable (logarithmic output quantity) in all model specifications except for the two "pooled" estimations (Table 3). The choice of the kernel for the continuous explanatory variables only affects the (statistically insignificant) results of the "pooled" model but does not affect the estimates of the other models. However, this observation should not be generalised, because in our application the irrelevance of the choice of the kernel is caused by the very large bandwidths of the continuous explanatory variables.

Although the choice of the kernels for continuous variables and ordered categorical variables (but not the choice between the two kernels for unordered categorical variables with successfully cross-validated bandwidths) could affect the estimation results, our results of the fully nonparametric panel data estimations do not notably depend on the choice of the kernel, e.g. the partial output elasticities are very similar across all model specifications and the elasticity of scale is always around 1.07 (Table 3). As the effect of time is smoothed out in most model specifications, the removal of the time variable as an explanatory variable does not crucially affect the results either. However, the "pooled" regression without the time and the ID as explanatory variables partly gives different estimation results while almost no continuous input variable (logarithmic input quantity) has a statistically significant effect on the dependent variable (logarithmic output quantity). This indicates—together with the statistical significance of the ID variable—that there is

Table 2: Bandwidths and R^2 values from fully nonparametric estimations of panel data models

kernels			ID	year	$\log(L)$	$\log(A)$	$\log(V)$	$\log(K)$	R^2
E	LRU	LRO	0.003	1.000	2316379	633905	406870	543084	0.952
E	LRU	WVR	0.006	1.000	372534	851210	260011	1031526	0.956
G	LRU	LRO	0.003	1.000	7525905	6768702	2483181	2948212	0.952
G	LRU	WVR	0.006	1.000	2992012	1897999	1063737	718268	0.956
E	AA	LRO	0.520	1.000	642045	454157	635336	1152330	0.952
E	AA	WVR	0.680	1.000	297105	1102811	6459913	14735939	0.955
G	AA	LRO	0.520	1.000	383454	389440	11385661	6398439	0.952
G	AA	WVR	0.677	1.000	21460075	1992734	1071484	17226491	0.956
E	LRU	LRU	0.003	1.000	1615763	724623	341968	519111	0.952
E	AA	AA	0.520	0.857	1766820	8571562	1395782	1860381	0.952
G	LRU	LRU	0.003	1.000	698517	1886805	251145	4821524	0.952
G	AA	AA	0.520	0.857	5694530	6108610	4872012	8777948	0.952
E	LRU	—	0.003	—	1528159	1918044	325765	361992	0.952
E	AA	—	0.520	—	736114	672573	2894058	12216425	0.952
G	LRU	—	0.003	—	1302064	1398033	365152	418384	0.952
G	AA	—	0.520	—	19559263	4317483	6661354	4884191	0.952
E	—	—	—	—	362667	0.181	0.413	0.375	0.905
G	—	—	—	—	286273	0.173	0.389	0.386	0.922

Notes: the first, second, and third abbreviation in the "kernels" column indicate the kernel for (continuous) input quantities, the kernel for the (unordered categorical) ID variable, and the kernel for the (ordered or unordered categorical) time variable, respectively. Kernel "E" indicates the Epanechnikov kernel for continuous explanatory variables, "G" indicates the Gaussian kernel for continuous explanatory variables, "LRO" indicates the Li and Racine (2004) kernel for ordered categorical variables, "LRU" indicates the Li and Racine (2004) kernel for unordered categorical variables, "WVR" indicates the Wang and van Ryzin (1981) kernel for ordered categorical variables, and "AA" indicates the Aitchison and Aitken (1976) kernel for unordered categorical variables.

considerable heterogeneity between the individual farms so that the "pooled" regression is unsuitable.

4.3. Comparison of different panel data specifications

When comparing the results in Tables 1 and 3, it is surprising that the mean partial output elasticities and the elasticities of scale from the (fully) nonparametric models are very similar to the corresponding elasticities that are estimated by the parametric models that do not allow for individual effects (e.g. pooled estimations), although these models were strongly rejected by several statistical tests. In contrast, the partial output elasticities of land and intermediate inputs strongly differ between the fully nonparametric

Table 3: Significance levels and elasticities from fully nonparametric estimations of panel data models

	kernels		ID	Year	Labour	Land	Int. inp.	Capital	Scale
E	LRU	LRO	0.157^{+}	0.000	0.056^{***}	0.168^{***}	0.760^{***}	0.088^{***}	1.071
E	LRU	WVR	0.133^{*}	0.032^{**}	0.054^{***}	0.147^{***}	0.785^{***}	0.084^{***}	1.070
G	LRU	LRO	0.157^{+}	0.000	0.056^{***}	0.168^{***}	0.760^{***}	0.088^{***}	1.071
G	LRU	WVR	0.133^{*}	0.032^{**}	0.054^{***}	0.147^{***}	0.785^{***}	0.084^{***}	1.070
E	AA	LRO	0.157^{+}	0.000	0.056^{***}	0.168^{***}	0.760^{***}	0.088^{***}	1.071
E	AA	WVR	0.131^{*}	0.032^{**}	0.054^{***}	0.145^{***}	0.787^{***}	0.084^{***}	1.070
G	AA	LRO	0.157^{+}	0.000^{***}	0.056^{***}	0.168^{***}	0.760^{***}	0.088^{***}	1.071
G	AA	WVR	0.131^{*}	0.032^{**}	0.054^{***}	0.146^{***}	0.786^{***}	0.084^{***}	1.070
E	LRU	LRU	0.157^{+}	0.000	0.056^{***}	0.168^{***}	0.760^{***}	0.088^{***}	1.071
E	AA	AA	0.157^{+}	0.000	0.056^{***}	0.168^{***}	0.760^{***}	0.088^{***}	1.071
G	LRU	LRU	0.157^{+}	0.000	0.056^{***}	0.168^{***}	0.760^{***}	0.088^{***}	1.071
G	AA	AA	0.157^{+}	0.000	0.056^{***}	0.168^{***}	0.760^{***}	0.088^{***}	1.071
E	LRU	—	0.157^{+}		0.056^{***}	0.168^{***}	0.760^{***}	0.088^{***}	1.071
E	AA	—	0.157^{+}		0.056^{***}	0.168^{***}	0.760^{***}	0.088^{***}	1.071
G	LRU	—	0.157^{+}		0.056^{***}	0.168^{***}	0.760^{***}	0.088^{***}	1.071
G	AA	—	0.157^{+}		0.056^{***}	0.168^{***}	0.760^{***}	0.088^{***}	1.071
E	—	—			0.052	0.385	0.769	0.100	1.306
G	—	—			0.063^{***}	0.168	0.759	0.098	1.087

Notes: the model specifications and abbreviations are explained below table 2. The numbers in the columns "ID" and "Year" indicate the standard deviations of the individual effects and time effects, respectively. The numbers in the following columns indicate the mean partial output elasticities of the inputs and the mean elasticity of scale. The significance levels are explained below table 1.

panel data models and the most suitable parametric model (and all other parametric models with fixed individual effects), although the small bandwidth of the ID variable allows for considerable variation between individual farms in the nonparametric model—similarly to the fixed individual effects in the parametric models. However, while the fully nonparametric panel data models allow the effects of the explanatory variables ("slopes") to differ between individuals, the parametric models do not allow for this. Furthermore, although the mean elasticities from the nonparametric models are similar to the elasticities from the parametric models with fixed individual effects derived at the sample mean, the elasticities evaluated at each individual observation considerably differ between the parametric models and the nonparametric models (see also Czekaj and Henningsen, 2012).

We use economic reasoning to further assess and compare the suitability of the nonparametric models and the most suitable parametric model. In case of profit maximisation, the marginal value products of the variable inputs are expected to be equal to the prices of these inputs. In our application, the intermediate input is perfectly suited for such an analysis, because it is the only input that the farmers in our sample can freely adjust at a given market price and the difference of the marginal value products between the two types of models is very large. We calculate the ratio of the costs of intermediate inputs and the total revenue for each individual observation and compare these ratios to the estimated output elasticities of the intermediate inputs. If the farmers maximise profit and the estimated production function is correct, then the ratio of the costs of intermediate inputs and the total revenue should be equal to the partial output elasticities.[15] Table 4 indicates the mean values and the median values of the partial output elasticities of intermediate inputs that were estimated by two nonparametric models and the most suitable parametric model. Furthermore, the table indicates the mean squared differences (MSDs) and the mean absolute differences (MADs) between the elasticities and the ratio between the costs of intermediate inputs and the total revenue.

Table 4: Partial output elasticities of intermediate inputs

	Mean	Median	MSE to c_V/R	MAD to c_V/R
c_V/R	0.6507	0.6246	—	—
ϵ_V (E LRU)	0.7596	0.7653	0.0718	0.2036
ϵ_V (E LRU WVR)	0.7851	0.7850	0.0715	0.2081
ϵ_V (TL 2WW)	0.4095	0.3984	0.1281	0.2769

Notes: ϵ_V denotes the partial output elasticity of intermediate inputs. c_V/R is the ratio between the costs of intermediate inputs and the total revenue for each individual observation, which should be equal to ϵ_V in case of profit maximisation. The abbreviations of the econometric models are explained below Tables 1 and 2.

The mean (and median) values of the ratio between the costs of intermediate inputs and the total revenue lie in between the mean (and median) values of the estimated partial output elasticities of the intermediate inputs from the fully nonparametric models and the fully parametric model but they are much closer to the estimates from the fully nonparametric models. Moreover, the mean squared deviations and the mean absolute deviations between the costs of intermediate inputs and the total revenue and the estimated partial output elasticities of the intermediate inputs are considerably smaller for the fully nonparametric models than for the fully parametric model. Finally, the nonparametric regression indicates that the farmers generally use slightly too few intermediate inputs, while the parametric regression indicates that most farmers use much too much intermediate inputs, of which the former case is much more plausible, as it could be explained by risk aversion and credit constraints. This indicates that the estimation results from the fully nonparametric panel data models are more reliable than the estimation results from the fully parametric panel data model—at least in our data set.

[15]The marginal value product of ith input is defined as $MVP_i = P_o \cdot (\partial Q_o/\partial Q_i)$, where P_o is the output price, Q_o is the output quantity and Q_i is the quantity of the ith input. The partial output elasticity of the ith input is defined as $\epsilon_i = (\partial Q_o/\partial Q_i) \cdot (Q_i/Q_o)$. Profit maximisation implies $MVP_i = P_i$, where P_i is the price of the ith input. By re-arranging, we get: $\epsilon_i = (\partial Q_o/\partial Q_i) \cdot (P_o\, Q_i)/(P_o\, Q_o) = MVP_i \cdot Q_i/(P_o\, Q_o) = (P_i\, Q_i)/(P_o\, Q_o)$.

5. Conclusion

Nonparametric regression methods are becoming increasingly popular in econometric analyses because they avoid the specification of a potentially unsuitable functional form for the relationship between the explanatory variables and the dependent variable. However, at the moment, nonparametric methods are not used very often for the analysis of panel data. In this paper we discuss different approaches to the nonparametric estimation of panel data regression models and we distinguish between (i) nonparametric regression models with parametric panel data specifications and (ii) nonparametric regression models with a nonparametric panel data specifications. As the assumption of additive and separable individual (or time) effects in the parametric panel data specifications contradicts the idea of nonparametric regression (Racine, 2009), we apply a nonparametric regression model with a fully nonparametric panel data specification (Henderson and Simar, 2005; Racine, 2008; Gyimah-Brempong and Racine, 2010) in our empirical application. In this specification, the firms' individual identifiers and the time variable are used as additional (categorical) explanatory variables in the nonparametric regression, thereby taking account of possible individual and time heterogeneity in the data. Compared to other nonparametric regression models with fully nonparametric panel data specifications, our specification does not rely on first differencing and hence, does not suffer from a reduction in the number of observations, particularly relevant in short and unbalanced panel data sets. Moreover, we discuss the use of different kernel functions for the ID and time variables and we show how the bandwidths of different kernels for categorical variables (e.g. the ID and time variables) can be compared. Finally, we compare the estimates from the fully nonparametric regression model with estimates from traditional fully parametric panel data methods.

In our empirical application, we estimate the production technology of Polish crop farmers based on a balanced panel data set with 2,394 observations in total. We found that the choice of the kernel for the ID and time variables only had a minor influence on the results. However, we found some economically relevant differences between the estimates of the fully nonparametric regression model and the estimates of traditional fully parametric estimators. Based on economic reasoning, we found that the estimates of the fully nonparametric panel data regression model were more reliable than the estimates of the traditional fully parametric panel data model.

Acknowledgements

The authors would like to thank Jeff Racine for his many very helpful comments and suggestions regarding nonparametric regression, Mette Asmild for her helpful suggestions regarding an earlier draft of this paper, and Lech Goraj as well as the Agricultural Accountancy Department of IAFE-NRI (IERiGŻ-PIB) in Warsaw for providing access to the data. The authors are also grateful to the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) for financially supporting this research. Of course, all remaining errors are the sole responsibility of the authors.

Appendix

A. Results of diagnostic tests for Cobb-Douglas and Translog production functions

Table A1: Poolability tests for parametric panel data models

Models	Statistics	Decision
CD pooled vs 1WWI	$F(341, 2048) = 5.355$, $p < 0.001$	pooled model rejected
CD pooled vs 1WWT	$F(6, 2383) = 18.508$, $p < 0.001$	pooled model rejected
CD pooled vs 2WW	$F(347, 2042) = 6.538$, $p < 0.001$	pooled model rejected
CD pooled vs 1VCI	$F(1705, 684) = 2.1072$, $p < 0.001$	pooled model rejected
CD pooled vs 1VCT	$F(30, 1339) = 5.6442$, $p < 0.001$	pooled model rejected
CD 1WWI vs 1VCI	$F(1364, 684) = 1.1561$, $p = 0.015$	1WWI model rejected
CD 1WWT vs 1VCT	$F(24, 2359) = 2.3648$, $p < 0.001$	1WWT model rejected
TL pooled vs 1WWI	$F(341, 2038) = 5.251$, $p < 0.001$	pooled model rejected
TL pooled vs 1WWT	$F(6, 2373) = 19.103$, $p < 0.001$	pooled model rejected
TL pooled vs 2WW	$F(347, 2042) = 6.451$, $p < 0.001$	pooled model rejected
TL pooled vs 1VCT	$F(90, 2289) = 3.1818$, $p < 0.001$	pooled model rejected
TL 1WWT vs 1VCT	$F(84, 2289) = 1.9964$, $p < 0.001$	1WWT model rejected

Notes: "1VCI" indicates the one-way varying coefficient model, where all coefficients may differ between individuals; "1VCT" indicates the one-way varying coefficient model, where all coefficients may differ between time periods; all other abbreviations are explained below table 1. The "1VCI" cannot be estimated for the Translog functional form, because this functional form has more parameters (11) than the number of time periods in our data set (7).

Table A2: Breusch and Pagan tests for parametric panel data models

Models	Statistics	Decision
CD individual	$\chi^2(1) = 685.357$, $p < 0.001$	significant individual effects
CD time	$\chi^2(1) = 707.415$, $p < 0.001$	significant time effects
CD two-ways	$\chi^2(2) = 1392.772$, $p < 0.001$	significant individual and time effects
TL individual	$\chi^2(1) = 644.163$, $p < 0.001$	significant individual effects
TL time	$\chi^2(1) = 756.113$, $p < 0.001$	significant time effects
TL two-ways	$\chi^2(2) = 1400.275$, $p < 0.001$	significant individual and time effects

Note: The abbreviations of the model specifications are explained below table 1.

Table A3: Hausman tests for parametric panel data models

Models	Statistics	Decision
CD 1WWI vs 1WRI	$\chi^2(4) = 109.733$, $p < 0.001$	random effects model rejected
CD 1WWT vs 1WRT	$\chi^2(4) = 12.876$, $p = 0.012$	random effects model rejected
CD 2WW vs 2WR	$\chi^2(4) = 121.808$, $p < 0.001$	random effects model rejected
TL 1WWI vs 1WRI	$\chi^2(14) = 113.087$, $p < 0.001$	random effects model rejected
TL 1WWI vs 1WRT	$\chi^2(14) = 39.163$, $p < 0.001$	random effects model rejected
TL 2WW vs 2WR	$\chi^2(14) = 124.391$, $p < 0.001$	random effects model rejected

Note: The abbreviations of the model specifications are explained below table 1.

Table A4: Tests for serial correlation in the parametric panel data models

Models	Statistics	Decision
Wooldridge's test for serial correlation of FD residuals ($\Delta\epsilon_{it}$)		
CD	$\chi^2(1) = 214.112$, $p < 0.001$	rejected (serial correlation in $\Delta\epsilon_{it}$)
TL	$\chi^2(1) = 213.507$, $p < 0.001$	rejected (serial correlation in $\Delta\epsilon_{it}$)
Breusch-Godfrey/Wooldridge's test for serial correlation of FE residuals ($\tilde{\epsilon}_{it}$)		
CD	$\chi^2(7) = 259.18$, $p < 0.001$	rejected (serial correlation in $\tilde{\epsilon}_{it}$)
TL	$\chi^2(7) = 260.64$, $p < 0.001$	rejected (serial correlation in $\tilde{\epsilon}_{it}$)
Wooldridge's test for serial correlation of FE residuals ($\tilde{\epsilon}_{it}$) in "short" panels		
CD	$\chi^2(1) = 8.217$, $p = 0.004$	rejected (serial correlation in $\tilde{\epsilon}_{it}$)
TL	$\chi^2(1) = 8.334$, $p = 0.004$	rejected (serial correlation in $\tilde{\epsilon}_{it}$)

Note: The abbreviations of the model specifications are explained below table 1.

Table A5: Wald tests of CD vs. TL parametric panel data models

Models	Statistics	Decision
TL pooled vs. CD pooled	$\chi^2(10) = 52.357$, $p < 0.001$	CD pooled rejected
TL 1WWI vs. CD 1WWI	$\chi^2(10) = 30.581$, $p < 0.001$	CD 1WWI rejected
TL 1WWT vs. CD 1WWT	$\chi^2(10) = 56.171$, $p < 0.001$	CD 1WWT rejected
TL 2WW vs. CD 2WW	$\chi^2(10) = 35.561$, $p < 0.001$	CD 2WW rejected
TL 1WRI vs. CD 1WRI	$\chi^2(10) = 33.339$, $p < 0.001$	CD 1WRI rejected
TL 1WRT vs. CD 1WRT	$\chi^2(10) = 56.040$, $p < 0.001$	CD 1WRT rejected
TL 2WR vs. CD 2WR	$\chi^2(10) = 39.109$, $p < 0.001$	CD 2WR rejected
TL FD vs. CD FD	$\chi^2(10) = 17.101$, $p = 0.072$	CD FD rejected

Note: The abbreviations of the model specifications are explained below table 1.

Table A6: Tests of the functional form of parametric panel data models

Model	RESET		NPCMSTEST LC		NPCMSTEST LL	
CD pooled	$RESET(8, 2381) = 3.687$, $p < 0.001$	reject	$Jn = 4.209$, $p < 0.001$	reject	$Jn = 3.332$, $p < 0.001$	reject
CD 1WWI	$RESET(8, 2382) = 3.415$, $p < 0.001$	reject	$Jn = 1.086$, $p = 0.093$	reject	$Jn = -0.800$, $p = 0.998$	accept
CD 1WWT	$RESET(8, 2382) = 3.125$, $p < 0.001$	reject	$Jn = 4.292$, $pp < 0.001$	reject	$Jn = 0.902$, $p = 0.015$	reject
CD 2WW	$RESET(8, 2382) = 3.230$, $p < 0.001$	reject	$Jn = 0.905$, $p = 0.060$	reject	$Jn = -0.805$, $p = 1$	accept
CD FD	$RESET(8, 2039) = 1.167$, $p = 0.316$	accept	$Jn = 3.794$, $p < 0.001$	reject	$Jn = -0.708$, $p = 0.584$	accept
TL pooled	$RESET(28, 2351) = 1.336$, $p = 0.112$	accept	$Jn = 2.854$, $p < 0.001$	reject	$Jn = 1.107$, $p < 0.001$	reject
TL 1WWI	$RESET(28, 2352) = 1.326$, $p = 0.118$	accept	$Jn = 1.160$, $p = 0.080$	reject	$Jn = 0.553$, $p = 0.243$	accept
TL 1WWT	$RESET(28, 2352) = 1.838$, $p < 0.001$	reject	$Jn = 3.044$, $p < 0.001$	reject	$Jn = 1.279$, $p = 0.018$	reject
TL 2WW	$RESET(28, 2352) = 1.334$, $p = 0.113$	accept	$Jn = -0.006$, $p = 0.301$	accept	$Jn = 0.373$, $p = 0.185$	accept
TL FD	$RESET(28, 2009) = 1.733$, $p = 0.010$	reject	$Jn = 4.391$, $p < 0.001$	reject	$Jn = 1.000$, $p = 0.351$	accept

Note: The abbreviations of the model specifications are explained below table 1.

B. Kernel functions for ordered and unordered factor variables

B.1. Kernel functions for unordered factor variables

- Aitchison and Aitken (1976) kernel:

$$l(x_i, x, \lambda) = \begin{cases} 1 - \lambda, & \text{if } x_i = x \\ \lambda/(c-1), & \text{if } x_i \neq x \end{cases} \tag{10}$$

  where c is the number of levels of the unordered categorical variable x and smoothing parameter λ must lie between 0 and $(c-1)/c$.

  Weight of the other levels relative to the current level:

$$\frac{l(x_i, x \neq x_i, \lambda)}{l(x_i, x = x_i, \lambda)} = \frac{\lambda/(c-1)}{1-\lambda} = \frac{\lambda}{(c-1)\cdot(1-\lambda)} \tag{11}$$

- Li and Racine (2004) kernel:

$$l(x_i, x, \lambda) = \begin{cases} 1, & \text{if } x_i = x \\ \lambda, & \text{if } x_i \neq x \end{cases} \tag{12}$$

  where smoothing parameter λ must lie between 0 and 1.

  Weight of the other levels relative to the current level:

$$\frac{l(x_i, x \neq x_i, \lambda)}{l(x_i, x = x_i, \lambda)} = \frac{\lambda}{1} = \lambda \tag{13}$$

- Relationship between the Aitchison and Aitken (1976) kernel and the Li and Racine (2004) kernel: both kernel functions result in identical relative weights and hence, identical estimation results, if:

$$\lambda_{LRU} = \frac{\lambda_{AA}}{(c-1)\cdot(1-\lambda_{AA})} \tag{14}$$

  and consequently

$$\lambda_{AA} = \frac{\lambda_{LRU}\cdot(c-1)}{1+\lambda_{LRU}\cdot(c-1)}, \tag{15}$$

  where λ_{AA} is the bandwidth for the Aitchison and Aitken (1976) kernel, λ_{LRU} is the bandwidth for the Li and Racine (2004) kernel for unordered categorical variables, and c is the number of levels of the unordered categorical variable x.

B.2. Kernel functions for ordered factor variables

- Wang and van Ryzin (1981) kernel:

$$l(x_i, x, \lambda) = \begin{cases} 1 - \lambda, & \text{if } |x_i - x| = 0 \\ \frac{(1-\lambda)}{2} \lambda^{|x_i - x|}, & \text{if } |x_i - x| > 1 \end{cases} \tag{16}$$

where smoothing parameter λ must lie between 0 and 1.

Weight of the other levels relative to the current level:

$$\frac{l(x_i, x \neq x_i, \lambda)}{l(x_i, x = x_i, \lambda)} = \frac{\frac{(1-\lambda)}{2} \lambda^{|x_i - x|}}{1 - \lambda} = \frac{\lambda^{|x_i - x|}}{2} \tag{17}$$

- Racine and Li (2004) kernel:

$$l(x_i, x, \lambda) = \begin{cases} 1, & \text{if } |x_i - x| = 0 \\ \lambda^{|x_i - x|}, & \text{if } |x_i - x| > 1 \end{cases} \tag{18}$$

where smoothing parameter λ must lie between 0 and 1.

Weight of the other levels relative to the current level:

$$\frac{l(x_i, x \neq x_i, \lambda)}{l(x_i, x = x_i, \lambda)} = \frac{\lambda^{|x_i - x|}}{1} = \lambda^{|x_i - x|} \tag{19}$$

- Relationship between the Wang and van Ryzin (1981) kernel and the Li and Racine (2004) kernel: both kernel functions result in identical relative weights and hence, identical estimation results, if:

$$\lambda_{LRO} = \left(\frac{\lambda_{WVR}^{|x_i - x|}}{2} \right)^{\frac{1}{|x_i - x|}} = \frac{\lambda_{WVR}}{2^{\frac{1}{|x_i - x|}}} = 0.5^{\frac{1}{|x_i - x|}} \lambda_{WVR} \tag{20}$$

and consequently

$$\lambda_{WVR} = \lambda_{LRO} \, 2^{\frac{1}{|x_i - x|}}, \tag{21}$$

where λ_{WVR} is the bandwidth for the Wang and van Ryzin (1981) kernel and λ_{LRO} is the bandwidth for the Li and Racine (2004) kernel for ordered categorical variables. Hence, the Wang and van Ryzin (1981) kernel and the Li and Racine (2004) kernel for ordered categorical variables can only result in the same relative weights if x has only two levels (so that $|x_i - x|$ only has one possible value for $x \neq x_i$) and $\lambda_{LRO} \leq 0.5^{\frac{1}{|x_i - x|}}$. where $|x_i - x| \geq 1$. As an ordered categorical variable with just two levels is not different from unordered categorical variable with two levels, the Wang and van Ryzin (1981) kernel is practically never equivalent to the Li and Racine (2004) kernel for ordered categorical variables.

C. R code

C.1. Preparing the variables

```
# load add-on packages
library( "plm" )
library( "lmtest" )
library( "np" )
# load the data
load("npPanel_est.RData")
# calculate additional variables for Translog function (squares and interactions)
PLF_TF13$lL_sq <- 0.5 * PLF_TF13$lL * PLF_TF13$lL
PLF_TF13$lL_lA <- PLF_TF13$lL * PLF_TF13$lA
PLF_TF13$lL_lV <- PLF_TF13$lL * PLF_TF13$lV
PLF_TF13$lL_lK <- PLF_TF13$lL * PLF_TF13$lK
PLF_TF13$lA_sq <- 0.5 * PLF_TF13$lA * PLF_TF13$lA
PLF_TF13$lA_lV <- PLF_TF13$lA * PLF_TF13$lV
PLF_TF13$lA_lK <- PLF_TF13$lA * PLF_TF13$lK
PLF_TF13$lV_sq <- 0.5 * PLF_TF13$lV * PLF_TF13$lV
PLF_TF13$lV_lK <- PLF_TF13$lV * PLF_TF13$lK
PLF_TF13$lK_sq <- 0.5 * PLF_TF13$lK * PLF_TF13$lK
# create the panel data frame PLF_TF13_pd
PLF_TF13_pd <- pdata.frame(PLF_TF13, index = c("ID", "T"))
# add within (individual effects) transformed regression variables to PLF_TF13
PLF_TF13$lY_w <- Within(PLF_TF13_pd$lY, effect = "individual")
PLF_TF13$lL_w <- Within(PLF_TF13_pd$lL, effect = "individual")
PLF_TF13$lA_w <- Within(PLF_TF13_pd$lA, effect = "individual")
PLF_TF13$lV_w <- Within(PLF_TF13_pd$lV, effect = "individual")
PLF_TF13$lK_w <- Within(PLF_TF13_pd$lK, effect = "individual")
PLF_TF13$lL_sq_w <- Within(PLF_TF13_pd$lL_sq, effect = ("individual"))
PLF_TF13$lL_lA_w <- Within(PLF_TF13_pd$lL_lA, effect = "individual")
PLF_TF13$lL_lV_w <- Within(PLF_TF13_pd$lL_lV, effect = "individual")
PLF_TF13$lL_lK_w <- Within(PLF_TF13_pd$lL_lK, effect = "individual")
PLF_TF13$lA_sq_w <- Within(PLF_TF13_pd$lA_sq, effect = "individual")
PLF_TF13$lA_lV_w <- Within(PLF_TF13_pd$lA_lV, effect = "individual")
PLF_TF13$lA_lK_w <- Within(PLF_TF13_pd$lA_lK, effect = "individual")
PLF_TF13$lV_sq_w <- Within(PLF_TF13_pd$lV_sq, effect = "individual")
PLF_TF13$lV_lK_w <- Within(PLF_TF13_pd$lV_lK, effect = "individual")
PLF_TF13$lK_sq_w <- Within(PLF_TF13_pd$lK_sq, effect = "individual")
# add within (time effects) transformed regression variables to PLF_TF13
PLF_TF13$lY_w_t <- Within(PLF_TF13_pd$lY, effect = ("time"))
PLF_TF13$lL_w_t <- Within(PLF_TF13_pd$lL, effect = ("time"))
PLF_TF13$lA_w_t <- Within(PLF_TF13_pd$lA, effect = "time")
PLF_TF13$lV_w_t <- Within(PLF_TF13_pd$lV, effect = "time")
PLF_TF13$lK_w_t <- Within(PLF_TF13_pd$lK, effect = "time")
PLF_TF13$lL_sq_w_t <- Within(PLF_TF13_pd$lL_sq, effect = ("time"))
PLF_TF13$lL_lA_w_t <- Within(PLF_TF13_pd$lL_lA, effect = "time")
```

```
PLF_TF13$lL_lV_w_t <- Within(PLF_TF13_pd$lL_lV, effect = "time")
PLF_TF13$lL_lK_w_t <- Within(PLF_TF13_pd$lL_lK, effect = "time")
PLF_TF13$lA_sq_w_t <- Within(PLF_TF13_pd$lA_sq, effect = "time")
PLF_TF13$lA_lV_w_t <- Within(PLF_TF13_pd$lA_lV, effect = "time")
PLF_TF13$lA_lK_w_t <- Within(PLF_TF13_pd$lA_lK, effect = "time")
PLF_TF13$lV_sq_w_t <- Within(PLF_TF13_pd$lV_sq, effect = "time")
PLF_TF13$lV_lK_w_t <- Within(PLF_TF13_pd$lV_lK, effect = "time")
PLF_TF13$lK_sq_w_t <- Within(PLF_TF13_pd$lK_sq, effect = "time")
# add within (two-ways (individual and time) effects) transformed regression
# variables to PLF_TF13
PLF_TF13$lY_w_tw <- Within(Within(PLF_TF13_pd$lY, effect = "individual"),
                           effect = "time")
PLF_TF13$lL_w_tw <- Within(Within(PLF_TF13_pd$lL, effect = "individual"),
                           effect = "time")
PLF_TF13$lA_w_tw <- Within(Within(PLF_TF13_pd$lA, effect = "individual"),
                           effect = "time")
PLF_TF13$lV_w_tw <- Within(Within(PLF_TF13_pd$lV, effect = "individual"),
                           effect = "time")
PLF_TF13$lK_w_tw <- Within(Within(PLF_TF13_pd$lK, effect = "individual"),
                           effect = "time")
PLF_TF13$lL_sq_w_tw <- Within(Within(PLF_TF13_pd$lL_sq, effect = "individual"),
                              effect = "time")
PLF_TF13$lL_lA_w_tw <- Within(Within(PLF_TF13_pd$lL_lA, effect = "individual"),
                              effect = "time")
PLF_TF13$lL_lV_w_tw <- Within(Within(PLF_TF13_pd$lL_lV, effect = "individual"),
                              effect = "time")
PLF_TF13$lL_lK_w_tw <- Within(Within(PLF_TF13_pd$lL_lK, effect = "individual"),
                              effect = "time")
PLF_TF13$lA_sq_w_tw <- Within(Within(PLF_TF13_pd$lA_sq, effect = "individual"),
                              effect = "time")
PLF_TF13$lA_lV_w_tw <- Within(Within(PLF_TF13_pd$lA_lV, effect = "individual"),
                              effect = "time")
PLF_TF13$lA_lK_w_tw <- Within(Within(PLF_TF13_pd$lA_lK, effect = "individual"),
                              effect = "time")
PLF_TF13$lV_sq_w_tw <- Within(Within(PLF_TF13_pd$lV_sq, effect = "individual"),
                              effect = "time")
PLF_TF13$lV_lK_w_tw <- Within(Within(PLF_TF13_pd$lV_lK, effect = "individual"),
                              effect = "time")
PLF_TF13$lK_sq_w_tw <- Within(Within(PLF_TF13_pd$lK_sq, effect = "individual"),
                              effect = "time")
# add first difference transformed regression variables
PLF_TF13$lY_fd <- diff(PLF_TF13_pd$lY)
PLF_TF13$lL_fd <- diff(PLF_TF13_pd$lL)
PLF_TF13$lA_fd <- diff(PLF_TF13_pd$lA)
PLF_TF13$lV_fd <- diff(PLF_TF13_pd$lV)
PLF_TF13$lK_fd <- diff(PLF_TF13_pd$lK)
PLF_TF13$lL_sq_fd <- diff(PLF_TF13_pd$lL_sq)
```

```
PLF_TF13$lL_lA_fd <- diff(PLF_TF13_pd$lL_lA)
PLF_TF13$lL_lV_fd <- diff(PLF_TF13_pd$lL_lV)
PLF_TF13$lL_lK_fd <- diff(PLF_TF13_pd$lL_lK)
PLF_TF13$lA_sq_fd <- diff(PLF_TF13_pd$lA_sq)
PLF_TF13$lA_lV_fd <- diff(PLF_TF13_pd$lA_lV)
PLF_TF13$lA_lK_fd <- diff(PLF_TF13_pd$lA_lK)
PLF_TF13$lV_sq_fd <- diff(PLF_TF13_pd$lV_sq)
PLF_TF13$lV_lK_fd <- diff(PLF_TF13_pd$lV_lK)
PLF_TF13$lK_sq_fd <- diff(PLF_TF13_pd$lK_sq)
# update the panel data frame PLF_TF13_pd
PLF_TF13_pd <- pdata.frame(PLF_TF13, index = c("ID", "T"))
```

C.2. Estimation of parametric models

```
##################### Estimation of parametric models ##########################
########################   Cobb-Douglas models   ###############################
#### Create the general formula of Cobb-Douglas functional form
cdFormula <- log(Y) ~ log(L) + log(A) + log(V) + log(K)
### Estimation of varying coefficient models
## with individual effects
cd_vc_i <- pvcm( cdFormula, data = PLF_TF13_pd,
                 model = "within", effect = "individual" )
summary( cd_vc_i )
## with time effects
cd_vc_t <- pvcm( cdFormula, data = PLF_TF13_pd,
                 model = "within", effect = "time" )
summary( cd_vc_t )
### Estimate first difference models
## with individual effects
cd_fd <- plm( cdFormula, data = PLF_TF13_pd,
              model = "fd", effect = "individual" )
summary( cd_fd )
# Estimation of the same model with lm() (necessary to conduct resettest()
# and npcmstest())
cd_fd_lm <- lm( lY_fd ~ lL_fd + lA_fd + lV_fd + lK_fd, data = PLF_TF13,
                x = TRUE, y = TRUE )
summary( cd_fd_lm )
# Check if the coefficients of the models estimated with plm() and lm()
# are equal
all.equal( coef( cd_fd ), coef( cd_fd_lm ), check.attributes = FALSE )

### Estimation of within (fixed effects) models
## with individual effects
cd_w_i <- plm( cdFormula, data = PLF_TF13_pd,
               model = "within", effect = "individual" )
summary( cd_w_i )
# Estimation of the same model with lm() (necessary to conduct resettest()
```

```
# and npcmstest())
cd_w_i_lm <- lm( lY_w ~ 0 + lL_w + lA_w + lV_w + lK_w, data = PLF_TF13,
                 x = TRUE, y = TRUE )
summary( cd_w_i_lm )
# Check if the coefficients of the models estimated with plm() and lm()
# are equal
all.equal( coef( cd_w_i ), coef( cd_w_i_lm ), check.attributes = FALSE )
## with time effects
cd_w_t <- plm( cdFormula, data = PLF_TF13_pd,
               model = "within", effect = "time" )
summary( cd_w_t )
# Estimation of the same model with lm() (necessary to conduct resettest()
# and npcmstest())
cd_w_t_lm <- lm( lY_w_t ~ 0 + lL_w_t + lA_w_t + lV_w_t + lK_w_t,
                 data = PLF_TF13, x = TRUE, y = TRUE ) )
summary( cd_w_t_lm )
# Check if the coefficients of the models estimated with plm() and lm()
# are equal
all.equal( coef( cd_w_t ), coef( cd_w_t_lm ), check.attributes = FALSE )
## with two-ways effects
cd_w_it <- plm( cdFormula, data = PLF_TF13_pd,
                model = "within", effect = "twoways" )
summary( cd_w_it )
# Estimation of the same model with lm() (necessary to conduct resettest()
# and npcmstest())
cd_w_it_lm <- lm( lY_w_tw ~ 0 + lL_w_tw + lA_w_tw + lV_w_tw + lK_w_tw,
                  data = PLF_TF13,  x = TRUE, y = TRUE )
summary( cd_w_it_lm)
# Check if the coefficients of the models estimated with plm() and lm()
# are equal
all.equal( coef( cd_w_it ), coef( cd_w_it_lm ), check.attributes = FALSE )
### Estimation of random effects models
## with individual effects
cd_r_i <- plm( cdFormula, data = PLF_TF13_pd,
               model = "random", effect = "individual",
               random.method = "amemiya" )
summary( cd_r_i )
## with time effects
cd_r_t <- plm( cdFormula, data = PLF_TF13_pd,
               model = "random", effect = "time", random.method = "amemiya" )
summary( cd_r_t )
# two-ways effects
cd_r_it <- plm( cdFormula, data = PLF_TF13_pd,
                model = "random", effect = "twoways", random.method = "amemiya" )
summary( cd_r_it )
### Estimation of pooled model
cd_pool <- plm( cdFormula, PLF_TF13_pd, model = "pooling" )
```

```
summary( cd_pool )
# Estimation of the same model with lm() (necessary to conduct npcmstest())
cd_pool_lm <-  lm( cdFormula, data = PLF_TF13, x = TRUE, y = TRUE )
# Check if the coefficients of the models estimated with plm() and lm()
# are equal
all.equal( coef( cd_pool ), coef( cd_pool_lm ), check.attributes = FALSE )


############################   Translog models   ###############################
#### Create the general formula of Translog functional form
tlFormula <- log(Y) ~ log(L) + log(A) + log(V) + log(K) +
  I(0.5*log(L)^2) + I(0.5*log(A)^2) + I(0.5*log(V)^2) + I(0.5*log(K)^2) +
  I(log(L)*log(A)) + I(log(L)*log(V)) + I(log(L)*log(K)) +
  I(log(A)*log(V)) + I(log(A)*log(K)) + I(log(V)*log(K))
# Function tlEla for calculating partial output elasticities at mean values
# of regression variables
tlEla <- function( object ) {
  tlCoef <- coef( object )
  mDat <- colMeans( PLF_TF13_pd[ , c( "L", "A", "V", "K" ) ] )
  result <- numeric(4)
  result[1] <- tlCoef["log(L)"] +
    tlCoef["I(0.5␣*␣log(L)^2)"] * log( mDat["L"] ) +
    tlCoef["I(log(L)␣*␣log(A))"] * log( mDat["A"] ) +
    tlCoef["I(log(L)␣*␣log(V))"] * log( mDat["V"] ) +
    tlCoef["I(log(L)␣*␣log(K))"] * log( mDat["K"] )
  result[2] <- tlCoef["log(A)"] +
    tlCoef["I(log(L)␣*␣log(A))"] * log( mDat["L"] ) +
    tlCoef["I(0.5␣*␣log(A)^2)"] * log( mDat["A"] ) +
    tlCoef["I(log(A)␣*␣log(V))"] * log( mDat["V"] ) +
    tlCoef["I(log(A)␣*␣log(K))"] * log( mDat["K"] )
  result[3] <- tlCoef["log(V)"] +
    tlCoef["I(log(L)␣*␣log(V))"] * log( mDat["L"] ) +
    tlCoef["I(log(A)␣*␣log(V))"] * log( mDat["A"] ) +
    tlCoef["I(0.5␣*␣log(V)^2)"] * log( mDat["V"] ) +
    tlCoef["I(log(V)␣*␣log(K))"] * log( mDat["K"] )
  result[4] <- tlCoef["log(K)"] +
    tlCoef["I(log(L)␣*␣log(K))"] * log( mDat["L"] ) +
    tlCoef["I(log(A)␣*␣log(K))"] * log( mDat["A"] ) +
    tlCoef["I(log(V)␣*␣log(K))"] * log( mDat["V"] ) +
    tlCoef["I(0.5␣*␣log(K)^2)"] * log( mDat["K"] )
  return( result )
}

### Estimation of varying coefficient model with time effects
tl_vc_t <- pvcm( tlFormula, data = PLF_TF13_pd,
                 model = "within", effect = "time" )
summary( tl_vc_t )
```

```
### Estimation of first difference models
## with individual effects
tl_fd <- plm( tlFormula, data = PLF_TF13_pd,
              model = "fd", effect = "individual" )
summary( tl_fd )
# Print partial output elasticities
tlEla( tl_fd )
# Estimation of the same model with lm() (necessary to conduct resettest()
# and npcmstest() )
tl_fd_lm <- lm( lY_fd ~ lL_fd + lA_fd + lV_fd + lK_fd +
                lL_sq_fd + lA_sq_fd + lV_sq_fd + lK_sq_fd +
                lL_lA_fd + lL_lV_fd + lL_lK_fd + lA_lV_fd + lA_lK_fd + lV_lK_fd,
                data = PLF_TF13, x = TRUE, y = TRUE )
summary( tl_fd_lm)
# Check if the coefficients of the models estimated with plm() and lm()
# are equal
all.equal(coef( tl_fd ), coef( tl_fd_lm ), check.attributes = FALSE )
### Estimation of within (fixed effects) models
## with individual effects
tl_w_i <- plm( tlFormula, data = PLF_TF13_pd,
               model = "within", effect = "individual" )
summary( tl_w_i )
# Print partial output elasticities
tlEla( tl_w_i )
# Estimation of the same model with lm() (necessary to conduct resettest()
# and npcmstest() )
tl_w_i_lm <- lm( lY_w ~ 0 + lL_w + lA_w + lV_w + lK_w +
                   lL_sq_w + lA_sq_w + lV_sq_w + lK_sq_w +
                   lL_lA_w + lL_lV_w + lL_lK_w + lA_lV_w + lA_lK_w + lV_lK_w,
                   data = PLF_TF13, x = TRUE, y = TRUE )
summary( tl_w_i_lm )
# Check if the coefficients of the models estimated with plm() and lm()
# are equal
all.equal(coef( tl_w_i ), coef( tl_w_i_lm ), check.attributes = FALSE )
## with time effects
tl_w_t <- plm( tlFormula, data = PLF_TF13_pd,
               model = "within", effect = "time" )
summary( tl_w_t )
# Print partial output elasticities
tlEla( tl_w_t )
# Estimation of the same model with lm() (necessary to conduct resettest()
# and npcmstest() )
tl_w_t_lm <- lm( lY_w_t ~ 0 + lL_w_t + lA_w_t + lV_w_t + lK_w_t +
                   lL_sq_w_t + lA_sq_w_t + lV_sq_w_t + lK_sq_w_t +
                   lL_lA_w_t + lL_lV_w_t + lL_lK_w_t +
                   lA_lV_w_t + lA_lK_w_t + lV_lK_w_t,
                   data = PLF_TF13, x = TRUE, y = TRUE )
```

```
summary( tl_w_t_lm )
# Check if the coefficients of the models estimated with plm() and lm()
# are equal
all.equal(coef( tl_w_t ), coef( tl_w_t_lm ), check.attributes = FALSE )
## with two-ways effects
tl_w_it <- plm( tlFormula, data = PLF_TF13_pd,
                model = "within", effect = "twoways" )
summary( tl_w_it )
# Print partial output elasticities
tlEla( tl_w_it )
# Estimation of the same model with lm() (necessary to conduct resettest()
# and npcmstest() )
tl_w_it_lm <- lm( lY_w_tw ~ 0 + lL_w_tw + lA_w_tw + lV_w_tw + lK_w_tw +
                    lL_sq_w_tw + lA_sq_w_tw + lV_sq_w_tw + lK_sq_w_tw +
                    lL_lA_w_tw + lL_lV_w_tw + lL_lK_w_tw + lA_lV_w_tw +
                    lA_lK_w_tw + lV_lK_w_tw,
                  data = PLF_TF13, x = TRUE, y = TRUE )
summary( tl_w_it_lm )
# Check if the coefficients of the models estimated with plm() and lm()
# are equal
all.equal( coef( tl_w_it ), coef( tl_w_it_lm ), check.attributes = FALSE )
### Estimation of random effects models
## with individual effects
tl_r_i <- plm( tlFormula, data = PLF_TF13_pd,
               model = "random", effect = "individual",
               random.method = "amemiya" )
summary( tl_r_i )
# Print partial output elasticities
tlEla( tl_r_i )
# time effects
tl_r_t <- plm( tlFormula, data = PLF_TF13_pd,
               model = "random", effect = "time", random.method = "amemiya" )
summary( tl_r_t )
# Print partial output elasticities
tlEla( tl_r_t )
## with two-ways effects
tl_r_it <- plm( tlFormula, data = PLF_TF13_pd,
                model = "random", effect = "twoways", random.method = "amemiya" )
summary( tl_r_it )
# Print partial output elasticities
tlEla( tl_r_it )
### Estimation of pooled model
tl_pool <- plm( tlFormula, data = PLF_TF13_pd, model = "pooling" )
summary( tl_pool )
# Print partial output elasticities
tlEla( tl_pool )
# Estimation of the same model with lm() (necessary to conduct npcmstest() )
```

```
tl_pool_lm <-  lm( tlFormula, data = PLF_TF13, x = TRUE, y = TRUE )
# Check if the coefficients of the models estimated with plm() and lm()
# are equal
all.equal( coef( tl_pool ), coef( tl_pool_lm ), check.attributes = FALSE )
```

C.3. Parametric tests of parametric models

```
############   Testing significance of input quantities   #####################
#Create the Translog formula without Labor variable
tlFormulaNoL <- log(Y) ~ log(A) + log(V) + log(K) +
  I(0.5*log(A)^2) + I(0.5*log(V)^2) + I(0.5*log(K)^2) +
  I(log(A)*log(V)) + I(log(A)*log(K)) + I(log(V)*log(K))
#Create the Translog formula without Land variable
tlFormulaNoA <- log(Y) ~ log(L) + log(V) + log(K) +
  I(0.5*log(L)^2) + I(0.5*log(V)^2) + I(0.5*log(K)^2) +
  I(log(L)*log(V)) + I(log(L)*log(K)) + I(log(V)*log(K))
#Create the Translog formula without Intermediate inputs variable
tlFormulaNoV <- log(Y) ~ log(L) + log(A) + log(K) +
  I(0.5*log(L)^2) + I(0.5*log(A)^2) + I(0.5*log(K)^2) +
  I(log(L)*log(A)) + I(log(L)*log(K)) + I(log(A)*log(K))
#Create the Translog formula without Capital variable
tlFormulaNoK <- log(Y) ~ log(L) + log(A) + log(V) +
  I(0.5*log(L)^2) + I(0.5*log(A)^2) + I(0.5*log(V)^2) +
  I(log(L)*log(A)) + I(log(L)*log(V)) + I(log(A)*log(V))
#Wald test for significance of input quantities in pooled model
waldtest( tl_pool, tlFormulaNoL )
waldtest( tl_pool, tlFormulaNoA )
waldtest( tl_pool, tlFormulaNoV )
waldtest( tl_pool, tlFormulaNoK )
#Wald test for significance of input quantities in within model with individual
#effects
waldtest( tl_w_i, tlFormulaNoL )
waldtest( tl_w_i, tlFormulaNoA )
waldtest( tl_w_i, tlFormulaNoV )
waldtest( tl_w_i, tlFormulaNoK )
#Wald test for significance of input quantities in within model with time
#effects
waldtest( tl_w_t, tlFormulaNoL )
waldtest( tl_w_t, tlFormulaNoA )
waldtest( tl_w_t, tlFormulaNoV )
waldtest( tl_w_t, tlFormulaNoK )
#Wald test for significance of input quantities in random effects model with
#individual effects
waldtest( tl_r_i, tlFormulaNoL )
waldtest( tl_r_i, tlFormulaNoA )
waldtest( tl_r_i, tlFormulaNoV )
waldtest( tl_r_i, tlFormulaNoK )
```

```
waldtest( tl_r_t, tlFormulaNoL )
waldtest( tl_r_t, tlFormulaNoA )
waldtest( tl_r_t, tlFormulaNoV )
waldtest( tl_r_t, tlFormulaNoK )
#Wald test for significance of input quantities in first differenced model
waldtest( tl_fd, tlFormulaNoL )
waldtest( tl_fd, tlFormulaNoA )
waldtest( tl_fd, tlFormulaNoV )
waldtest( tl_fd, tlFormulaNoK )
#Wald test for significance of input quantities in within (fixed effects) model
#with twoways (individual and time) effects
waldtest( tl_w_it, tlFormulaNoL )
waldtest( tl_w_it, tlFormulaNoA )
waldtest( tl_w_it, tlFormulaNoV )
waldtest( tl_w_it, tlFormulaNoK )
#Wald test for significance of input quantities in random effects model
#with twoways (individual and time) effects
waldtest( tl_r_it, tlFormulaNoL )
waldtest( tl_r_it, tlFormulaNoA )
waldtest( tl_r_it, tlFormulaNoV )
waldtest( tl_r_it, tlFormulaNoK )

############  Testing individual and time effects   ###########################
####Calculate standard deviations of individual and time effects
### Cobb-Douglas
sd( fixef( cd_w_i ) )
sd( fixef( cd_w_t ) )
sqrt( cd_r_i$ercomp$sigma$id )
sqrt( cd_r_t$ercomp$sigma$id )
sd( fixef( cd_w_it ) )
sd( fixef( cd_w_it, effect = "time" ) )
sqrt( cd_r_it$ercomp$sigma$id )
sqrt( cd_r_it$ercomp$sigma$time )
### Translog
sd( fixef( tl_w_i ) )
sd( fixef( tl_w_t ) )
sqrt( tl_r_i$ercomp$sigma$id )
sqrt( tl_r_t$ercomp$sigma$id )
sd( fixef( tl_w_it ) )
sd( fixef( tl_w_it, effect = "time" ) )
sqrt( tl_r_it$ercomp$sigma$id )
sqrt( tl_r_it$ercomp$sigma$time )
############   Testing poolability   ###########################################
### Cobb-Douglas
# test equality of all individual or time effects
pooltest( cd_pool, cd_w_i )
pooltest( cd_pool, cd_w_t )
```

```
pooltest( cd_pool, cd_w_it )
pooltest( cd_w_i, cd_w_it )
pooltest( cd_w_t, cd_w_it )
# test equality of all parameters (incl. individual or time effects)
pooltest( cd_pool, cd_vc_i )
pooltest( cd_pool, cd_vc_t )
# test equality of all slope parameters
pooltest( cd_w_i, cd_vc_i )
pooltest( cd_w_t, cd_vc_t )
### Translog
# test equality of all individual or time effects
pooltest( tl_pool, tl_w_i )
pooltest( tl_pool, tl_w_t )
pooltest( tl_pool, tl_w_it )
pooltest( tl_w_i, tl_w_it )
pooltest( tl_w_t, tl_w_it )
# test equality of all parameters (incl. time effects)
pooltest( tl_pool, tl_vc_t )
# test equality of all slope parameters
pooltest( tl_w_t, tl_vc_t )

############   Breusch Pagan tests   ####################

### Cobb-Douglas
plmtest( cd_pool, effect = "individual", type = "bp" )
plmtest( cd_pool, effect = "time", type = "bp" )
plmtest( cd_pool, effect = "twoways", type = "bp" )

### Translog
plmtest( tl_pool, effect = "individual", type = "bp" )
plmtest( tl_pool, effect = "time", type = "bp" )
plmtest( tl_pool, effect = "twoways", type = "bp" )

############   Hausman tests   ####################
### Cobb-Douglas
phtest( cd_w_i, cd_r_i )
phtest( cd_w_t, cd_r_t )
phtest( cd_w_it, cd_r_it )

### Translog
phtest( tl_w_i, tl_r_i )
phtest( tl_w_t, tl_r_t )
phtest( tl_w_it, tl_r_it )

############  Testing serial correlation   ####################
### first-differences
pwfdtest( cd_fd )
```

```
pwfdtest( tl_fd )
### Breusch-Godfrey / Wooldridge test
pbgtest( cd_w_i )
pbgtest( tl_w_i )
### Wooldridge's test for "short" panels
pwartest( cd_w_i )
pwartest( tl_w_i )

############   Wald tests   ####################
##Test Cobb-Douglas vs. Translog
waldtest( tl_pool, cd_pool )
waldtest( tl_w_i, cd_w_i )
waldtest( tl_w_t, cd_w_t )
waldtest( tl_w_it, cd_w_it )
waldtest( tl_r_i, cd_r_i )
waldtest( tl_r_t, cd_r_t )
waldtest( tl_r_it, cd_r_it )
waldtest( tl_fd, cd_fd )

############   RESET tests   ####################
##Test the Cobb-Douglas specification
resettest( cd_pool , power = 2:3, type = "regressor" )
resettest( cd_w_i_lm , power = 2:3, type = "regressor" )
resettest( cd_w_t_lm , power = 2:3, type = "regressor" )
resettest( cd_w_it_lm , power = 2:3, type = "regressor" )
resettest( cd_fd_lm , power = 2:3, type = "regressor" )
##Test the Translog specification
resettest( tl_pool , power = 2:3, type = "regressor" )
resettest( tl_w_i_lm , power = 2:3, type = "regressor" )
resettest( tl_w_t_lm , power = 2:3, type = "regressor" )
resettest( tl_w_it_lm , power = 2:3, type = "regressor" )
resettest( tl_fd_lm , power = 2:3, type = "regressor" )
```

C.4. Nonparametric tests of parametric models

```
###########   Nonparametric Consistent Model Specifications Tests #############
# Cobb-Douglas
# Pooled model
# Nonparametric Consistent Model Specifications Test (local constant version)
npcmstest_cd_lc_pooled <- npcmstest( model = cd_pool_lm,
                                     xdat = model.frame(cd_pool_lm)[-1],
                                     ydat = model.frame(cd_pool_lm)[1],
                                     nmulti = 10,
                                     regtype = "lc",
                                     method = "cv.aic",
                                     type = "fixed",
                                     ckertype = "epanechnikov",
```

```
                                    ckerorder = 2,
                                    pckertype = "Second-Order␣Epanechnikov",
                                    boot.num = 399 )
summary( npcmstest_cd_lc_pooled)
# Nonparametric Consistent Model Specifications Test (local linear version)
npcmstest_cd_ll_pooled <- npcmstest( model = cd_pool_lm,
                                    xdat = model.frame(cd_pool_lm)[-1],
                                    ydat = model.frame(cd_pool_lm)[1],
                                    nmulti = 10,
                                    regtype = "lL",
                                    method = "cv.aic",
                                    type = "fixed",
                                    ckertype = "epanechnikov",
                                    ckerorder = 2,
                                    pckertype = "Second-Order␣Epanechnikov",
                                    boot.num = 399 )
summary( npcmstest_cd_ll_pooled)
# Within (fixed effects) individual effects
# Nonparametric Consistent Model Specifications Test (local constant version)
npcmstest_cd_lc_1WWI <- npcmstest( model = cd_w_i_lm,
                                  xdat = model.frame(cd_w_i_lm)[-1],
                                  ydat = model.frame(cd_w_i_lm)[1],
                                  nmulti = 10,
                                  regtype = "lc",
                                  method = "cv.aic",
                                  type = "fixed",
                                  ckertype = "epanechnikov",
                                  ckerorder = 2,
                                  pckertype = "Second-Order␣Epanechnikov",
                                  boot.num = 399 )
summary( npcmstest_cd_lc_1WWI )
# Nonparametric Consistent Model Specifications Test (local linear version)
npcmstest_cd_ll_1WWI <- npcmstest( model = cd_w_i_lm,
                                  xdat = model.frame(cd_w_i_lm)[-1],
                                  ydat = model.frame(cd_w_i_lm)[1],
                                  nmulti = 10,
                                  regtype = "ll",
                                  method = "cv.aic",
                                  type = "fixed",
                                  ckertype = "epanechnikov",
                                  ckerorder = 2,
                                  pckertype = "Second-Order␣Epanechnikov",
                                  boot.num = 399)
summary( npcmstest_cd_ll_1WWI )
# Within (fixed effects) time effects
# Nonparametric Consistent Model Specifications Test (local constant version)
# of pooled Cobb-Douglas parametric model
```

```
npcmstest_cd_lc_1WWT <- npcmstest( model = cd_w_t_lm,
                                  xdat = model.frame(cd_w_t_lm)[-1],
                                  ydat = model.frame(cd_w_t_lm)[1],
                                  nmulti = 10,
                                  regtype = "lc",
                                  method = "cv.aic",
                                  type = "fixed",
                                  ckertype = "epanechnikov",
                                  ckerorder = 2,
                                  pckertype = "Second-Order␣Epanechnikov",
                                  boot.num = 399 )
summary( npcmstest_cd_lc_1WWT )
# Nonparametric Consistent Model Specifications Test (local linear version)
npcmstest_cd_ll_1WWT <- npcmstest( model = cd_w_t_lm,
                                  xdat = model.frame(cd_w_t_lm)[-1],
                                  ydat = model.frame(cd_w_t_lm)[1],
                                  nmulti = 10,
                                  regtype = "ll",
                                  method = "cv.aic",
                                  type = "fixed",
                                  ckertype = "epanechnikov",
                                  ckerorder = 2,
                                  pckertype = "Second-Order␣Epanechnikov",
                                  boot.num = 399 )
summary(  npcmstest_cd_ll_1WWT )
# Within (fixed effects) twoways (individual and time) effects
# Nonparametric Consistent Model Specifications Test (local constant version)
npcmstest_cd_lc_2WW <- npcmstest( model = cd_w_it_lm,
                                 xdat = model.frame(cd_w_it_lm)[-1],
                                 ydat = model.frame(cd_w_it_lm)[1],
                                 nmulti = 10,
                                 regtype = "lc",
                                 method = "cv.aic",
                                 type = "fixed",
                                 ckertype = "epanechnikov",
                                 ckerorder = 2,
                                 pckertype = "Second-Order␣Epanechnikov",
                                 boot.num = 399 )
summary( npcmstest_cd_lc_2WW)
# Nonparametric Consistent Model Specifications Test (local linear version)
npcmstest_cd_ll_2WW <- npcmstest( model = cd_w_it_lm,
                                 xdat = model.frame(cd_w_it_lm)[-1],
                                 ydat = model.frame(cd_w_it_lm)[1],
                                 nmulti = 10,
                                 regtype = "ll",
                                 method = "cv.aic",
                                 type = "fixed",
```

```
                                ckertype = "epanechnikov",
                                ckerorder = 2,
                                pckertype = "Second-Order␣Epanechnikov",
                                boot.num = 399 )
summary( npcmstest_cd_ll_2WW)
# First difference model
# Estimation of the parametric model
# Nonparametric Consistent Model Specifications Test (local constant version)
npcmstest_cd_lc_FD <- npcmstest( model = cd_fd_lm,
                               xdat = model.frame(cd_fd_lm)[-1],
                               ydat = model.frame(cd_fd_lm)[1],
                               nmulti = 10,
                               regtype = "lc",
                               method = "cv.aic",
                               type = "fixed",
                               ckertype = "epanechnikov",
                               ckerorder = 2,
                               pckertype = "Second-Order␣Epanechnikov",
                               boot.num = 399 )
summary( npcmstest_cd_lc_FD)
# Nonparametric Consistent Model Specifications Test (local linear version)
npcmstest_cd_ll_FD <- npcmstest( model = cd_fd_lm,
                               xdat = model.frame(cd_fd_lm)[-1],
                               ydat = model.frame(cd_fd_lm)[1],
                               nmulti = 10,
                               regtype = "ll",
                               method = "cv.aic",
                               type = "fixed",
                               ckertype = "epanechnikov",
                               ckerorder = 2,
                               pckertype = "Second-Order␣Epanechnikov",
                               boot.num = 399 )
summary( npcmstest_cd_ll_FD )
# Translog
# Pooled model
# Nonparametric Consistent Model Specifications Test (local constant version)
npcmstest_tl_lc_pooled <- npcmstest( model = tl_pool_lm,
                                   xdat = model.frame(tl_pool_lm)[-1],
                                   ydat = model.frame(tl_pool_lm)[1],
                                   nmulti = 10,
                                   regtype = "lc",
                                   method = "cv.aic",
                                   type = "fixed",
                                   ckertype = "epanechnikov",
                                   ckerorder = 2,
                                   pckertype = "Second-Order␣Epanechnikov",
                                   boot.num = 399 )
```

```
summary( npcmstest_tl_lc_pooled )
# Nonparametric Consistent Model Specifications Test (local linear version)
npcmstest_tl_ll_pooled <- npcmstest( model = tl_pool_lm,
                                     xdat = model.frame(tl_pool_lm)[-1],
                                     ydat = model.frame(tl_pool_lm)[1],
                                     nmulti = 10,
                                     regtype = "ll",
                                     method = "cv.aic",
                                     type = "fixed",
                                     ckertype = "epanechnikov",
                                     ckerorder = 2,
                                     pckertype = "Second-Order␣Epanechnikov",
                                     boot.num = 399 )
summary( npcmstest_tl_lc_pooled )
# Within (fixed effects) individual effects
# Nonparametric Consistent Model Specifications Test (local constant version)
npcmstest_tl_lc_1WWI <- npcmstest( model = tl_w_i_lm,
                                   xdat = model.frame(tl_w_i_lm)[-1],
                                   ydat = model.frame(tl_w_i_lm)[1],
                                   nmulti = 10,
                                   regtype = "lc",
                                   method = "cv.aic",
                                   type = "fixed",
                                   ckertype = "epanechnikov",
                                   ckerorder = 2,
                                   pckertype = "Second-Order␣Epanechnikov",
                                   boot.num = 399 )
summary( npcmstest_tl_lc_1WWI)
# Nonparametric Consistent Model Specifications Test (local linear version)
npcmstest_tl_ll_1WWI <- npcmstest( model = tl_w_i_lm,
                                   xdat = model.frame(tl_w_i_lm)[-1],
                                   ydat = model.frame(tl_w_i_lm)[1],
                                   nmulti = 10,
                                   regtype = "ll",
                                   method = "cv.aic",
                                   type = "fixed",
                                   ckertype = "epanechnikov",
                                   ckerorder = 2,
                                   pckertype = "Second-Order␣Epanechnikov",
                                   boot.num = 399 )
summary( npcmstest_tl_ll_1WWI)
# Within (fixed effects) time effects
# Nonparametric Consistent Model Specifications Test (local constant version)
npcmstest_tl_lc_1WWT <- npcmstest( model = tl_w_t_lm,
                                   xdat = model.frame(tl_w_t_lm)[-1],
                                   ydat = model.frame(tl_w_t_lm)[1],
                                   nmulti = 10,
```

```
                                    regtype = "lc",
                                    method = "cv.aic",
                                    type = "fixed",
                                    ckertype = "epanechnikov",
                                    ckerorder = 2,
                                    pckertype = "Second-Order␣Epanechnikov",
                                    boot.num = 399 )
summary( npcmstest_tl_lc_1WWT )
# Nonparametric Consistent Model Specifications Test (local linear version)
npcmstest_tl_ll_1WWT <- npcmstest( model = tl_w_t_lm,
                                    xdat = model.frame(tl_w_t_lm)[-1],
                                    ydat = model.frame(tl_w_t_lm)[1],
                                    nmulti = 10,
                                    regtype = "ll",
                                    method = "cv.aic",
                                    type = "fixed",
                                    ckertype = "epanechnikov",
                                    ckerorder = 2,
                                    pckertype = "Second-Order␣Epanechnikov",
                                    boot.num = 399 )
summary( npcmstest_tl_ll_1WWT)
# Within (fixed effects) twoways (individual and time) effects
# Nonparametric Consistent Model Specifications Test (local constant version)
npcmstest_tl_lc_2WW <- npcmstest( model = tl_w_it_lm,
                                  xdat = model.frame(tl_w_it_lm)[-1],
                                  ydat = model.frame(tl_w_it_lm)[1],
                                  nmulti = 10,
                                  regtype = "lc",
                                  method = "cv.aic",
                                  type = "fixed",
                                  ckertype = "epanechnikov",
                                  ckerorder = 2,
                                  pckertype = "Second-Order␣Epanechnikov",
                                  boot.num = 399 )
summary( npcmstest_tl_lc_2WW )
# Nonparametric Consistent Model Specifications Test (local linear version)
npcmstest_tl_ll_2WW <- npcmstest( model = tl_w_it_lm,
                                  xdat = model.frame(tl_w_it_lm)[-1],
                                  ydat = model.frame(tl_w_it_lm)[1],
                                  nmulti = 10,
                                  regtype = "ll",
                                  method = "cv.aic",
                                  type = "fixed",
                                  ckertype = "epanechnikov",
                                  ckerorder = 2,
                                  pckertype = "Second-Order␣Epanechnikov",
                                  boot.num = 399 )
```

```
summary( npcmstest_tl_ll_2WW )
# First difference model
# Nonparametric Consistent Model Specifications Test (local constant version)
npcmstest_tl_lc_FD <- npcmstest( model = tl_fd_lm,
                                 xdat = model.frame(tl_fd_lm)[-1],
                                 ydat = model.frame(tl_fd_lm)[1],
                                 nmulti = 10,
                                 regtype = "lc",
                                 method = "cv.aic",
                                 type = "fixed",
                                 ckertype = "epanechnikov",
                                 ckerorder = 2,
                                 pckertype = "Second-Order␣Epanechnikov",
                                 boot.num = 399 )
summary( npcmstest_tl_lc_FD )
# Nonparametric Consistent Model Specifications Test (local linear version)
npcmstest_tl_ll_FD <- npcmstest( model = tl_fd_lm,
                                 xdat = model.frame(tl_fd_lm)[-1],
                                 ydat = model.frame(tl_fd_lm)[1],
                                 nmulti = 10,
                                 regtype = "lc",
                                 method = "cv.aic",
                                 type = "fixed",
                                 ckertype = "epanechnikov",
                                ckerorder = 2,
                                pckertype = "Second-Order␣Epanechnikov",
                                boot.num = 399 )
summary( npcmstest_tl_ll_FD )
```

C.5. Estimation of nonparametric models

```
#model.np.e.lr.of.PLF_TF13
bw.e.lr.of.PLF_TF13 <- npregbw( lY ~ lL + lA + lV + lK + T_of +ID,
                               regtype = "ll",
                               bwmethod = "cv.aic",
                               ckertype = "epanechnikov",
                               ukertype = "liracine",
                               okertype = "liracine",
                               nmulti = 10,
                               data = PLF_TF13 )
summary( bw.e.lr.of.PLF_TF13 )
model.np.e.lr.of.PLF_TF13 <- npreg( bws = bw.e.lr.of.PLF_TF13,
                                   data = PLF_TF13,
                                   gradients = TRUE,
                                   residuals = TRUE )
summary( model.np.e.lr.of.PLF_TF13 )
sigtest.model.np.e.lr.of.PLF_TF13 <- npsigtest( model.np.e.lr.of.PLF_TF13,
```

```
                                                  boot.num = 399 )
summary( sigtest.model.np.e.lr.of.PLF_TF13 )
summary( model.np.e.lr.of.PLF_TF13$grad )
# model.np.e.lr.wvr.of.PLF_TF13
bw.e.lr.wvr.of.PLF_TF13 <- npregbw( lY ~ lL + lA + lV + lK + T_of +ID,
                                   regtype = "ll",
                                   bwmethod = "cv.aic",
                                   ckertype = "epanechnikov",
                                   ukertype = "liracine",
                                   okertype = "wangvanryzin",
                                   nmulti = 10,
                                   data = PLF_TF13 )
summary( bw.e.lr.wvr.of.PLF_TF13 )
model.np.e.lr.wvr.of.PLF_TF13 <- npreg( bws = bw.e.lr.wvr.of.PLF_TF13,
                                       data = PLF_TF13,
                                       gradients = TRUE,
                                       residuals = TRUE )
summary( model.np.e.lr.wvr.of.PLF_TF13 )
sigtest.model.np.e.lr.wvr.of.PLF_TF13 <- npsigtest( model.np.e.lr.wvr.of.PLF_TF13,
                                                    boot.num = 399 )
summary( sigtest.model.np.e.lr.wvr.of.PLF_TF13 )
summary( model.np.e.lr.wvr.of.PLF_TF13$grad )
# model.np.g.lr.of.PLF_TF13
bw.g.lr.of.PLF_TF13 <- npregbw( lY ~ lL + lA + lV + lK + T_of +ID,
                                regtype = "ll",
                                bwmethod = "cv.aic",
                                ckertype = "gaussian",
                                ukertype = "liracine",
                                okertype = "liracine",
                                nmulti = 10,
                                data = PLF_TF13 )
summary( bw.g.lr.of.PLF_TF13 )
model.np.g.lr.of.PLF_TF13 <- npreg( bws = bw.g.lr.of.PLF_TF13,
                                   data = PLF_TF13,
                                   gradients = TRUE,
                                   residuals = TRUE )
summary( model.np.g.lr.of.PLF_TF13 )
sigtest.model.np.g.lr.of.PLF_TF13 <- npsigtest( model.np.g.lr.of.PLF_TF13,
                                                boot.num = 399 )
summary( sigtest.model.np.g.lr.of.PLF_TF13 )
summary( model.np.g.lr.of.PLF_TF13$grad )
# model.np.g.lr.wvr.of.PLF_TF13
bw.g.lr.wvr.of.PLF_TF13 <- npregbw( lY ~ lL + lA + lV + lK + T_of +ID,
                                   regtype = "ll",
                                   bwmethod = "cv.aic",
                                   ckertype = "gaussian",
                                   ukertype = "liracine",
```

```
                                   okertype = "wangvanryzin",
                                   nmulti = 10,
                                   data = PLF_TF13 )
summary( bw.g.lr.wvr.of.PLF_TF13 )
model.np.g.lr.wvr.of.PLF_TF13 <- npreg( bws = bw.g.lr.wvr.of.PLF_TF13,
                                       data = PLF_TF13,
                                       gradients = TRUE,
                                       residuals = TRUE )
summary( model.np.g.lr.wvr.of.PLF_TF13 )
sigtest.model.np.g.lr.wvr.of.PLF_TF13 <- npsigtest( model.np.g.lr.wvr.of.PLF_TF13,
                                                   boot.num = 399 )
summary( sigtest.model.np.g.lr.wvr.of.PLF_TF13 )
summary( model.np.g.lr.wvr.of.PLF_TF13$grad )
# model.np.e.aa.lr.of.PLF_TF13
bw.e.aa.lr.of.PLF_TF13 <- npregbw( lY ~ lL + lA + lV + lK + T_of +ID,
                                  regtype = "ll",
                                  bwmethod = "cv.aic",
                                  ckertype = "epanechnikov",
                                  ukertype = "aitchisonaitken",
                                  okertype = "liracine",
                                  nmulti = 10,
                                  data = PLF_TF13 )
summary( bw.e.aa.lr.of.PLF_TF13 )
model.np.e.aa.lr.of.PLF_TF13 <- npreg( bws = bw.e.aa.lr.of.PLF_TF13,
                                      data = PLF_TF13,
                                      gradients = TRUE,
                                      residuals = TRUE )
summary( model.np.e.aa.lr.of.PLF_TF13 )
sigtest.model.np.e.aa.lr.of.PLF_TF13 <- npsigtest( model.np.e.aa.lr.of.PLF_TF13,
                                                  boot.num = 399 )
summary( sigtest.model.np.e.aa.lr.of.PLF_TF13 )
summary( model.np.e.aa.lr.of.PLF_TF13$grad)
# model.np.e.aa.wvr.of.PLF_TF13
bw.e.aa.wvr.of.PLF_TF13 <- npregbw( lY ~ lL + lA + lV + lK + T_of +ID,
                                   regtype = "ll",
                                   bwmethod = "cv.aic",
                                   ckertype = "epanechnikov",
                                   ukertype = "aitchisonaitken",
                                   okertype = "wangvanryzin",
                                   nmulti = 10,
                                   data = PLF_TF13 )
summary( bw.e.aa.wvr.of.PLF_TF13 )
model.np.e.aa.wvr.of.PLF_TF13 <- npreg( bws = bw.e.aa.wvr.of.PLF_TF13,
                                       data = PLF_TF13,
                                       gradients = TRUE,
                                       residuals = TRUE )
summary( model.np.e.aa.wvr.of.PLF_TF13 )
```

```
sigtest.model.np.e.aa.wvr.of.PLF_TF13 <- npsigtest( model.np.e.aa.wvr.of.PLF_TF13,
                                                    boot.num = 399 )
summary( sigtest.model.np.e.aa.wvr.of.PLF_TF13 )
summary( model.np.e.aa.wvr.of.PLF_TF13$grad )
# model.np.g.aa.wvr.of.PLF_TF13
bw.g.aa.wvr.of.PLF_TF13 <- npregbw( lY ~ lL + lA + lV + lK + T_of +ID,
                                   regtype = "ll",
                                   bwmethod = "cv.aic",
                                   ckertype = "gaussian",
                                   ukertype = "aitchisonaitken",
                                   okertype = "wangvanryzin",
                                   nmulti = 10,
                                   data = PLF_TF13 )
summary( bw.g.aa.wvr.of.PLF_TF13 )
model.np.g.aa.wvr.of.PLF_TF13 <- npreg( bws = bw.g.aa.wvr.of.PLF_TF13,
                                        data = PLF_TF13,
                                        gradients = TRUE,
                                        residuals = TRUE )
summary( model.np.g.aa.wvr.of.PLF_TF13 )
sigtest.model.np.g.aa.wvr.of.PLF_TF13 <- npsigtest( model.np.g.aa.wvr.of.PLF_TF13,
                                                    boot.num = 399 )
summary( sigtest.model.np.g.aa.wvr.of.PLF_TF13 )
summary( model.np.g.aa.wvr.of.PLF_TF13$grad )
# model.np.g.aa.lr.of.PLF_TF13
bw.g.aa.lr.of.PLF_TF13 <- npregbw( lY ~ lL + lA + lV + lK + T_of +ID,
                                   regtype = "ll",
                                   bwmethod = "cv.aic",
                                   ckertype = "gaussian",
                                   ukertype = "aitchisonaitken",
                                   okertype = "liracine",
                                   nmulti = 10,
                                   data = PLF_TF13 )
summary( bw.g.aa.lr.of.PLF_TF13 )
model.np.g.aa.lr.of.PLF_TF13 <- npreg( bws = bw.g.aa.lr.of.PLF_TF13,
                                       data = PLF_TF13,
                                       gradients = TRUE,
                                       residuals = TRUE )
summary( model.np.g.aa.lr.of.PLF_TF13 )
sigtest.model.np.g.aa.lr.of.PLF_TF13 <- npsigtest( model.np.g.aa.lr.of.PLF_TF13,
                                                   boot.num = 399 )
summary( sigtest.model.np.g.aa.lr.of.PLF_TF13 )
summary( model.np.g.aa.lr.of.PLF_TF13$grad )
# model.np.e.lr.PLF_TF13
bw.e.lr.PLF_TF13 <- npregbw( lY ~ lL + lA + lV + lK + T_uf +ID,
                             regtype = "ll",
                             bwmethod = "cv.aic",
                             ckertype = "epanechnikov",
```

```
                            ukertype = "liracine",
                            nmulti = 10,
                            data = PLF_TF13 )
summary( bw.e.lr.PLF_TF13 )
model.np.e.lr.PLF_TF13 <- npreg( bws = bw.e.lr.PLF_TF13,
                                data = PLF_TF13,
                                gradients = TRUE,
                                residuals = TRUE )
summary( model.np.e.lr.PLF_TF13 )
sigtest.model.np.e.lr.PLF_TF13 <- npsigtest( model.np.e.lr.PLF_TF13,
                                            boot.num = 399 )
summary( sigtest.model.np.e.lr.PLF_TF13 )
summary( model.np.e.lr.PLF_TF13$grad )
# model.np.e.aa.PLF_TF13
bw.e.aa.PLF_TF13 <- npregbw( lY ~ lL + lA + lV + lK + T_uf +ID,
                            regtype = "ll",
                            bwmethod = "cv.aic",
                            ckertype = "epanechnikov",
                            ukertype = "aitchisonaitken",
                            nmulti = 10,
                            data = PLF_TF13 )
summary( bw.e.aa.PLF_TF13 )
model.np.e.aa.PLF_TF13 <- npreg( bws = bw.e.aa.PLF_TF13,
                                data = PLF_TF13,
                                gradients = TRUE,
                                residuals = TRUE )
summary( model.np.e.aa.PLF_TF13 )
sigtest.model.np.e.aa.PLF_TF13 <- npsigtest( model.np.e.aa.PLF_TF13,
                                            boot.num = 399 )
summary( sigtest.model.np.e.aa.PLF_TF13 )
summary( model.np.e.aa.PLF_TF13$grad )
# model.np.g.lr.PLF_TF13
bw.g.lr.PLF_TF13 <- npregbw( lY ~ lL + lA + lV + lK + T_uf +ID,
                            regtype = "ll",
                            bwmethod = "cv.aic",
                            ckertype = "gaussian",
                            ukertype = "liracine",
                            nmulti = 10,
                            data = PLF_TF13 )
summary( bw.g.lr.PLF_TF13 )
model.np.g.lr.PLF_TF13 <- npreg( bws = bw.g.lr.PLF_TF13,
                                data = PLF_TF13,
                                gradients = TRUE,
                                residuals = TRUE )
summary( model.np.g.lr.PLF_TF13 )
sigtest.model.np.g.lr.PLF_TF13 <- npsigtest( model.np.g.lr.PLF_TF13,
                                            boot.num = 399 )
```

```
summary( sigtest.model.np.g.lr.PLF_TF13 )
summary( model.np.g.lr.PLF_TF13$grad )
# model.np.g.aa.PLF_TF13
bw.g.aa.PLF_TF13 <- npregbw( lY ~ lL + lA + lV + lK + T_uf +ID,
                            regtype = "ll",
                            bwmethod = "cv.aic",
                            ckertype = "gaussian",
                            ukertype = "aitchisonaitken",
                            nmulti = 10,
                            data = PLF_TF13 )
summary( bw.g.aa.PLF_TF13 )
model.np.g.aa.PLF_TF13 <- npreg( bws = bw.g.aa.PLF_TF13,
                                data = PLF_TF13,
                                gradients = TRUE,
                                residuals = TRUE )
summary( model.np.g.aa.PLF_TF13 )
sigtest.model.np.g.aa.PLF_TF13 <- npsigtest( model.np.g.aa.PLF_TF13,
                                            boot.num = 399 )
summary( sigtest.model.np.g.aa.PLF_TF13 )
summary( model.np.g.aa.PLF_TF13$grad )
# model.np.e.lr.ow.PLF_TF13
bw.e.lr.ow.PLF_TF13 <- npregbw( lY ~ lL + lA + lV + lK +ID,
                               regtype = "ll",
                               bwmethod = "cv.aic",
                               ckertype = "epanechnikov",
                               ukertype = "liracine",
                               nmulti = 10,
                               data = PLF_TF13 )
summary( bw.e.lr.ow.PLF_TF13 )
model.np.e.lr.ow.PLF_TF13 <- npreg( bws = bw.e.lr.ow.PLF_TF13,
                                   data = PLF_TF13,
                                   gradients = TRUE,
                                   residuals = TRUE )
summary( model.np.e.lr.ow.PLF_TF13 )
sigtest.model.np.e.lr.ow.PLF_TF13 <- npsigtest( model.np.e.lr.ow.PLF_TF13,
                                               boot.num = 399 )
summary( sigtest.model.np.e.lr.ow.PLF_TF13 )
summary( model.np.e.lr.ow.PLF_TF13$grad )
# model.np.e.aa.ow.PLF_TF13
bw.e.aa.ow.PLF_TF13 <- npregbw( lY ~ lL + lA + lV + lK +ID,
                               regtype = "ll",
                               bwmethod = "cv.aic",
                               ckertype = "epanechnikov",
                               ukertype = "aitchisonaitken",
                               nmulti = 10,
                               data = PLF_TF13 )
summary( bw.e.aa.ow.PLF_TF13 )
```

```
model.np.e.aa.ow.PLF_TF13 <- npreg( bws = bw.e.aa.ow.PLF_TF13,
                                   data = PLF_TF13,
                                    gradients = TRUE,
                                    residuals = TRUE )
summary( model.np.e.aa.ow.PLF_TF13 )
sigtest.model.np.e.aa.ow.PLF_TF13 <- npsigtest( model.np.e.aa.ow.PLF_TF13,
                                                boot.num = 399 )
summary( sigtest.model.np.e.aa.ow.PLF_TF13 )
summary( model.np.e.aa.ow.PLF_TF13$grad )
# model.np.g.lr.ow.PLF_TF13
bw.g.lr.ow.PLF_TF13 <- npregbw( lY ~ lL + lA + lV + lK +ID,
                               regtype = "ll",
                               bwmethod = "cv.aic",
                               ckertype = "gaussian",
                               ukertype = "liracine",
                               nmulti = 10,
                               data = PLF_TF13 )
summary( bw.g.lr.ow.PLF_TF13 )
model.np.g.lr.ow.PLF_TF13 <- npreg( bws = bw.g.lr.ow.PLF_TF13,
                                   data = PLF_TF13,
                                   gradients = TRUE,
                                   residuals = TRUE )
summary( model.np.g.lr.ow.PLF_TF13 )
sigtest.model.np.g.lr.ow.PLF_TF13 <- npsigtest( model.np.g.lr.ow.PLF_TF13,
                                               boot.num = 399 )
summary( sigtest.model.np.g.lr.ow.PLF_TF13 )
summary( model.np.g.lr.ow.PLF_TF13$grad )
# model.np.g.aa.ow.PLF_TF13
bw.g.aa.ow.PLF_TF13 <- npregbw( lY ~ lL + lA + lV + lK +ID,
                               regtype = "ll",
                               bwmethod = "cv.aic",
                               ckertype = "gaussian",
                               ukertype = "aitchisonaitken",
                               nmulti = 10,
                               data = PLF_TF13 )
summary( bw.g.aa.ow.PLF_TF13 )
model.np.g.aa.ow.PLF_TF13 <- npreg( bws = bw.g.aa.ow.PLF_TF13,
                                   data = PLF_TF13,
                                   gradients = TRUE,
                                   residuals = TRUE )
summary( model.np.g.aa.ow.PLF_TF13 )
sigtest.model.np.g.aa.ow.PLF_TF13 <- npsigtest( model.np.g.aa.ow.PLF_TF13,
                                               boot.num = 399 )
summary( sigtest.model.np.g.aa.ow.PLF_TF13 )
summary( model.np.g.aa.ow.PLF_TF13$grad )
# model.np.e.pool.PLF_TF13
bw.e.pool.PLF_TF13 <- npregbw( lY ~ lL + lA + lV + lK,
```

```
                              regtype = "ll",
                              bwmethod = "cv.aic",
                              ckertype = "epanechnikov",
                              nmulti = 10,
                              data = PLF_TF13 )
summary( bw.e.pool.PLF_TF13 )
model.np.e.pool.PLF_TF13 <- npreg( bws = bw.e.pool.PLF_TF13,
                                  data = PLF_TF13,
                                  gradients = TRUE,
                                  residuals = TRUE )
summary( model.np.e.pool.PLF_TF13 )
sigtest.model.np.e.pool.PLF_TF13 <- npsigtest( model.np.e.pool.PLF_TF13,
                                              boot.num = 399 )
summary( sigtest.model.np.e.pool.PLF_TF13 )
summary( model.np.e.pool.PLF_TF13$grad )
# model.np.g.pool.PLF_TF13
bw.g.pool.PLF_TF13 <- npregbw( lY ~ lL + lA + lV + lK,
                              regtype = "ll",
                              bwmethod = "cv.aic",
                              ckertype = "gaussian",
                              nmulti = 10,
                              data = PLF_TF13 )
summary( bw.g.pool.PLF_TF13 )
model.np.g.pool.PLF_TF13 <- npreg( bws = bw.g.pool.PLF_TF13,
                                  data = PLF_TF13,
                                  gradients = TRUE,
                                  residuals = TRUE )
summary( model.np.g.pool.PLF_TF13 )
sigtest.model.np.g.pool.PLF_TF13 <- npsigtest( model.np.g.pool.PLF_TF13,
                                              boot.num = 399 )
summary( sigtest.model.np.g.pool.PLF_TF13 )
summary( model.np.g.pool.PLF_TF13$grad )
```

C.6. Partial output elasticities of intermediate inputs (Table 4)

```
## create E_int matrix
E_int <- matrix( data = NA, nrow = 4, ncol = 4 )
colnames( E_int ) <- c( "Mean", "Median","MSE␣to␣E","MAD␣to␣E" )
rownames( E_int ) <- c( "cv_R","eV(E␣LRU)", "eV(E␣LRU␣WVR)", "eV(TL␣2WW)" )

## calculate the value of the intermediate inputs
V_int <- ( PLF_TF13$SE281 + PLF_TF13$SE336 ) / 1000
## calculate the quantity of the intermediate inputs
Q_int <- ( PLF_TF13$SE281_2004 + PLF_TF13$SE336_2004 ) / 1000
## calculate the price of the intermediate inputs
P_int <- V_int / Q_int
## calculate the value of the output
```

```r
V_out <- PLF_TF13$SE131 / 1000
## calculate the quantity of the output
Q_out <- PLF_TF13$SE131_2004 / 1000
## calculate the price of the output
P_out <- V_out / Q_out
## calculate the ratio between the costs of intermediate inputs and
## the total revenue
E <- ( P_int * Q_int )/( P_out * Q_out )
summary( E )
E2 <- V_int / V_out
all.equal( E, E2 )
## calculate partial output elasticities (tlEla2 function is based on
## tlEla function)
tlEla2 <- function( object ) {
  tlCoef <- coef( object )
  mDat <- ( PLF_TF13_pd[ , c( "L", "A", "V", "K" ) ] )
  result <- numeric(4)
  result[1] <- tlCoef["log(L)"] +
    tlCoef["I(0.5␣*␣log(L)^2)"] * log( mDat["L"] ) +
    tlCoef["I(log(L)␣*␣log(A))"] * log( mDat["A"] ) +
    tlCoef["I(log(L)␣*␣log(V))"] * log( mDat["V"] ) +
    tlCoef["I(log(L)␣*␣log(K))"] * log( mDat["K"] )
  result[2] <- tlCoef["log(A)"] +
    tlCoef["I(log(L)␣*␣log(A))"] * log( mDat["L"] ) +
    tlCoef["I(0.5␣*␣log(A)^2)"] * log( mDat["A"] ) +
    tlCoef["I(log(A)␣*␣log(V))"] * log( mDat["V"] ) +
    tlCoef["I(log(A)␣*␣log(K))"] * log( mDat["K"] )
  result[3] <- tlCoef["log(V)"] +
    tlCoef["I(log(L)␣*␣log(V))"] * log( mDat["L"] ) +
    tlCoef["I(log(A)␣*␣log(V))"] * log( mDat["A"] ) +
    tlCoef["I(0.5␣*␣log(V)^2)"] * log( mDat["V"] ) +
    tlCoef["I(log(V)␣*␣log(K))"] * log( mDat["K"] )
  result[4] <- tlCoef["log(K)"] +
    tlCoef["I(log(L)␣*␣log(K))"] * log( mDat["L"] ) +
    tlCoef["I(log(A)␣*␣log(K))"] * log( mDat["A"] ) +
    tlCoef["I(log(V)␣*␣log(K))"] * log( mDat["V"] ) +
    tlCoef["I(0.5␣*␣log(K)^2)"] * log( mDat["K"] )
  return( result )
}
## fill the Table 4 (E_int matrix ) with calculated values
E_int[1,1] <- mean( E )
E_int[2,1] <- mean( model.np.e.lr.ow.PLF_TF13$grad[,3] )
E_int[3,1] <- mean( model.np.e.lr.wvr.of.PLF_TF13$grad[,3] )
E_int[4,1] <- mean( tlEla2( tl_w_it )[[3]] )
E_int[1,2] <- median( E )
E_int[2,2] <- median( model.np.e.lr.ow.PLF_TF13$grad[,3] )
E_int[3,2] <- median( model.np.e.lr.wvr.of.PLF_TF13$grad[,3] )
```

```
E_int[4,2] <- median( tlEla2( tl_w_it )[[3]])
E_int[2,3] <- mean( ( E - model.np.e.lr.ow.PLF_TF13$grad[,3] )^2 )
E_int[3,3] <- mean( ( E - model.np.e.lr.wvr.of.PLF_TF13$grad[,3] )^2 )
E_int[4,3] <- mean( ( E - (tlEla2( tl_w_it )[[3]] ) )^2 )
E_int[2,4] <- sum(abs( E - model.np.e.lr.ow.PLF_TF13$grad[,3] ) )/
  dim( PLF_TF13 )[1]
E_int[3,4] <- sum( abs( E - model.np.e.lr.wvr.of.PLF_TF13$grad[,3] ))/
  dim( PLF_TF13 )[1]
E_int[4,4] <- sum( abs( E - ( tlEla2( tl_w_it )[[3]]) ) )/dim( PLF_TF13 )[1]
print( E_int, digits=4 )
```

References

Ai, C. and Li, Q. (2008). Semi-parametric and non-parametric methods in panel data models. In Mátyás, L. and Sevestre, P. (eds), *The Econometrics of Panel Data*. Springer Berlin Heidelberg, Advanced Studies in Theoretical and Applied Econometrics *46*, 451–478.

Aitchison, J. and Aitken, C. G. G. (1976). Multivariate binary discrimination by the kernel method. *Biometrika* 63: 413–420.

Altonji, J. G. and Matzkin, R. L. (2005). Cross section and panel data estimators for nonseparable models with endogenous regressors. *Econometrica* 73: 1053–1102.

Arellano, M. (2003). *Panel Data Econometrics*. Advanced Texts In Econometrics. Oxford University Press, USA.

Baltagi, B. H. (2005). *Econometric Analysis of Panel Data*. New York: Wiley.

Baltagi, B. H. and Li, D. (2002). Series estimation of partially linear panel data models with fixed effects. *Annals of Economic and Finance* 3: 103–116.

Buja, A., Hastie, T. and Tibshirani, R. (1989). Linear smoothers and additive models. *The Annals of Statistics* 17: 453–510.

Croissant, Y. and Millo, G. (2008). Panel data econometrics in R: The plm package. *Journal of Statistical Software* 27: 1–43.

Czekaj, T. and Henningsen, A. (2012). Comparing Parametric and Nonparametric Regression Methods for Panel Data: the Optimal Size of Polish Crop Farms. FOI Working Paper 2012/12, Institute of Food and Resource Economics, University of Copenhagen, http://EconPapers.repec.org/RePEc:foi:wpaper:2012_12.

Evdokimov, K. (2010). Identifcation and Estimation of a Nonparametric Panel Data Model with Unobserved Heterogeneity. Working paper, Department of Economics, Princeton University.

Gyimah-Brempong, K. and Racine, J. S. (2010). Aid and investment in ldcs: A robust approach. *The Journal of International Trade & Economic Development: An International and Comparative Review* 19: 319–349.

Hausman, J. A. (1978). Specification test in econometrics. *Econometrica* 46: 1251–1272.

Hayfield, T. and Racine, J. S. (2008). Nonparametric econometrics: The np package. *Journal of Statistical Software* 27: 1–32.

Henderson, D., Carroll, R. and Li, Q. (2008). Nonparametric estimation and testing of fixed effects panel data models. *Journal of Econometrics* 144: 257–275.

Henderson, D. and Ullah, A. (2005). A nonparametric random effects estimator. *Economics Letters* 88: 403–407.

Henderson, D. J. and Simar, L. (2005). A Fully Nonparametric Stochastic Frontier Model for Panel Data. Working Paper 0519, Departmentment of Economics, State University of New York at Binghamton.

Hoderlein, S. and White, H. (2012). Nonparametric identification in nonseparable panel data models with generalized fixed effects. *Journal of Econometrics* 168: 300–314.

Hsiao, C. (2003). *Analysis of Panel Data*. Cambridge University Press.

Hsiao, C., Li, Q. and Racine, J. (2007). A consistent model specification test with mixed discrete and continuous data. *Journal of Econometrics* 140: 802–826.

Hurvich, C. M., Simonoff, J. S. and Tsai, C. L. (1998). Smooting parameter selection in nonparametric regression using an improved Akaike information criterion. *Journal of the Royal Statistical Society Series B* 60: 271–293.

Lee, Y. and Mukherjee, D. (2008). New nonparametric estimation of the marginal effects in fixed-effects panel models: An application on the environmental kuznets curve.

Li, Q. and Racine, J. S. (2004). Cross-validated local linear nonparametric regression. *Statistica Sinica* 14: 485–512.

Li, Q. and Racine, J. S. (2007). *Nonparametric Econometrics: Theory and Practice*. Princeton: Princeton University Press.

Li, Q. and Stengos, T. (1996). Semiparametric estimation of partially linear panel data models. *Journal of Econometrics* 71: 389–397.

Lin, X. and Carroll, R. J. (2000). Nonparametric function estimation for clustered data when the predictor is measured without/with error. *Journal of the American Statistical Association* 95.

Linton, O. and Nielsen, J. P. (1995). A kernel method of estimating structured nonparametric regression based on marginal integration. *Biometrika* 82: 93–100.

Opsomer, J. D. and Ruppert, D. (1997). Fitting a bivariate additive model by local polynomial regression. *The Annals of Statistics* 25: 186–211.

Porter, J. R. (1996). Essay in Econometrics. Ph.D. thesis, Massachusetts Institute of Technology. Dept. of Economics.

R Development Core Team (2012). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0.

Racine, J. S. (1997). Consistent significance testing for nonparametric regression. *Journal of Business and Economic Statistics* 15: 369–379.

Racine, J. S. (2008). Nonparametric econometrics: A primer. *Foundations and Trends in Econometrics* 3: 1–88.

Racine, J. S. (2009). Nonparametric and semiparametric methods in r. In *Nonparametric Econometric Methods*. Emerald Group Publishing Limited, Advances in Econometrics *25*, 335–375.

Racine, J. S., Hart, J. and Li, Q. (2006). Testing the significance of categorical predictor variables in nonparametric regression models. *Econometric Reviews* 25: 523–544.

Racine, J. S. and Li, Q. (2004). Nonparametric estimation of regression functions with both categorical and continuous data. *Journal of Econometrics* 119: 99–130.

Ramsey, J. B. (1969). Tests for specification errors in classical linear least-squares regression analysis. *Journal of the Royal Statistical Society. Series B (Methodological)* 31: 350–371.

Silverman, B. (1986). *Density Estimation for Statistics and Data Analysis*. London and New York: Chapman and Hall.

Su, L. and Ullah, A. (2007). More efficient estimation of nonparametric panel data models with random effects. *Economics Letters* 96: 375–380.

Su, L. and Ullah, A. (2010). Nonparametric and semiparametric panel econometric models: Estimation and testing. In Ullah, A. and Giles, D. E. A. (eds), *Handbook of Empirical Economics and Finance*. Chapman and Hall/CRC, 455–497.

Taylor, C. (1989). Bootstrap choice of the smoothing parameter in kernel density estimation. *Biometrika* 76: 705–712.

Ullah, A. and Roy, N. (1998). Nonparametric and semiparametric econometrics of panel data. In Ullah, A. and Giles, D. E. A. (eds), *Handbook of Applied Economics Statistics*. Marcel Dekker, Statistics, Textbooks and Monographs *155*, 579–604.

Utts, J. (1982). The rainbow test for lack of fit in regression. *Communications in Statistics-Theory and Methods* 11: 2801–2815.

Wang, M.-C. and van Ryzin, J. (1981). A class of smooth estimators for discrete distributions. *Biometrika* 68: 301–309.

Wang, N. (2003). Marginal nonparametric kernel regression accounting for within-subject correlation. *Biometrika* 90: 43–52.

Wooldridge, J. M. (2002). *Econometric Analysis of Cross Section and Panel Data*. The MIT press.

Zeileis, A. and Hothorn, T. (2002). Diagnostic checking in regression relationships. *R News* 2: 7–10.