unibz
Fakultät für Wirtschaftswissenschaften
Facoltà di Economia
Faculty of Economics and Management

BEMPS –
Bozen Economics & Management
Paper Series

NO 113/ 2025

PanelTM: an R package for two- and three- way dynamic panel threshold regression model

F. Marta L. Di Lascio, Selene Perazzini

PanelTM: an R package for two- and three- way dynamic panel threshold regression model

F. Marta L. Di Lascio[*], Selene Perazzini [†]

Abstract

This paper presents the R package PanelTM, which provides tools for estimating two- and three-way dynamic panel threshold regression models. Estimation is performed using a generalized method of moments approach based on first-difference transformations and instrumental variables as developed by Seo and Shin (2016) and applied in a three-way fashion by Di Lascio and Perazzini (2024, 2022). In addition to model estimation, PanelTM offers functionalities for change point detection, simulation and performance evaluation within panel structures with regime switches. The package is particularly suited to applications requiring the identification of structural breaks in complex panel data, with support for both exogenous and endogenous variables and for threshold heterogeneity across multiple dimensions.

Keywords: Generalized method of moments, panel data, threshold model, R

JEL Code: C87, C13, C23, L66

1 Introduction

Threshold regression models for panel data have attracted growing interest due to their ability to detect structural breaks and to capture asymmetric effects of explanatory variables relative to a threshold variable. These models are particularly suited to identifying non-linear dynamics and regime-dependent behaviour in complex datasets. Methodological progress in this area has ranged from early self-exciting threshold autoregressive models (Tong, 1990; Hansen, 2000) to the more recent dynamic panel threshold models with endogenous transition variables proposed by Seo and Shin (2016). Despite these developments, software tools for estimating threshold models in panel data remain relatively scarce, especially when compared to the broad range of packages available for time series and cross-sectional data.

[*]Faculty of Economics and Management, Free University of Bozen-Bolzano, Piazza Università, 1 - 39100, Bozen-Bolzano, Italy, e-mail: marta.dilascio@unibz.it

[†]Faculty of Economics and Management, Free University of Bozen-Bolzano, Piazza Università, 1 - 39100, Bozen-Bolzano, Italy, e-mail: selene.perazzini@unibz.it

Among existing software, R stands out for its extensive list of packages tailored to panel data analysis. The most widely used is plm (Croissant and Millo, 2008), which provides a comprehensive framework for estimating static and dynamic panel models. Additional packages include OrthoPanels (Cubranic et al., 2022), which implements the orthogonal reparameterisation for dynamic models by Lancaster (2002); pdynmc (Fritsch et al., 2021), for linear dynamic panel estimation; and panelr (Long, 2020), focused on general regression models with panel structures. While not specific to panel data, packages such as lme4 (Bates et al., 2015) also offer tools applicable to panel-like structures through multilevel modelling. Nevertheless, none of these packages support threshold regression, with the exception of pdR (Tsung-wu, 2024), which estimates the non-dynamic panel threshold model of Hansen (1999). Classical threshold models of this kind assume static specifications and homogeneity in threshold effects. By contrast, more recent approaches, such as Seo and Shin (2016), allow for endogenous thresholds and dynamic elements, enabling the analysis of both temporal and cross-sectional heterogeneity. An implementation of Seo and Shin (2016)'s method is available in STATA via the xthenreg command (Seo et al., 2019), which performs first-differenced (FD) Generalized Method of Moments (GMM) estimation for dynamic panel threshold models. While this implementation is useful, it is limited to two-dimensional panel structures and does not accommodate three-way panel data. Moreover, STATA is proprietary software, which may limit accessibility for researchers operating in open-source environments.

The R package PanelTM addresses the aforementioned limitations by providing a flexible framework for estimating dynamic panel threshold models, supporting both two-way and three-way data structures. The inclusion of three-way panels constitutes a novel contribution. To date, no existing threshold regression framework allows for panel data with three dimensions in which the threshold parameters are not assumed to be common across all time series, despite the growing interest in multi-dimensional panel models (see, e.g., Mátyás, 1997, 2017; Balazsi et al., 2018). The core methodology implemented in PanelTM builds on the FD-GMM estimator introduced by Seo and Shin (2016) and its adaptation to three-way panels by Di Lascio and Perazzini (2024), which is relevant in applications where threshold effects may differ along a third dimension. The package allows for time-varying regressors (either exogenous or endogenous), lagged dependent variables as transition variables, and, for three-way panels, a threshold parameter that varies across the third dimension of the data structure. This provides researchers with a powerful and versatile tool to detect and interpret structural breaks in panel data, especially when traditional methods are either infeasible or misspecified due to restrictive assumptions on threshold homogeneity or exogeneity. In addition to panel regression, the package offers an easy tool to identify the change point in the time series based on the threshold variable. The reference literature is very extensive (see e.g. Horvàth and Huškovà, 2012; Chan et al., 2013; Cho, 2016) but, to the best of our knowledge, there is still a gap for change point detection in three-way panel data with temporal dynamics and thresholds varying across time series.

In this paper, we present the theoretical foundations and computational implementation of the PanelTM R package. Section 2 introduces the model specification, the GMM estimation procedure, the associated asymptotic properties, the testing procedure for threshold effects, and the criteria for change point identification. The finite sample performance of the estimator is then assessed through an extensive Monte Carlo simulation study. Section 3 describes the main features and functionalities of the package. Section 4 illustrates its practical use and potential applications through two examples: a case study on fruit bioimpedance and a simulation exercise. Section 5 concludes the paper.

2 Methods

The package provides estimation routines for two classes of dynamic panel threshold models: a two-way specification, implemented via the function `ptm2`, and a three-way extension, implemented via `ptm3`. The two-way model corresponds to the formulation introduced by Seo and Shin (2016), while the three-way model builds upon this framework, as developed by Di Lascio and Perazzini (2024). Here, we focus on the three-way model, noting that the third-way levels are assumed independent so that the two-way model is a special case where the third dimension is degenerate, i.e. $J = 1$. The following subsections present, in order: the model specification, the estimation strategy, the asymptotic theory, the test for threshold effects, the criteria for change point identification, and a Monte Carlo simulation study to show its performance.

2.1 Model

Let $i = 1, \ldots, n$ denote statistical units, $t = 1, \ldots, T$ time periods, and $j = 1, \ldots, J$ levels of a third dimension. The three-way dynamic panel threshold model is:

$$\begin{aligned} y_{ijt} &= (1, \boldsymbol{x}'_{ijt})\boldsymbol{\phi}_{1j}\,\mathbb{1}\{q_{ijt} \leq \gamma_j\} + (1, \boldsymbol{x}'_{ijt})\boldsymbol{\phi}_{2j}\,\mathbb{1}\{q_{ijt} > \gamma_j\} + \varepsilon_{ijt}, \\ \varepsilon_{ijt} &= \mu_i + \lambda_j + \nu_{ijt}, \end{aligned} \tag{1}$$

where y_{ijt} be the dependent variable, q_{ijt} is the threshold variable, and $\boldsymbol{x}_{ijt}$ is a vector of k_1 time-varying covariates, which may include lagged values of y_{ijt}. The coefficient vectors $\boldsymbol{\phi}_{1j}$ and $\boldsymbol{\phi}_{2j}$ represent regime-specific intercepts and slopes, while γ_j is a threshold parameter specific to each j. $\mathbb{1}\{\cdot\}$ is the indicator function for the regime switch based on the threshold variable q_{ijt}. The error term is composed by μ_i, which captures individual fixed effects, and λ_j, which accounts for third-way fixed effects, and the idiosyncratic error ν_{ijt} satisfying $E(\nu_{ijt} \mid \mathcal{F}_{t-1}) = 0$, with $\{\mathcal{F}_t\}$ denoting the natural filtration. Thus, ν_{ijt} is a martingale difference sequence.

2.2 Estimation

The `ptm3` function estimates the model in Eq. (1) using the two-step FD-GMM described in Di Lascio and Perazzini (2024), Sect. 3, and based on the contribution by Seo and Shin (2016). The estimation procedure can be summarized as follows.

Let us consider the first-differentiated model

$$\Delta y_{ijt} = y_{ijt} - y_{ij(t-1)} = \boldsymbol{\beta}_j' \Delta x_{ijt} + \boldsymbol{\delta}_j' \boldsymbol{X}_{ijt}' \mathbb{1}_{ijt}(\gamma_j) + \Delta\varepsilon_{ijt}$$

where Δ is the first difference operator, $\boldsymbol{\beta}_j = \left(\phi_{1j}^{(2)}, \ldots, \phi_{1j}^{(k_1+1)}\right)'$, $\boldsymbol{\delta}_j = (\boldsymbol{\phi}_{2j} - \boldsymbol{\phi}_{1j})$,

$$\boldsymbol{X}_{ijt} = \begin{pmatrix} (1, \boldsymbol{x}_{ijt}') \\ \left(1, \boldsymbol{x}_{ij(t-1)}'\right) \end{pmatrix}, \; \mathbb{1}_{ijt}(\gamma_j) = \begin{pmatrix} \mathbb{1}\{q_{ij(t)} > \gamma_j\} \\ -\mathbb{1}\{q_{ij(t-1)} > \gamma_j\} \end{pmatrix},$$

and $\Delta\varepsilon_{ijt} = \varepsilon_{ijt} - \varepsilon_{ij(t-1)} = \nu_{ijt} - \nu_{ij(t-1)}$. Therefore, $\boldsymbol{\theta}_j = (\boldsymbol{\beta}_j', \boldsymbol{\delta}_j', \gamma_j)'$ is the $(2k_1 + 2)$-dimensional vector of parameters to estimate for each level j. In the first step, the GMM estimator of $\boldsymbol{\beta}_j$ and $\boldsymbol{\delta}_j$ for a fixed γ_j is defined through a grid search algorithm by exploiting l instrumental variables; the GMM estimator of γ_j is then obtained by exploiting the estimate $\left(\hat{\boldsymbol{\beta}}_j(\gamma_j), \hat{\boldsymbol{\delta}}_j(\gamma_j)\right)$ as follows

$$\hat{\gamma}_j = \arg \min_{\boldsymbol{\gamma}_j \in \Gamma_j} \hat{\mathbf{J}}_n^{(j)}(\gamma_j)$$

where $\hat{\mathbf{J}}_n^{(j)}(\gamma_j)$ denotes the objective function evaluated at $\hat{\boldsymbol{\beta}}_j(\gamma_j)$ and $\hat{\boldsymbol{\delta}}_j(\gamma_j)$, given by $\hat{\mathbf{J}}_n^{(j)}(\gamma_j) = \bar{\boldsymbol{g}}_n^{(j)}(\gamma_j)' \mathbf{W}_n^{(j)} \bar{\boldsymbol{g}}_n^{(j)}(\gamma_j)$, and the $(l \times 1)$ vector of sample moment conditions is defined as follows

$$\begin{aligned} \bar{\boldsymbol{g}}_n^{(j)}(\gamma_j) &= \bar{\boldsymbol{g}}_{1n}^{(j)} - \bar{\boldsymbol{g}}_{2n}^{(j)}(\gamma_j) \left(\hat{\boldsymbol{\beta}}_j(\gamma_j)', \hat{\boldsymbol{\delta}}_j(\gamma_j)'\right)' \\ &= \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{g}_{1i}^{(j)} - \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{g}_{2i}^{(j)}(\gamma_j) \left(\hat{\boldsymbol{\beta}}_j(\gamma_j)', \hat{\boldsymbol{\delta}}_j(\gamma_j)'\right)' \end{aligned}$$

and $\boldsymbol{W}_n^{(j)}$ is a $(l \times l)$ weight matrix whose form can be specified in different ways, e.g. $\boldsymbol{W}_n^{(j)} = \boldsymbol{I}_l$ (see Seo and Shin (2016) for additional details). The first-step GMM parameters estimates are then $(\hat{\boldsymbol{\beta}}_j', \hat{\boldsymbol{\delta}}_j', \hat{\gamma}_j)'$ where $(\hat{\boldsymbol{\beta}}_j', \hat{\boldsymbol{\delta}}_j')' = (\hat{\boldsymbol{\beta}}_j(\hat{\gamma}_j)', \hat{\boldsymbol{\delta}}_j(\hat{\gamma}_j)')'$.

The second-step GMM estimators are given by the procedure based on the grid search algorithm described above, updated by exploiting the first-step estimates. Therefore, the (final) GMM estimator of $\boldsymbol{\theta}_j$, which is $\hat{\boldsymbol{\theta}}_j$, is obtained as follows

$$\hat{\boldsymbol{\theta}}_j = \arg \min_{\boldsymbol{\theta}_j \in \Theta_j} \hat{\mathbf{J}}_n^{(j)}(\boldsymbol{\theta}_j)$$

where

$$\hat{\mathbf{J}}_n^{(j)}(\boldsymbol{\theta}_j) = \bar{\boldsymbol{g}}_n^{(j)}(\boldsymbol{\theta}_j)' \mathbf{W}_n^{(j)} \bar{\boldsymbol{g}}_n^{(j)}(\boldsymbol{\theta}_j) \tag{2}$$

with $\mathbf{W}_n^{(j)} = \left(\frac{1}{n}\sum_{i=1}^n \hat{\boldsymbol{g}}_i^{(j)}(\boldsymbol{\theta}_j)\hat{\boldsymbol{g}}_i^{(j)\prime}(\boldsymbol{\theta}_j) - \frac{1}{n^2}\sum_{i=1}^n \hat{\boldsymbol{g}}_i^{(j)}(\boldsymbol{\theta}_j)\sum_{i=1}^n \hat{\boldsymbol{g}}_i^{(j)\prime}(\boldsymbol{\theta}_j)\right)^{-1}$,

$$\begin{aligned}
\hat{\boldsymbol{g}}_i^{(j)}(\boldsymbol{\theta}_j) &= \begin{pmatrix} z_{ijt_0}\left(\Delta y_{ijt_0} - \hat{\boldsymbol{\beta}}_j' \Delta \boldsymbol{x}_{ijt_0} - \hat{\boldsymbol{\delta}}_j' \boldsymbol{X}_{ijt_0}' \mathbb{1}_{ijt_0}(\hat{\gamma}_j)\right) \\ \vdots \\ z_{ijT}\left(\Delta y_{ijT} - \hat{\boldsymbol{\beta}}_j' \Delta \boldsymbol{x}_{ijt_0} - \hat{\boldsymbol{\delta}}_j' \boldsymbol{X}_{ijT}' \mathbb{1}_{ijT}(\hat{\gamma}_j)\right) \end{pmatrix} \\
&= \left(\widehat{\Delta\varepsilon_{ijt}} z_{ijt_0}', \dots, \widehat{\Delta\varepsilon_{ijt}} z_{ijT}'\right),
\end{aligned}$$

and $\widehat{\Delta\varepsilon_{ijt}}$ are the residuals obtained from the first-step estimation. The estimation procedure described is repeated for each value of j to obtain the FD-GMM estimates $\hat{\boldsymbol{\theta}}$ of all the parameters in the three-way model introduced in Eq. (1) of the manuscript.

As previously mentioned, the $(lJ \times lJ)$ weight matrix $\boldsymbol{W}$ is a diagonal block matrix whose blocks on the diagonal are given by a $(l \times l)$ weight matrix $\boldsymbol{W}_n^{(j)}$ concerning the j-th level. Therefore, the whole parameters vector $\boldsymbol{\theta}$ is estimated by applying the GMM estimator above described for each $\boldsymbol{\theta}_j$ taken separately, where $j = 1, \dots, J$. Specifically, the closed-form solution to produce GMM estimates of the whole parameters vector $\hat{\boldsymbol{\theta}}$ is in the diagonal of the following matrix

$$\begin{aligned}
\hat{\mathbf{J}}(\boldsymbol{\theta}) &= \begin{pmatrix} \overline{\boldsymbol{g}}_n^{(1)}(\boldsymbol{\theta}_1)' & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots & \dots & \vdots \\ \mathbf{0} & \dots & \overline{\boldsymbol{g}}_n^{(j)}(\boldsymbol{\theta}_j)' & \dots & \mathbf{0} \\ \vdots & \dots & \dots & \ddots & \vdots \\ \mathbf{0} & \dots & \dots & \mathbf{0} & \overline{\boldsymbol{g}}_n^{(J)}(\boldsymbol{\theta}_J)' \end{pmatrix} \begin{pmatrix} \mathbf{W}_n^{(1)} & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots & \dots & \vdots \\ \mathbf{0} & \dots & \mathbf{W}_n^{(j)} & \dots & \mathbf{0} \\ \vdots & \dots & \dots & \ddots & \vdots \\ \mathbf{0} & \dots & \dots & \mathbf{0} & \mathbf{W}_n^{(J)} \end{pmatrix} \\
& \begin{pmatrix} \overline{\boldsymbol{g}}_n^{(1)}(\boldsymbol{\theta}_1) & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots & \dots & \vdots \\ \mathbf{0} & \dots & \overline{\boldsymbol{g}}_n^{(j)}(\boldsymbol{\theta}_j) & \dots \mathbf{0} & \\ \vdots & \dots & \dots & \ddots & \vdots \\ \mathbf{0} & \dots & \dots & \dots \mathbf{0} & \overline{\boldsymbol{g}}_n^{(J)}(\boldsymbol{\theta}_J) \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{J}}_n^{(1)}(\boldsymbol{\theta}_1) & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots & \dots & \vdots \\ \mathbf{0} & \dots & \hat{\mathbf{J}}_n^{(j)}(\boldsymbol{\theta}_j) & \dots & \mathbf{0} \\ \vdots & \dots & \dots & \ddots & \vdots \\ \mathbf{0} & \dots & \dots & \mathbf{0} & \hat{\mathbf{J}}_n^{(J)}(\boldsymbol{\theta}_J) \end{pmatrix}
\end{aligned} \quad (3)$$

where $\hat{\mathbf{J}}_n^{(j)}(\boldsymbol{\theta}_j) = \overline{\boldsymbol{g}}_n^{(j)}(\boldsymbol{\theta}_j)'\mathbf{W}_n^{(j)}\overline{\boldsymbol{g}}_n^{(j)}(\boldsymbol{\theta}_j)$ with $j = 1, \dots, J$. Hence,

$$\hat{\boldsymbol{\theta}} = \left(\arg\min_{\boldsymbol{\theta}_1 \in \Theta_1} \hat{\mathbf{J}}_n^{(1)}(\boldsymbol{\theta}_1), \dots, \arg\min_{\boldsymbol{\theta}_j \in \Theta_j} \hat{\mathbf{J}}_n^{(j)}(\boldsymbol{\theta}_j), \dots, \arg\min_{\boldsymbol{\theta}_J \in \Theta_J} \hat{\mathbf{J}}_n^{(J)}(\boldsymbol{\theta}_J)\right) \quad (4)$$

where $\hat{\mathbf{J}}_n^{(j)}(\boldsymbol{\theta}_j)$ is as given in Eq. (2) and subsequent equations.

2.3 Asymptotic theory

The asymptotic properties of the GMM estimator based on the FD transformation were first developed by Hansen (2000). In the case of exogenous threshold variables, asymptotic theory has been extensively studied in the context of static panel models (Hansen, 1999). Separately, a rich literature addresses

GMM estimation in linear dynamic panel models (see, e.g., Arellano and Bond, 1991; Blundell and Bond, 1998; Hsiao and Zhang, 2015). More recently, Seo and Shin (2016) developed a comprehensive asymptotic framework for dynamic panel threshold models. This includes consistent and efficient estimation of the threshold parameter, as well as inference procedures for both threshold effects and the endogeneity of the transition variable.

Since Eq. (1) assumes independence between the levels of the third way, we estimate the model's parameters for each level j of the third way taken separately and, for each j, the standard GMM asymptotics as well as the FD-GMM asymptotics are still valid. Thus, the standard GMM asymptotics and the further development in Seo and Shin (2016) are also valid for the model in Eq. (1), so we can state that, for each j, (i) the FD-GMM estimator always follows a normal distribution asymptotically

$$\begin{pmatrix} \sqrt{n}\begin{pmatrix} \hat{\boldsymbol{\beta}}_j - \boldsymbol{\beta}_{jn} \\ \hat{\boldsymbol{\delta}}_j - \boldsymbol{\delta}_{j0} \end{pmatrix} \\ n^{1/2-\alpha}(\hat{\gamma}_j - \gamma_{j0}) \end{pmatrix} \xrightarrow[n\to\infty]{d} \mathcal{N}\left(0, (\boldsymbol{G}_j'\boldsymbol{\Omega}_j^{-1}\boldsymbol{G}_j)^{-1}\right)$$

where the true value of $\boldsymbol{\beta}_j$ is fixed at $\boldsymbol{\beta}_{j0}$ while that of $\boldsymbol{\delta}_j$ depends on n such that $\boldsymbol{\delta}_{jn} = \boldsymbol{\delta}_{j0}n^{-\alpha}$ for some $0 \leq \alpha < 1/2$ and $\boldsymbol{\delta}_{j0} \neq 0$, Ω_j is finite and positive definite, $\boldsymbol{G}_j = (\boldsymbol{G}_{\beta_j}, \boldsymbol{G}_{\delta_j}(\gamma_{j0}), \boldsymbol{G}_{\gamma_j}(\gamma_{j0}))$ is of full rank and it is composed of

$$\boldsymbol{G}_{\beta_j} = \begin{bmatrix} -E(z_{ijt_0}\Delta\boldsymbol{x}'_{ijt_0}) \\ \vdots \\ -E(z_{ijT}\Delta\boldsymbol{x}'_{ijT}) \end{bmatrix}, \quad \boldsymbol{G}_{\delta_j}(\gamma_j) = \begin{bmatrix} -E(z_{ijt_0}\mathbb{1}_{ijt_0}(\gamma_j)'\boldsymbol{x}_{ijt_0}) \\ \vdots \\ -E(z_{ijT}\mathbb{1}_{ijT}(\gamma_j)'\boldsymbol{x}_{ijT}) \end{bmatrix},$$

and

$$\boldsymbol{G}_{\gamma_j}(\gamma_j) = \begin{bmatrix} \{E_{t_0-1}[z_{ijt_0}(\mathbb{1}_{ijt_0-1}, \boldsymbol{x}'_{ij(t_0-1)})|\gamma_j]p_{t_0-1}(\gamma_j) - E_{t_0}[z_{ijt_0}(\mathbb{1}_{ijt_0}, \boldsymbol{x}'_{ij(t_0)})|\gamma_j]p_{t_0}(\gamma_j)\}\delta_{0j} \\ \vdots \\ \{E_{T-1}[z_{ijT}(\mathbb{1}_{ijT-1}, \boldsymbol{x}'_{ij(T-1)})|\gamma_j]p_{T-1}(\gamma_j) - E_T[z_{ijT}(\mathbb{1}_{ijT}, \boldsymbol{x}'_{ijT})|\gamma_j]p_T(\gamma_j)\}\delta_{0j} \end{bmatrix},$$

where $E_t[\cdot|\gamma_j]$ denotes the conditional expectation given $q_{ijt} = \gamma_j$ and $p_t(\cdot)$ the density of q_{ijt} assumed continuous and bounded, and $\boldsymbol{\Omega}_j$ can be obtained as $\boldsymbol{W}_n^{(j)-1}$.

2.4 Testing for threshold effect

An important issue related to the three-way panel threshold model in Eq. (1) is to test whether there is a statistically significant regime switch in a sequence of chronologically ordered data. Therefore, a bootstrap-based testing procedure for the presence of the threshold effect was included in both `ptm2` and `ptm3` through the option `test.lin=TRUE`. The test is based on the work by Seo et al. (2019) and is defined over the following hypothesis system for each j

$$\begin{cases} H_0 : \boldsymbol{\delta}_{j0} & = 0, \text{ for any } \gamma_j \in \Gamma_j \\ H_1 : \boldsymbol{\delta}_{j0} & \neq 0, \text{ for some } \gamma_j \in \Gamma_j \end{cases}$$

where Γ_j is the parametric space for γ_j. Using the standard approach based on the supremum statistics

$$\sup \boldsymbol{W}_j = \sup_{\gamma_j \in \Gamma_j} \mathcal{W}_n(\gamma_j)$$

where $\mathcal{W}_n(\gamma_j) = n\hat{\boldsymbol{\delta}}_j(\gamma_j)'\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\delta}_j}(\gamma_j)^{-1}\hat{\boldsymbol{\delta}}_j(\gamma_j)$ is the standard Wald statistic for each fixed γ_j,

$$\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\delta}_j}(\gamma_j) = \boldsymbol{R}\left(\left(\hat{\boldsymbol{\Omega}}_j(\hat{\boldsymbol{\theta}}_j(\gamma_j))^{-1/2}(\hat{\boldsymbol{G}}_{\boldsymbol{\beta}_j}, \hat{\boldsymbol{G}}_{\boldsymbol{\delta}_j}(\hat{\boldsymbol{\theta}}_j(\gamma_j)))\right)'\left(\hat{\boldsymbol{\Omega}}_j(\hat{\boldsymbol{\theta}}_j(\gamma_j))^{-1/2}(\hat{\boldsymbol{G}}_{\boldsymbol{\beta}_j}, \hat{\boldsymbol{G}}_{\boldsymbol{\delta}_j}(\hat{\boldsymbol{\theta}}_j(\gamma_j)))\right)\right)^{-1}\boldsymbol{R}'$$

that is a consistent asymptotic variance estimator with $\boldsymbol{R} = (\boldsymbol{0}_{(k_1+1)k_1}, \boldsymbol{I}_{k_1+1})$.

To compute the statistic test, a bootstrap procedure is used. Since bootstrapping in dynamic panel models can be challenging, as discussed by Gong and Seo (2024) who notably demonstrated that the nonparametric bootstrap may be inconsistent, a parametric bootstrap was implemented instead. The main idea is that the residuals $\widehat{\Delta\varepsilon_{ijt}}$ from the original samples are used to compute $\Delta y^{\star}_{ijt} = \widehat{\Delta\epsilon_{ijt}}\eta_i$ where η_i, with $i = 1, \ldots, n$, are i.i.d. observations from the standard normal; next, $\hat{\boldsymbol{\delta}}_j(\gamma_j)^{\star}$ and a bootstrap statistics $\mathcal{W}^{\star}_n(\gamma_j)$ are computed to obtain $\sup \boldsymbol{W}^{\star}_j$. The empirical p-values of the test are computed as the proportion of suprema $\sup \boldsymbol{W}^{\star}_j$ (over Γ_j) in the bootstrap replications that are larger than $\sup \boldsymbol{W}_j$. For firther details refers to Seo et al. (2019).

2.5 Change point identification

We define the change point as the regime switch time in two- or three-way panel threshold regression models by applying the criteria outlined in Sect. 3.2 of Di Lascio and Perazzini (2024). From an applied perspective, this corresponds to detecting the time point at which the series ij exceeds the estimated threshold parameter $\hat{\gamma}_j$:

$$\widehat{\text{CP}}_{ij} = \underset{t \in \{1,\ldots,T\}}{\arg\min} \left\{\mathbb{1}(q_{ijt} > \hat{\gamma}_j)\right\}, \quad (5)$$

thus providing the time of regime change for each pair (i, j). It is important to note that Eq. (5) becomes

$$\widehat{\text{CP}}_{ij} = \underset{t \in \{1,\ldots,T\}}{\arg\min} \left\{\mathbb{1}(q_{ijt} < \hat{\gamma}_j)\right\}$$

when the upper regime occurs at the beginning of the time series. The last two equations provide the time of the regime change for all i, j, i.e. for each time series and level of the third-way, so that we next summarise the change points over i through their mean indicated by CP_j.

2.6 Monte Carlo simulation study

In this section, we explore the finite sample performance of the FD-GMM estimator described in Sect. 2.2. To this end, we perform a Monte Carlo study and investigate its performance in terms of bias and mean squared error of the estimator for β_j, $\delta_{Ij} = \left(\phi_{2j}^{(1)} - \phi_{1j}^{(1)}\right)$, $\delta_{Xj} = \left(\phi_{2j}^{(2)} - \phi_{1j}^{(2)}\right)$, γ_j, and CP_j for $j = 1, \ldots, J$.

We consider the data-generating process as in Eq. (1), and distinguish two cases: one without time-varying regressors and the other with time-varying regressors. We assume that the error term is distributed as a Gaussian white noise, that is $\varepsilon_{ijt} \sim \mathcal{GWN}(0,1)$, $\forall i = 1, \ldots, n$ and $\forall j = 1, \ldots, J$. As for the model with time-varying regressors, we assume the regressor X_{ijt} is distributed as a stationary autoregressive model, $AR(1)$, with coefficient equal to 0.7 $\forall i = 1, \ldots, n$ and $\forall j = 1, \ldots, J$. We simulate different scenarios by varying sample size n in $(50, 150)$ and time series length T in $(11, 50)$, for each level j. We also vary the time-varying exogenous regressor coefficients and the threshold parameter across the values of j (see specific parameter values in the scenarios listed below). Moreover, when simulated time series are short, i.e. $T = 11$, the true value used for the change point is $\text{CP}_{ij} = 8$, while, when $T = 50$, $\text{CP}_{ij} = 20$, for all $i = 1, \ldots, n$ and $j = 1, \ldots, J$. We thus simulate the following scenarios:

1. Eq. (1) with $J = 2$ and without time-varying regressors:
   $y_{i1t} = -\mathbb{1}\left\{y_{i1(t-1)} \leq 0\right\} + \mathbb{1}\left\{y_{i1(t-1)} > 0\right\} + \varepsilon_{i1t}$
   $y_{i2t} = -0.7\mathbb{1}\left\{y_{i2(t-1)} \leq 0\right\} + 1.8\mathbb{1}\left\{y_{i2(t-1)} > 0\right\} + \varepsilon_{i2t}$;

2. Eq. (1) with $J = 2$ and a time-varying regressor, and $\gamma_1 \neq \gamma_2 \neq 0$:
   $y_{i1t} = (0.5 + 0.8x_{i1t})\mathbb{1}\left\{y_{i1(t-1)} \leq 3\right\} + (5 - 0.7x_{i1t})\mathbb{1}\left\{y_{i1(t-1)} > 3\right\} + \varepsilon_{i1t}$
   $y_{i2t} = (5 + 1.2x_{i2t})\mathbb{1}\left\{y_{i2(t-1)} \leq 10\right\} + (11 + 0.3x_{i2t})\mathbb{1}\left\{y_{i2(t-1)} > 10\right\} + \varepsilon_{i2t}$;

3. Model as in the previous case but with $\text{CP}_{i1} \neq \text{CP}_{i2}$, $\text{CP}_{i1} = 20$, and $\text{CP}_{i2} = 30\ \forall i$; for obvious reasons, here we only simulate the case with $T = 50$;

4. Model as in scenario 2. but with $\text{CP}_{i1} \neq \text{CP}_{i2}$, $\text{CP}_{i1} = 7$, and $\text{CP}_{i2} = 8$ $\forall i$; for obvious reasons, here we only simulate the case with $T = 11$.

Further experiments were conducted to assess the performance of the model when the two regimes are purely random processes. The results obtained are consistent with those shown below and related to the aforementioned experiments. They have thus been omitted here and are available upon request.

To assess the performance of the model and its estimation, we perform $B = 500$ replications for each scenario considered and compute the relative bias (RB) and the relative root mean squared error (RRMSE) for the parameters of each model and for the change point. The sample versions of RB and RRMSE are as follows

$$\widehat{\text{RB}} = \frac{1}{B}\sum_{b=1}^{B}\left(\frac{\hat{\psi}_b - \psi}{\psi}\right), \qquad \widehat{\text{RRMSE}} = \sqrt{\frac{1}{B}\sum_{b=1}^{B}\left(\frac{\hat{\psi}_b - \psi}{\psi}\right)^2} \tag{6}$$

where ψ is one of the parameters in the model, i.e. β_j, δ_{Ij}, δ_{Xj}, γ_j, CP_j, and $\hat{\psi}_b$ is the corresponding estimated value at the b-th Monte Carlo replication.

Monte Carlo estimation results are shown in Tabs. from 1 to 4. Note that in order to clearly show the results of the estimation accuracy of the intercept and all coefficients of the model, we have indicated with δ_{Ij} the difference between the lower and upper regime's intercept at the j-th level of the third way and with δ_{Xj} the difference between the lower and upper regime's slope parameter for the j-th level. Regarding the scenario 1. (results in Tab. 1), it appears that the analysed model is able to find the true CP irrespective of the sample size and the time series length, even though a slight worsening is present when $n = T = 50$. As for the estimation accuracy of the model coefficients, the $\widehat{\text{RB}}$ and the $\widehat{\text{RRMSE}}$ of all the estimates show satisfactory values that further improve as n increases and the length of time series decreases. Also for δ_{Ij} and γ_j the worst case is when $n = T = 50$, probably due to an excessively long time series w.r.t. the sample size. Also in the scenario 2. (results in Tab. 2) the identified CPs are very close to the true ones. As expected, all the model parameters show the best performance when the sample size is large ($n = 150$). Finally, β_j coefficients are the only ones showing an accuracy that would merit further study. However, the overall performance of the model is very satisfactory and the introduction of a time-varying regressor does not show a negative impact on it. Finally, in Tabs. (3) and (4) we present the Monte Carlo results for the two most general cases where $\gamma_1 \neq \gamma_2 \neq 0$ and $\text{CP}_{i1} \neq \text{CP}_{i2}$ and the model includes a time-varying regressor. These results confirm the satisfactory performance of the model in terms of both estimation accuracy and change point detection.

Table 1: Simulation results scenario 1.: model in Eq. (1) with $J = 2$ and without time-varying regressors. Note that * indicates that a not relative version of the measure is computed due to null denominator.

CP=8					CP=20				
		$n = 50, T = 11$					$n = 50, T = 50$		
j	Parameter	True value	$\widehat{\text{RB}}$	$\widehat{\text{RRMSE}}$	j	Parameter	True value	$\widehat{\text{RB}}$	$\widehat{\text{RRMSE}}$
1	δ_{I1}	2.0	0.213	0.548	1	δ_{I1}	2.0	-1.489	1.539
1	γ_1	0	-0.219*	0.396*	1	γ_1	0	0.241*	1.109*
1	CP_1	8	-0.049	0.092	1	CP_1	20	0.254	0.481
2	δ_{I2}	2.5	0.091	0.516	2	δ_{I2}	2.5	-1.573	1.610
2	γ_2	0	-0.201*	0.391*	2	γ_2	0	0.531*	1.322*
2	CP_2	8	-0.055	0.101	2	CP_2	20	0.226	0.448
		$n = 150, T = 11$					$n = 150, T = 50$		
j	Parameter	True value	$\widehat{\text{RB}}$	$\widehat{\text{RRMSE}}$	j	Parameter	True value	$\widehat{\text{RB}}$	$\widehat{\text{RRMSE}}$
1	δ_{I1}	2.0	0.240	0.257	1	δ_{I1}	2.0	-0.380	0.428
1	γ_1	0	-0.054*	0.089*	1	γ_1	0	-0.540*	0.700*
1	CP_1	8	-0.012	0.020	1	CP_1	20	-0.100	0.168
2	δ_{I2}	2.5	0.160	0.181	2	δ_{I2}	2.5	-0.482	0.528
2	γ_2	0	-0.039*	0.063*	2	γ_2	0	-0.472*	0.723*
2	CP_2	8	-0.012	0.019	2	CP_2	20	-0.114	0.210

Table 2: Simulation results for scenario 2.: model in Eq. (1) with $J = 2$, a time-varying regressor, $\gamma_1 \neq \gamma_2 \neq 0$.

CP=8					CP=20				
$n = 50, T = 11$					$n = 50, T = 50$				
j	Parameter	True value	$\widehat{\text{RB}}$	$\widehat{\text{RRMSE}}$	j	Parameter	True value	$\widehat{\text{RB}}$	$\widehat{\text{RRMSE}}$
1	β_1	0.8	0.583	1.372	1	β_1	0.8	-0.685	1.212
1	δ_{I1}	4.5	-0.466	0.651	1	δ_{I1}	4.5	-1.680	1.702
1	δ_{X1}	-1.5	0.402	1.017	1	δ_{X1}	-1.5	-0.464	0.747
1	γ_1	3	-0.432	0.549	1	γ_1	3	0.221	0.668
1	CP_1	8	-0.107	0.152	1	CP_1	20	0.275	0.436
2	β_2	1.2	0.171	1.173	2	β_2	1.2	-0.508	0.651
2	δ_{I2}	6	-0.442	0.607	2	δ_{I2}	6	-1.315	1.331
2	δ_{X2}	-0.9	-0.022	2.419	2	δ_{X2}	-0.9	-0.661	0.898
2	γ_2	10	-0.286	0.336	2	γ_2	10	0.030	0.190
2	CP_2	8	-0.120	0.171	2	CP_2	20	0.427	0.527
$n = 150, T = 11$					$n = 150, T = 50$				
j	Parameter	True value	$\widehat{\text{RB}}$	$\widehat{\text{RRMSE}}$	j	Parameter	True value	$\widehat{\text{RB}}$	$\widehat{\text{RRMSE}}$
1	β_1	0.8	-0.121	0.276	1	β_1	0.8	0.085	0.535
1	δ_{I1}	4.5	-0.086	0.132	1	δ_{I1}	4.5	-0.755	0.784
1	δ_{X1}	-1.5	-0.040	0.322	1	δ_{X1}	-1.5	0.021	0.310
1	γ_1	3	-0.160	0.219	1	γ_1	3	-0.266	0.408
1	CP_1	8	-0.028	0.044	1	CP_1	20	-0.048	0.129
2	β_2	1.2	0.105	0.222	2	β_2	1.2	-0.268	0.416
2	δ_{I2}	6	-0.058	0.086	2	δ_{I2}	6	-1.159	1.166
2	δ_{X2}	-0.9	0.121	0.644	2	δ_{X2}	-0.9	-0.380	0.596
2	γ_2	10	-0.137	0.163	2	γ_2	10	-0.057	0.230
2	CP_2	8	-0.028	0.041	2	CP_2	20	0.271	0.441

3 The PanelTM package

`PanelTM` provides two primary functions: `ptm2` and `ptm3`, which are designed to specify and estimate two-way and three-way panel models, respectively. These functions differ in terms of input requirements: `ptm2` expects the user to input the variables as separate matrices, while `ptm3` requires a single `data.frame` containing all relevant variables, with their roles in the model specified explicitly. Despite this difference in input structure, both functions offer the same set of options and features for model specification. The variables to be defined are listed in Tab. 5. If the threshold variable is not specified, the first lag of the dependent variable is used by default. When no regressors are provided, the model will include only a constant term. In case both exogenous and endogenous regressors are specified, the functions always arrange the regressors in the output by placing the exogenous variables first, followed by the endogenous variables, in the same order as specified in the input. An optional set of additional instrumental variables can also be included through the `IV` or `nameIV` option, along with lags of the regressors and the dependent variable that are included by default.

The remaining options in the two functions allow the analyst to tailor the estimation procedure according to their preferences. Specifically, the arguments `trimrate`, `ngrid`, and `h0` are related to the grid search algorithm, setting the

Table 3: Simulation results for scenario 3.: model in Eq. (1) with $J = 2$, a time-varying regressor, $\gamma_1 \neq \gamma_2 \neq 0$, and $\text{CP}_{i1} = 7$ and $\text{CP}_{i2} = 8$, $\forall\, i$.

CP=7					CP=8				
		$n = 50, T = 11$					$n = 50, T = 11$		
j	Parameter	True value	$\widehat{\text{RB}}$	$\widehat{\text{RRMSE}}$	j	Parameter	True value	$\widehat{\text{RB}}$	$\widehat{\text{RRMSE}}$
1	β_1	0.8	0.447	1.290	2	β_2	1.2	0.149	1.089
1	δ_{I1}	4.5	-0.462	0.648	2	δ_{I2}	6	-0.462	0.607
1	δ_{X1}	-1.5	0.281	0.944	2	δ_{X2}	-0.9	0.103	2.198
1	γ_1	3	-0.231	0.553	2	γ_2	10	-0.281	0.328
1	CP_1	7	-0.067	0.127	2	CP_2	8	-0.114	0.161
		$n = 150, T = 11$					$n = 150, T = 11$		
j	Parameter	True value	$\widehat{\text{RB}}$	$\widehat{\text{RRMSE}}$	j	Parameter	True value	$\widehat{\text{RB}}$	$\widehat{\text{RRMSE}}$
1	β_1	0.8	-0.218	0.322	2	β_2	1.2	0.093	0.224
1	δ_{I1}	4.5	-0.051	0.101	2	δ_{I2}	6	-0.061	0.091
1	δ_{X1}	-1.5	-0.174	0.281	2	δ_{X2}	-0.9	0.115	0.652
1	γ_1	3	-0.093	0.255	2	γ_2	10	-0.140	0.166
1	CP_1	7	-0.019	0.045	2	CP_2	8	-0.029	0.042

Table 4: Simulation results for scenario 4.: model in Eq. (1) with $J = 2$ and a time-varying regressor, $\gamma_1 \neq \gamma_2 \neq 0$, $\text{CP}_{i1} = 20$, $\text{CP}_{i2} = 30$, $\forall\, i$.

CP=20					CP=30				
		$n = 50, T = 50$					$n = 50, T = 50$		
j	Parameter	True value	$\widehat{\text{RB}}$	$\widehat{\text{RRMSE}}$	j	Parameter	True value	$\widehat{\text{RB}}$	$\widehat{\text{RRMSE}}$
1	β_1	0.8	-0.638	1.289	2	β_2	1.2	-0.340	0.547
1	δ_{I1}	4.5	-1.714	1.741	2	δ_{I2}	6	-1.424	1.439
1	δ_{X1}	-1.5	-0.420	0.790	2	δ_{X2}	-0.9	-0.565	0.904
1	γ_1	3	0.131	0.689	2	γ_2	10	-0.155	0.292
1	CP_1	20	0.227	0.421	2	CP_2	30	0.002	0.206
		$n = 150, T = 50$					$n = 150, T = 50$		
j	Parameter	True value	$\widehat{\text{RB}}$	$\widehat{\text{RRMSE}}$	j	Parameter	True value	$\widehat{\text{RB}}$	$\widehat{\text{RRMSE}}$
1	β_1	0.8	0.086	0.534	2	β_2	1.2	-0.082	0.274
1	δ_{I1}	4.5	-0.761	0.791	2	δ_{I2}	6	-1.14	1.149
1	δ_{X1}	-1.5	0.025	0.293	2	δ_{X2}	-0.9	-0.201	0.464
1	γ_1	3	-0.247	0.402	2	γ_2	10	-0.193	0.305
1	CP_1	20	-0.038	0.136	2	CP_2	30	-0.035	0.195

trimming rate and the number of grid points used to estimate the threshold, and the parameter for Silverman's rule of thumb in kernel estimation, respectively. The option `Iweight` specifies the weight matrix, where `TRUE` corresponds to the identity matrix and `FALSE` follows the specification provided in Box I of Seo and Shin (2016). Finally, setting `test.lin=TRUE` performs the test for threshold effects described in Sect. 2.4 using `B` bootstrap replications. Note that the methods requires at least 6 times of observation: four lags of the dependent and independent variables are used as instruments, and two more are necessary to identify the regime switch (i.e., one per regime).

Results obtained using the `ptm2` function may exhibit slight differences from those computed with the `xthenreg` function in `STATA` due to the computation of the inverse of matrices when deriving the weight matrix and the GMM estimator. In `STATA`, if a matrix is not positive definite, its rows are sequentially inverted until the diagonal elements become zero or negative, yielding a g_2 inverse. In

Table 5: List of input variables of functions ptm2 and ptm3. Note that in the first column we omitted the pedices of the variables since they change accordingly to the kind of the model, i.e. two- or three-way model.

Variable in Eq. (1)	ptm2	ptm3
y	Y	nameY
q	TV	nameTV
x (exogenous)	Xexo	nameXexo
x (endogenous)	Xendo	nameXendo
z	IV	nameIV

contrast, in ptm2 and ptm3, the pseudo-inverse is computed. A performance comparison between these two implementations is presented in Table 6. For this purpose, three datasets comprising 150 statistical units observed over 50 time periods were randomly generated from the following dynamic panel models:

i $y_{it} = (-1 - 0.2x_{it})\mathbb{1}\left\{y_{i(t-1)} \leq 0\right\} + (1 + 0.2x_{it})\mathbb{1}\left\{y_{i(t-1)} > 0\right\} + \varepsilon_{it};$

ii $y_{it} = (-5 - 0.14x_{it})\mathbb{1}\left\{y_{i(t-1)} \leq -1\right\} + (2 + 0.7x_{it})\mathbb{1}\left\{y_{i(t-1)} > -1\right\} + \varepsilon_{it};$

iii $y_{it} = (5 + 1.2x_{it})\mathbb{1}\left\{y_{i(t-1)} \leq 10\right\} + (11 + 0.3x_{it})\mathbb{1}\left\{y_{i(t-1)} > 10\right\} + \varepsilon_{it}.$

The simulated datasets were subsequently estimated using both ptm2 and xthenreg. The resulting parameter estimates are largely comparable, with neither method demonstrating a consistent advantage over the other.

Table 6: Comparison of the estimates obtained using ptm2 and xthenreg for the three simulated datasets i–iii, with $N = 150$ and $T = 50$.

	Model i			Model ii			Model iii		
	True value	PanelTM	xthenreg	True value	PanelTM	xthenreg	True value	PanelTM	xthenreg
γ	0	-0.08	-0.38	-1	-2.77	-4.49	10	11.74	5.95
β	-0.2	-0.15	-0.25	-0.14	-0.10	-0.41	1.2	0.60	2.67
δ_I	2	0.46	2.48	7	-4.34	9.44	6	-1.36	9.60
δ_X	0.4	0.23	0.38	0.84	0.85	1.05	-0.9	-0.03	-2.37

3.1 Other functions

Alongside the functions ptm2 and ptm3, the PanelTM package includes three additional functions: cpoint, simptm, and perfm.

The cpoint function identifies the regime switch time in two- or three-way panel threshold regression models where $q_{ijt} = y_{ij(t-1)}$ as in Eq. (5). Since a time series may cross above and subsequently fall below the threshold multiple times, the change point is defined as the time t after which the longest uninterrupted sequence of observations within the lower (or upper) regime begins.

The simptm function generates synthetic datasets based on two- or three-way dynamic panel threshold regression models, where the threshold variable is given by $q_{ijt} = y_{ij(t-1)}$. For each level j of the third dimension, the simulated data exhibit two distinct regimes separated by a change point occurring at a prespecified time CP_j, with the transition governed by a regime-specific threshold value γ_j. The model allows for the inclusion of a constant term and (optionally) one exogenous time-varying regressor. Specifically, the data are generated from the following model:

$$y_{ijt} = \begin{cases} \phi_{1j}^{(1)} + \phi_{1j}^{(2)} x_{ijt} + \sigma\varepsilon_{ijt}, & \text{if } y_{ij(t-1)} \leq \gamma_j, \quad t \leq \text{CP}_j, \\ \phi_{2j}^{(1)} + \phi_{2j}^{(2)} x_{ijt} + \sigma\varepsilon_{ijt}, & \text{if } y_{ij(t-1)} > \gamma_j, \quad t > CP_j, \end{cases} \tag{7}$$

where:

- $i = 1, \ldots, n$ represents statistical units,
- $t = 1, \ldots, T$ indicates the time periods,
- $j = 1, \ldots, J$ indexes the level of the third dimension,
- x_{ijt} denotes the value of the exogenous regressor for individual i, level j, and time t, simulated according to an AR(1) process:

  $$x_{ijt} = \rho_j x_{ij(t-1)} + \eta_{ijt}, \tag{8}$$

  where ρ_j is the autoregressive coefficient for level j and η_{ijt} is drawn from a white noise process with zero mean and constant variance,
- ε_{ijt} is drawn from a standard Gaussian white noise for each fixed j,
- σ is a user-defined scaling constant.

The function returns a list of B independently simulated datasets, each structured as a balanced panel. In the panels, each unit i satisfies a regime-switching condition: observations before the change point CP_j correspond to one regime (e.g., $q_{ijt} \leq \gamma_j$), while those after CP_j correspond to the other regime (e.g., $q_{ijt} > \gamma_j$), ensuring a clean regime separation in the simulated sample. This function was used to generate the data for the simulation studies presented in Sect. 1 and 3, and its usage is further illustrated in Sect. 4.2.

Finally, the function `perfm` computes the performance metrics employed to evaluate the Monte Carlo simulation study reported in Tables 1-4, namely the bias, relative bias, root mean squared error, and relative root mean squared error as defined in Eq. (6). For further details on its use, please refer to the example provided in Sect. 4.2.

4 Illustrations

The `R` package `PanelTM` can be accessed and downloaded from `R` CRAN:

```r
#Install
install.packages("PanelTM")

#Import
library(PanelTM)
```

In the subsections below, we propose two examples to illustrate its usage and potential applications. The first example is a case study on fruit bioimpedance, while the second one is a simulation exercise. Furthermore, the code reproducing the Monte Carlo simulation study in Sects. 2.6 and 3 using `PanelTM` is provided in the supplementary material. Beyond these examples, the package is applicable to a wide range of data with similar characteristics. Potential applications include climate change studies, to analyse phenomena driven by extreme precipitation events; medical research, to assess patients' responses to different drug dosages; economics, to detect structural shocks affecting industrial production, unemployment, or prices; finance, to identify regime shifts in market behaviour; and engineering, for monitoring system failures or changes in operating conditions.

4.1 Application to fruit bioimpedance

Alongside the functions described in Sect. 3, `PanelTM` also contains some data about fruit bioimpedance. The data is a subset of a broader database produced and owned by the Sensing Technologies Laboratory at the Free University of Bozen-Bolzano (Italy)[1] and analysed in Di Lascio and Perazzini (2024). The data available on the package concerns a 50-banana production batch observed for 11 days. Variables collected are the weight and bioimpedance measurements at three different electrical frequencies measured using a portable electrical impedance spectroscopy device, called FruitMeter. The data can be easily accessed using the `data()` function:

```r
data(banana)
str(banana)
# > Classes 'data.table' and 'data.frame': 1650 obs. of  5
# >variables:
# >  $ i           : num  1 1 1 1 1 1 1 1 1 1 ...
# >  $ t           : num  1 2 3 4 5 6 7 8 9 10 ...
# >  $ j           : num  1 1 1 1 1 1 1 1 1 1 ...
# >  $ bioimpedance: num  17256 18705 21563 23041 23511 ...
# >  $ weight      : int  213 208 203 196 192 187 183 178 ...
# >  - attr(*, ".internal.selfref")=<externalptr>
```

In this framework, i denotes the statistical unit (i.e., a banana from the production batch), t represents the day of observation, and j indicates the electrical frequency at which bioimpedance was measured. Specifically, the frequency levels correspond to 20 Hz for $j = 1$, 100 Hz for $j = 2$, and 500 Hz for $j = 3$.

[1] https://sensingtechnologies.groups.unibz.it/

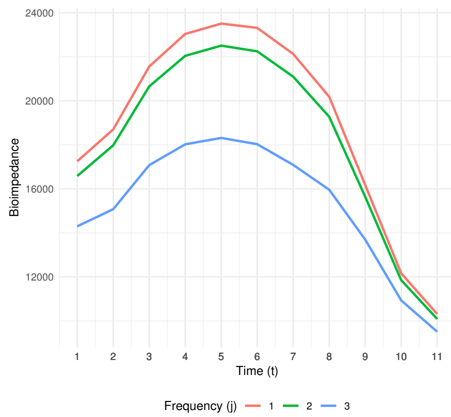

Figure 1: Time series of bioimpedance, banana $i = 1$.

The bioimpedance time series for a given frequency level j exhibit a consistent pattern across statistical units: an initial increasing phase followed by a decreasing trend, indicating the presence of a regime switch. This pattern is clearly revealed through a graphical representation of the time series, which can be produced using the graphical tools available in R, such as:

```
library(ggplot2)

# Select a banana (e.g. i = 1)

banana_i <- banana[which(banana$i == 1),]

# Time series of bioimpedance for the selected banana (Fig.1)

ggplot(banana_i, aes(x = t, y = bioimpedance,
    colour = factor(j))) + geom_line(size = 1) +
    scale_x_continuous(breaks = banana_i$t) +
    labs( x = "Time (t)",   y = "Bioimpedance",
        colour = "Frequency (j)") +
    theme_minimal() + theme(legend.position = "bottom") +
    guides(colour = guide_legend(nrow = 1))
```

Fig. 1 displays the bioimpedance time series for a randomly selected banana ($i = 1$), across the three frequency levels j. A comparable pattern is observed across all bananas in the dataset.

Focusing on a single level of j, a simple model to investigate the bioimpedance is:

$$y_{it} = \phi_1 \mathbb{1}\{y_{it-1} \leq \gamma\} + \phi_2 \mathbb{1}\{y_{it-1} > \gamma\} + \varepsilon_{it},$$

which can be specified and estimated as follows:

```
# Select a j value
```

```
data_j <- banana[which(banana$j==1),]

# Prepare input matrix Y for ptm2

bioimpedance <- matrix(data_j$bioimpedance,
    ncol=length(unique(data_j$t)), byrow=TRUE)

# Estimation

res_m1 <- ptm2(Y=bioimpedance)
str(res_m1)

# > Formal class 'ptm2' [package "PanelTM"] with 5 slots
# >   ..@ threshold : Named num [1:2] 16717 0
# >   .. ..- attr(*, "names")= chr [1:2] "est.coef" "p.val"
# >   ..@ estimates : num [1, 1:2] 5167 0
# >   .. ..- attr(*, "dimnames")=List of 2
# >   .. .. ..$ : chr "delta.c"
# >   .. .. ..$ : chr [1:2] "est.coef" "p.val"
# >   ..@ cov.      : num [1:2, 1:2] 9736 745 745 2795
# >   .. ..- attr(*, "dimnames")=List of 2
# >   .. .. ..$ : chr [1:2] "delta.c" "threshold"
# >   .. .. ..$ : chr [1:2] "delta.c" "threshold"
# >   ..@ residuals.: num [1:45, 1:50] 2858 1479 1479 470 470 ...
# >   ..@ test.lin. :List of 3
# >   .. ..$ method: chr "parametric bootstrap-based linearity test"
# >   .. ..$ B     : num 1000
# >   .. ..$ p.val : num 1
```

The output of the `ptm2` function comprises five elements: `threshold`, `estimates`, `cov.`, `residuals.`, and `test.lin.`. The first two components provide the estimated values of the threshold parameter γ and of the model parameters, which in this case consist solely of δ_I. For each estimate, the associated p-value - computed using the corresponding z-score - is also reported. The `cov.` element returns the covariance matrix $\boldsymbol{G}_j$ as defined in Sect. 2.3, while `residuals` contains the first-differenced residuals $\Delta\epsilon_{it}$. Finally, the `test.lin.` slot provides the number of bootstrap replications and the resulting p-value for the threshold effect test described in Sect. 2.4. In our application, the output indicates that the bootstrap test supports the presence of a regime switch, which is estimated to occur at the threshold value $\hat{\gamma} = 16717$ of the lagged dependent variable y_{t-1}. Furthermore, both γ and δ_I are found to be statistically significant.

Additional variables may be incorporated into the model. For instance, the weight of the banana could be introduced as an endogenous regressor. In such a case, the corresponding model specification would be:

```
# Prepare input matrix X for ptm2

weight <- matrix(data_j$weight,
```

```
    ncol=length(unique(data_j$t)), byrow=TRUE)

# Estimation

res_m2 <- ptm2(Y=bioimpedance, Xendo=weight)

res_m2@threshold

# >     est.coef        p.val
# > 1.779751e+04 8.646353e-01

res_m2@estimates

# >             est.coef     p.val
# > beta.X1     95.03142 0.9400501
# > delta.c   5426.32248 0.9201690
# > delta.X1    23.24755 0.8270913

res_m2@test.lin.

# > $method
# > [1] "parametric bootstrap-based linearity test"
# > $B
# > [1] 1000
# > $p.val
# > [1] 1
```

In this case, the test supports the presence of a threshold effect, and the estimated value $\hat{\gamma} = 17797.51$ aligns with the estimate obtained under the previous model specification. However, both the threshold and the coefficients associated with the regressor and the constant are not found to be statistically significant.

Once the threshold $\hat{\gamma}$ has been estimated, this information can be used to identify the time at which a regime switch occurs in the time series of each statistical unit using the `cpoint` function:

```
# Apply cpoint to the two-way panel using the estimated
# threshold from the model with a constant only (res_m1)

cp_m1 <- cpoint(data.=data_j, nameI="i",
    nameT="t", nameJ="j", nameY="bioimpedance",
    thresholds=res_m1@"threshold"[[1]])

str(cp_m1)

# > Formal class 'cpoint' [package "PanelTM"] with 1 slot
# >   ..@ CP:'data.frame':  50 obs. of  2 variables:
# >   .. ..$ i    : num [1:50] 1 2 3 4 5 6 7 8 9 10 ...
# >   .. ..$ CP_j1: num [1:50] 8 8 8 9 8 8 7 7 8 9 ...

summary(cp_m1@CP[,2])
```

```
# > Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
# >   6.000   7.000   8.000   7.915   8.500   9.000       3
```

It is important to note that the `cpoint` function returns `NA` values when the statistical unit does not exhibit any observation above (or below) the estimated threshold $\hat{\gamma}$, or in cases where, due to the presence of multiple regime switches, a unique time point cannot be identified after which the threshold variable remains consistently above (or below) $\hat{\gamma}$ for a longer duration. In our example, we find that the change point is on average at time $t = 8$.

As an alternative to analysing each of the three electrical frequencies separately, the `ptm3` function allows for joint estimation across all frequency levels within a unified model specification. The following application illustrates the use of `ptm3` to estimate a model with a constant as the sole regressor and for a model with an endogenous regressor. The output and results are consistent with those obtained using the `ptm2` function for individual frequency levels. In the example below, the `test.lin` option is disabled to reduce computational time; however, this option remains available and, when enabled, performs the linearity test independently for each value of j.

```
# Model with a constant as the only regressor

res_m3 <- ptm3(data.=banana, nameI="i", nameT="t", nameJ="j",
    nameY="bioimpedance", test.lin=FALSE)

str(res_m3)

# > Formal class 'ptm3' [package "PanelTM"] with 5 slots
# >  ..@ threshold : num [1:3, 1:3] 1 2 3 16717 16845 ...
# >  .. ..- attr(*, "dimnames")=List of 2
# >  .. .. ..$ : NULL
# >  .. .. ..$ : chr [1:3] "j" "coef" "p.val"
# >  ..@ param     : num [1:3, 1:3] 1 2 3 5167 3848 ...
# >  .. ..- attr(*, "dimnames")=List of 2
# >  .. .. ..$ : NULL
# >  .. .. ..$ : chr [1:3] "j" "delta.c" "p.val"
# >  ..@ cov.      : num [1:2, 1:2, 1:3] 9736 745 745 2795 ..
# >  ..@ residuals.:List of 3
# >  .. ..$ : num [1:45, 1:50] 2858 1479 1479 470 470 ...
# >  .. ..$ : num [1:45, 1:50] 2675 1390 1390 464 464 ...
# >  .. ..$ : num [1:45, 1:50] -3801 946 946 290 290 ...
# >  ..@ test.lin. : list()

res_m3@threshold

# >      j     coef p.val
# > [1,] 1 16717.14     0
# > [2,] 2 16845.23     0
# > [3,] 3 15322.87     0
```

```
res_m3@param

# >      j  delta.c p.val
# > [1,] 1 5166.999     0
# > [2,] 2 3847.818     0
# > [3,] 3 5801.612     0

# Model with the weight of the banana as endogenous regressor

res_m4 <- ptm3(data.=banana, nameI="i", nameT="t", nameJ="j",
    nameY="bioimpedance", nameXendo="weight", test.lin=FALSE)
res_m4@threshold

# >      j     coef     p.val
# > [1,] 1 17797.51 0.8646353
# > [2,] 2 17762.85 0.8986385
# > [3,] 3 14463.02 0.7303314

res_m4@param

# >      j beta.X1.est.coef beta.X1.p.val delta.c.est.coef
# > [1,] 1         95.03142     0.9400501         5426.322
# > [2,] 2        123.50210     0.8953514        11375.184
# > [3,] 3        149.01122     0.4979287        16562.033
# >      delta.c.p.val delta.X1.est.coef delta.X1.p.val
# > [1,]     0.9201690          23.24755      0.8270913
# > [2,]     0.7105165         -25.18488      0.8997474
# > [3,]     0.3275378         -71.82204      0.5628072
```

Finally, as with the `ptm2` function, results from the `ptm3` function can be used for applying the `cpoint` function and identifying the time of regime switch in the analysed time series:

```
# Apply cpoint to the three-way panel using the estimated
# threshold from the model with a constant only (res_m3)

cp_m3 <- cpoint(data.=banana, nameI="i", nameT="t",
    nameJ="j", nameY="bioimpedance",
    thresholds=res_m3@"threshold"[,2])

str(cp_m3)

# > l class 'cpoint' [package "PanelTM"] with 1 slot
# >   ..@ CP:'data.frame':  50 obs. of  4 variables:
# >   .. ..$ i    : num [1:50] 1 2 3 4 5 6 7 8 9 10 ...
# >   .. ..$ CP_j1: num [1:50] 8 8 8 9 8 8 7 7 8 9 ...
# >   .. ..$ CP_j2: num [1:50] 8 7 8 8 8 8 7 7 8 8 ...
# >   .. ..$ CP_j3: num [1:50] 8 7 7 8 8 8 6 7 8 8 ...
```

```
summary(cp_m3@CP[,2:4])

# > CP_j1           CP_j2           CP_j3
# >  Min.   :6.000   Min.   :6.000   Min.   :3.000
# >  1st Qu.:7.000   1st Qu.:7.000   1st Qu.:7.000
# >  Median :8.000   Median :8.000   Median :7.000
# >  Mean   :7.915   Mean   :7.646   Mean   :7.261
# >  3rd Qu.:8.500   3rd Qu.:8.000   3rd Qu.:8.000
# >  Max.   :9.000   Max.   :9.000   Max.   :9.000
# >  NA's   :3       NA's   :2       NA's   :4
```

4.2 Data random generation and their analysis

The `PanelTM` package also provides functionality for simulating data from two- or three-way panel structures through the `simptm` function. For instance, suppose to generate 50 datasets, each consisting of 20 individuals observed over 15 time periods, according to the following two-way panel model:

$$y_{it} = (-0.7 - 0.5x_{it})\mathbb{1}(y_{it-1} \le 0) + (1.8 + 0.8x_{it})\mathbb{1}(y_{it-1} > 0) + \varepsilon_{it},$$

where the regime switch occurs at time $t = 12$, x_{it} follows an autoregressive process with parameter 0.7, and ε_{it} is a random error with unit variance. This data can be simulated using the following lines of code:

```
sims <- simptm(n=20, T.=15, J=1, CP=12, gamma=c(0),
    phi_c=matrix(c(-0.7,1.8), nrow=1, byrow=TRUE),
    phi_X=matrix(c(-0.5,0.8), nrow=1, byrow=TRUE),
    sigmau=1, parAR=0.2, B=50, seedstart=1)

str(sims)

# > Formal class 'simptm' [package "PanelTM"] with 1 slot
# >   ..@ simulation:List of 2
# >  .. ..$ Data matrix for B=1:'data.frame': 300 obs. of
# > 5 variables:
# >  .. .. ..$ i : int [1:300] 1 1 1 1 1 1 1 1 1 1 ...
# >  .. .. ..$ t : int [1:300] 1 2 3 4 5 6 7 8 9 10 ...
# >  .. .. ..$ j : int [1:300] 1 1 1 1 1 1 1 1 1 1 ...
# >  .. .. ..$ Y : num [1:300] -2.676 -0.569 -1.252 ...
# >  .. .. ..$ X1: num [1:300, 1] 1.5076 -0.0283 ...
# >  .. ..$ Data matrix for B=2:'data.frame': 300 obs. of
# > 5 variables:
# >  .. .. ..$ i : int [1:300] 1 1 1 1 1 1 1 1 1 1 ...
# >  .. .. ..$ t : int [1:300] 1 2 3 4 5 6 7 8 9 10 ...
# >  .. .. ..$ j : int [1:300] 1 1 1 1 1 1 1 1 1 1 ...
# >  .. .. ..$ Y : num [1:300] -0.8 -0.778 -0.467  ...
# >  .. .. ..$ X1: num [1:300, 1] -1.455 -0.349 0.179 ...

[...]
```

Notice that the argument `sigmau` stands for σ in Eq. (7). The simulated data can be fitted using either the `ptm2` or `ptm3` function. In fact, estimating a `ptm3` model where the column j contains a single unique value is equivalent to fitting a `ptm2` model. To illustrate this, we use the `ptm3` function to estimate the following model on each of the two simulated datasets:

$$y_{it} = (1, \boldsymbol{x}'_{it})\boldsymbol{\phi}_1 \mathbb{1}\{y_{it-1} \leq \gamma\} + (1, \boldsymbol{x}'_{it})\boldsymbol{\phi}_2 \mathbb{1}\{y_{it-1} > \gamma\} + \varepsilon_{it},$$

```
# Estimate the two simulated two-way panel threshold models
# on the first simulated dataset

estimates1 <- ptm3(sims@simulation[[1]], nameI="i",
    nameT="t", nameJ="j", nameY="Y", nameXexo="X1",
    test.lin=FALSE)

str(estimates1)

# > Formal class 'ptm3' [package "PanelTM"] with 5 slots
# >   ..@ threshold : num [1, 1:3] 1 -0.172 0.893
# >   .. ..- attr(*, "dimnames")=List of 2
# >   .. .. ..$ : NULL
# >   .. .. ..$ : chr [1:3] "j" "coef" "p.val"
# >   ..@ param     : num [1, 1:7] 1 -1.3895 0.0679 0.791 ...
# >   .. ..- attr(*, "dimnames")=List of 2
# >   .. .. ..$ : NULL
# >   .. .. ..$ : chr [1:7] "j" "beta.X1.est.coef" ...
# >   ..@ cov.      : num [1:4, 1:4, 1] 0.5794 -0.8876 ...
# >   ..@ residuals.:List of 1
# >   .. ..$ : num [1:117, 1:20] 0.522 0.522 0.522 -1.371 ...
# >   ..@ test.lin. : list()
```

To expedite computation across all datasets, the `lapply()` function can be used to apply `ptm3` to each simulated dataset contained in the output list of `sims`:

```
# Estimation

estimates_list <- lapply(sims@simulation,
    function(data){
        ptm3(data, nameI = "i", nameT = "t", nameJ = "j",
        nameY = "Y", nameXexo = "X1", test.lin= FALSE)
        }
    )

# Check the results (e.g., on the first dataset)

str(estimates_list[[1]])

# > Formal class 'ptm3' [package "PanelTM"] with 5 slots
# >    ..@ threshold : num [1, 1:3] 1 -0.172 0.893
```

```
#>   .. ..- attr(*, "dimnames")=List of 2
#>   .. .. ..$ : NULL
#>   .. .. ..$ : chr [1:3] "j" "coef" "p.val"
#>   ..@ param     : num [1, 1:7] 1 -1.3895 0.0679 0.791 ...
#>   .. ..- attr(*, "dimnames")=List of 2
#>   .. .. ..$ : NULL
#>   .. .. ..$ : chr [1:7] "j" "beta.X1.est.coef" ...
#>   ..@ cov.      : num [1:4, 1:4, 1] 0.5794 -0.8876 ...
#>   ..@ residuals.:List of 1
#>   .. ..$ : num [1:117, 1:20] 0.522 0.522 0.522 -1.371 ...
#>   ..@ test.lin. : list()
```

The performance of the model on the simulated datasets can now be evaluated using the `perfm` function, which computes the bias (`nrb`), relative bias (`rb`), root mean squared error (`rmse`), and relative root mean squared error (`rrmse`). The following code lines illustrate the application of this function to the three three-differenced parameters of the model estimated on the 50 simulated dataset, namely: β, δ_I, and δ_X.

```
# Collect the 50 estimated values of beta (the coefficient of
# the exogenous regressor in the lower regime) into a vector

betaX1_estimates <- sapply(estimates_list,
    function(est) est@param[,2])

# Compute performance measures

pm1 <- perfm(truepar = -0.5, estpar = betaX1_estimates)

pm1

#> An object of class "perfm"
#> Slot "trueval":
#> [1] -0.5

#> Slot "rb":
#> [1] -0.1494372

#> Slot "nrb":
#> [1] 0.07471858

#> Slot "rrmse":
#> [1] 2.085514

#> Slot "rmse":
#> [1] 1.042757

# Collect the 50 estimated values of delta.c (the difference
# between the constants of the model) into a vector
```

```
delta_c_estimates <- sapply(estimates_list,
    function(est) est@param[,4])

# Compute performance measures

pm2 <- perfm(truepar = 1.8 + 0.7, estpar = delta_c_estimates)

pm2

# > An object of class "perfm"
# > Slot "trueval":
# > [1] 2.5

# > Slot "rb":
# > [1] -1.801049

# > Slot "nrb":
# > [1] -4.502623

# > Slot "rrmse":
# > [1] 1.944078

# > Slot "rmse":
# > [1] 4.860196

# Collect the 50 estimated values of delta.X1 (the difference
# between the exogenous regressor coefficients) into a vector

delta_X1_estimates <- sapply(estimates_list,
    function(est) est@param[,6])

# Compute performance measures

pm3 <- perfm(truepar = 0.8 + 0.5,
    estpar = delta_X1_estimates)
pm3

# > An object of class "perfm"
# > Slot "trueval":
# > [1] 1.3

# > Slot "rb":
# > [1] -0.3119362

# > Slot "nrb":
# > [1] -0.4055171

# > Slot "rrmse":
# > [1] 1.18448
```

```
# > Slot "rmse":
# > [1] 1.539824
```

5 Summary and discussion

This paper introduces PanelTM, an R package designed to estimate dynamic panel threshold models with two- or three-way structures. Built upon the generalized method of moments estimation approach of Seo and Shin (2016) and its adaptation to three-way panels by Di Lascio and Perazzini (2024), PanelTM offers a substantial contribution to the toolbox available for modern panel data analysis. In particular, it provides a flexible and accessible framework capable of handling lagged dependent variables as threshold variables, endogenous and exogenous covariates, and allowing for threshold heterogeneity across a third dimension. Beyond model estimation, PanelTM also offers functionalities for detecting regime switches via the cpoint function, simulating panel threshold data through the simptm function, and evaluating estimator performance using the perfm function. The package thus provides a comprehensive environment not only for applied researchers aiming to detect structural breaks in panel settings, but also for methodologists interested in conducting simulation studies or exploring theoretical properties of threshold estimators. Its implementation in an open-source software environment further enhances accessibility and reproducibility, offering an important tool for researchers in econometrics, biostatistics, and other fields dealing with dynamic, regime-switching processes in panel data.

Acknowledgments

The authors acknowledge the financial support from the Free University of Bozen-Bolzano via the research project "New directions in statistical methods for bio-impedance analysis of fruit ripeness (BIOFRUIT - ID 2020)" and the Italian Ministry of University and Research (MUR) under the Research Project of National Interest (PRIN 2022) (grant 20223CEZSR) funded by European Union – Next Generation EU.

References

Arellano, M. and Bond, S. (1991). Some tests of specification for panel data: Monte carlo evidence and an application to employment equations. *The Review of Economic Studies*, 58:277–297.

Balazsi, L., Matyas, L., and Wansbeek, T. (2018). The estimation of multidimensional fixed effects panel data models. *Econometric Reviews*, 37(3):212–227.

Bates, D., Mächler, M., Bolker, B., and Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48.

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

Chan, J., Horvàth, L., and Huškovà, M. (2013). Darling–erdòs limit results for change-point detection in panel data. *Journal of Statistical Planning and Inference*, 143:955–970.

Cho, H. (2016). Change-point detection in panel data via double cusum statistic. *Electronic Journal of Statistics*, 10:2000–2038.

Croissant, Y. and Millo, G. (2008). Panel data econometrics in r: The plm package. *Journal of Statistical Software*, 27(2):1–43.

Cubranic, D., Pickup, M., Gustafson, P., Evans, G., and Biljana, J. S. (2022). *OrthoPanels: Dynamic Panel Models with Orthogonal Reparameterization of Fixed Effects*. R package version 1.2-4.

Di Lascio, F. and Perazzini, S. (2022). Change point detection in fruit bioimpedance using a three-way panel model. In A. Balzanella, M. Bini, C. C. and Verde, R., editors, *Book of short papers - SIS2022*, volume 159, pages 1184–1189. Pearson.

Di Lascio, F. and Perazzini, S. (2024). Insights into bioimpedance analyser via three-way dynamic panel threshold regression modelling. Technical Report 104, Faculty of Economics and Management at the Free University of Bozen-Bolzano.

Fritsch, M., Pua, A. A. Y., and Schnurbus, J. (2021). pdynmc: A package for estimating linear dynamic panel data models based on nonlinear moment conditions. *The R Journal*, 13(1):218–231.

Gong, W. and Seo, M. (2024). Bootstraps for dynamic panel threshold models. *arXiv*, pages 1–52.

Hansen, B. E. (1999). Threshold effects in non-dynamic panels: Estimation, testing, and inference. *Journal of Econometrics*, 93(2):345–368.

Hansen, B. E. (2000). Sample splitting and threshold estimation. *Econometrica*, 68(3):575–603.

Horvàth, L. and Huškovà, M. (2012). Change-point detection in panel data. *Journal of Time Series Analysis*, 33(4):631–648.

Hsiao, C. and Zhang, J. (2015). Iv,gmmor likelihood approach to estimate dynamic panel models when either n or t or both are large. *Jorunal of Econometrics*, 187:312–322.

Lancaster, T. (2002). Orthogonal parameters and panel data. *Review of Economic Studies*, 69:647–66.

Long, J. A. (2020). *panelr: Regression Models and Utilities for Repeated Measures and Panel Data*. R package version 0.7.3.

Mátyás, L. (1997). Proper econometric specification of the gravity model. *The World Economy*, 20:363–368.

Mátyás, L., editor (2017). *The Econometrics of Multi-dimensional Panels*, volume 50 of *Advanced Studies in Theoretical and Applied Econometrics*. Springer.

Seo, M., Kim, S., and Kim, Y.-J. (2019). Estimation of dynamic panel threshold model using stata. *The Stata Journal: Promoting communications on statistics and Stata*, 19:685–697.

Seo, M. and Shin, Y. (2016). Dynamic panels with threshold effect and endogeneity. *Journal of Econometrics*, 195(2):169–186.

Tong, H. (1990). Non-linear time series: A dynamical system approach. *Oxford Statistical Science Series*, 6.

Tsung-wu, H. (2024). *pdR: Threshold Model and Unit Root Tests in Cross-Section and Time Series Data*. R package version 1.9.3.