

**R-Package “pdynmc”:  
GMM Estimation of Dynamic Panel Data Models  
Based on Nonlinear Moment Conditions**

Markus Fritsch, Andrew Pua, Joachim Schnurbus

University of Passau & Xiamen University

June 11, 2021

# Outline

- 1 Introduction
- 2 GMM estimation
- 3 A short note on nonlinear moment conditions
- 4 Package `pdynmc` — a first tour
- 5 Conclusion

# What is pdynmc?

**pdynmc** → panel data

**pdynmc** → (linear) dynamic models  $\Rightarrow$  GMM

**pdynmc** → (linear and/or) nonlinear moment conditions (w.r.t.  $\alpha_j, \beta_k$ )

pdynmc is intended to efficiently estimate models like

$$\begin{aligned} y_{i,t} &= \alpha_1 y_{i,t-1} + \dots + \alpha_p y_{i,t-p} \\ &+ \beta_1 x_{i,t^*,1}^* + \dots + \beta_q x_{i,t^*,q}^* + \underbrace{\eta_i + \varepsilon_{i,t}}_{u_{i,t}} \end{aligned}$$

where

$x^*$  means that we allow for endogenous, predetermined, and/or exogenous covariates (could also be time/etc. dummies), and

$t^*$  means that arbitrary lags of the covariates can be included.

# Key features of pdynmc (and conclusion)

pdynmc allows for GMM estimation of linear dynamic panel data models based on linear and/or **nonlinear moment conditions** and provides the following features:

- R-package  $\Rightarrow$  **open source**.
- **Comprehensive control** over all configuration/specification decisions.
- Can handle **arbitrary unbalancedness** (given moment conditions can be derived).
- **State-of-the-art estimation** (iterated GMM, Hansen & Lee, 2020) of linear dynamic panel data models.
- Specification tests and analysis of stability of coefficient estimates.
- Panel structure analysis (visualizations and figures).

# GMM estimation, moment conditions, assumptions

GMM estimation is performed by minimizing the objective function

$$L = \overline{\mathbf{m}}' \cdot \mathbf{W} \cdot \overline{\mathbf{m}}$$

where

$\overline{\mathbf{m}}$  is the sample analogon to the population moment conditions  $E(\cdot)$ ,

$\mathbf{W}$  is the (moment condition) weighting matrix.

The moment conditions are derived from different (sets of) assumptions.

# Sets of assumptions

**A1** (Ahn & Schmidt, 1995):

The data are independently distributed across  $i$ ,

$$E(\eta_i) = 0, \quad i = 1, \dots, n,$$

$$E(\varepsilon_{i,t}) = 0, \quad i = 1, \dots, n, \quad t = 2, \dots, T,$$

$$E(\varepsilon_{i,t} \cdot \eta_i) = 0, \quad i = 1, \dots, n, \quad t = 2, \dots, T,$$

$$E(\varepsilon_{i,t} \cdot \varepsilon_{i,s}) = 0, \quad i = 1, \dots, n, \quad t \neq s,$$

$$E(y_{i,1} \cdot \varepsilon_{i,t}) = 0, \quad i = 1, \dots, n, \quad t = 2, \dots, T,$$

$$n \rightarrow \infty, \text{ while } T \text{ is fixed, such that } \frac{T}{n} \rightarrow 0.$$

**A2** (Arellano, 2003; Kiviet, 2007; Bun & Sarafidis, 2015):

$$E(\Delta y_{i,t} \cdot \eta_i) = 0, \quad i = 1, \dots, n.$$

# Moment conditions are derived w.r.t.

## Equation in levels

$$\begin{aligned} y_{i,t} &= \alpha_1 y_{i,t-1} + \dots + \alpha_p y_{i,t-p} \\ &+ \beta_1 x_{i,t^*,1}^* + \dots + \beta_q x_{i,t^*,q}^* + \underbrace{\eta_i + \varepsilon_{i,t}}_{u_{i,t}} \end{aligned}$$

## Equation in (first) differences

$$\begin{aligned} \Delta y_{i,t} &= \alpha_1 \Delta y_{i,t-1} + \dots + \alpha_p \Delta y_{i,t-p} \\ &+ \beta_1 \Delta x_{i,t^*,1}^* + \dots + \beta_q \Delta x_{i,t^*,q}^* + \Delta \varepsilon_{i,t} \end{aligned}$$

# Standard moment conditions

## under A1

**Linear** moment conditions w.r.t. **equation in differences**

$$E(y_{i,s} \cdot \Delta u_{i,t}) = 0, \quad t = 3, \dots, T; \quad s = 1, \dots, t-2. \quad (\text{MYD})$$

**Nonlinear** moment conditions

$$E(u_{i,t} \cdot \Delta u_{i,t-1}) = 0, \quad t = 4, \dots, T. \quad (\text{MN})$$

$$E(u_{i,T} \cdot \Delta u_{i,t-1}) = 0, \quad t = 4, \dots, T. \quad (\text{MNAS})$$

## under A1 & A2

**Linear** moment conditions w.r.t. **equation in levels**

$$E(\Delta y_{i,t-1} \cdot u_{i,t}) = 0, \quad t = 3, \dots, T. \quad (\text{MYL})$$



# Moment conditions from covariates

## Linear moment conditions w.r.t. equation in differences

$$E \left( \sum_{t=2}^T \Delta x_{it} \Delta u_{it} \right) = 0 \quad \text{for exogenous } x. \quad (\text{MFCD})$$

$$\text{Alternatively } E(x_{i,s} \cdot \Delta u_{i,t}) = 0, \quad t = 3, \dots, T, \quad (\text{MXD})$$

where  $s = 1, \dots, t - 2$ , for endogenous  $x$ ,

$s = 1, \dots, t - 1$ , for predetermined  $x$ ,

$s = 1, \dots, T$ , for strictly exogenous  $x$ .

## Linear moment conditions w.r.t. equation in levels

$$E \left( \sum_{t=1}^T x_{it} u_{it} \right) = 0 \quad \text{for exogenous } x. \quad (\text{MFCL})$$

$$\text{Alternatively } E(\Delta x_{i,v} \cdot u_{i,t}) = 0, \quad (\text{MXL})$$

where  $v = t - 1$ ;  $t = 3, \dots, T$ , for endogenous  $x$ ,

$v = t$ ;  $t = 2, \dots, T$ , otherwise.

**Note:** MXD/MXL require analogous assumptions to A1 and/or A2 w.r.t.  $x$ .

# Why we should care about nonlinear moment conditions

When the lag parameter is close to one, ...

... linear moment conditions derived from A1 fail to identify the lag parameter.

... additional linear moment conditions derived from A2

- provide a remedy, but:
- A2 may be suspect in many contexts (e.g., Arellano's worker example).

... nonlinear moment conditions from A1 can

- identify the lag parameter  $\Rightarrow$  estimate consistently.
- serve as robustness check  $\Rightarrow$  A2 valid?

# Installing and loading package

```
### Install CRAN-Version
```

```
install.packages("pdynmc")
```

```
### Install most recent version from Github
```

```
install.packages("devtools")
```

```
library(devtools)
```

```
install_github("markusfritsch/pdynmc")
```

```
### Load installed package
```

```
library(pdynmc)
```

Note: Copy & paste the code to R should work.

# Load and adjust example data set

## **Employment and Wages in the United Kingdom**

(Arellano & Bond, 1991)

```
data(EmplUK, package = "plm")
dat <- EmplUK
dat[,c(4:7)] <- log(dat[,c(4:7)])
names(dat)[4:7] <- c("n", "w", "k", "ys")
```

# Function `data.info`

```
data.info(  
  dat,  
  i.name = "firm",  
  t.name = "year"  
)
```

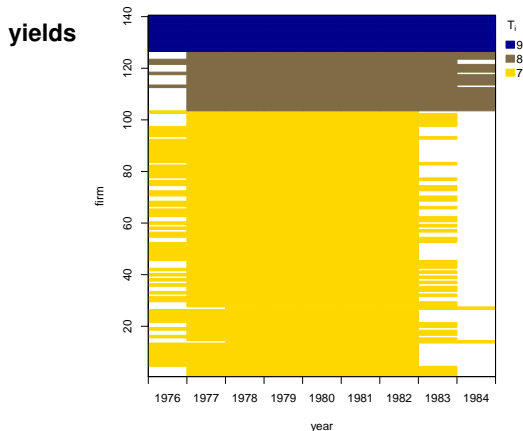
## **yields**

Unbalanced panel data set with 1031 rows and the following time period frequencies:

1976	1977	1978	1979	1980	1981	1982	1983	1984
80	138	140	140	140	140	140	78	35

# Function strucUPD.plot

```
strucUPD.plot(  
  dat,  
  i.name = "firm",  
  t.name = "year"  
)
```



# Function `pdynmc`

```
reg <- pdynmc(  
  dat = dat, varname.i = "firm", varname.t = "year",  
  
  use.mc.diff = TRUE, use.mc.lev = FALSE, use.mc.nonlin = TRUE,  
  
  include.y = TRUE,  
  varname.y = "n", lagTerms.y = 2,  
  
  fur.con = TRUE,  
  fur.con.diff = TRUE, fur.con.lev = TRUE,  
  varname.reg.fur = c("w", "k", "ys"),  
  lagTerms.reg.fur = c(1,2,2),  
  
  include.dum = TRUE,  
  dum.diff = TRUE, dum.lev = FALSE,  
  varname.dum = "year",  
  
  w.mat = "iid.err", std.err = "corrected",  
  estimation = "iterative",  
  # max.iter = 4,  
  opt.meth = "BFGS"  
)  
  
summary(reg)
```

**yields ...**

## Model output for object `reg` (excerpt)

Dynamic linear panel estimation (iterative)

Estimation steps: 13

Coefficients:

	Estimate	Std.Err.rob	z-value.rob	Pr(> z.rob )	
L1.n	1.19704	0.06855	17.463	< 2e-16	***
L2.n	-0.12589	0.06799	-1.852	0.06403	.
L0.w	-0.21935	0.12697	-1.728	0.08399	.
L1.w	0.25791	0.13753	1.875	0.06079	.
L0.k	0.25521	0.05568	4.583	< 2e-16	***
L1.k	-0.15546	0.07673	-2.026	0.04276	*
L2.k	-0.15599	0.05498	-2.837	0.00455	**
L0.ys	0.53006	0.18336	2.891	0.00384	**
L1.ys	-0.37925	0.22256	-1.704	0.08838	.
L2.ys	-0.20770	0.15186	-1.368	0.17131	.
1979	0.03124	0.01015	3.077	0.00209	**

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

53 total instruments are employed to estimate 16 parameters

27 linear (DIF) 4 nonlinear

8 further controls (DIF) 8 further controls (LEV)

6 time dummies (DIF)

J-Test (overid restrictions): 48.1 with 37 DF, pvalue: 0.1046

F-Statistic (slope coeff): 92232.95 with 10 DF, pvalue: <0.001

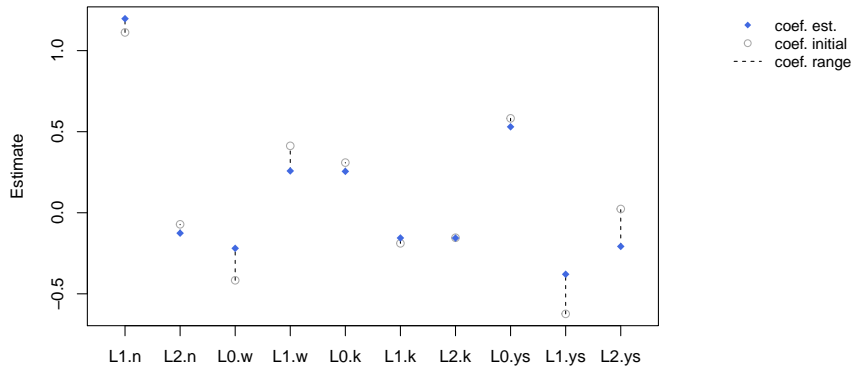
F-Statistic (time dummies): 20.63 with 6 DF, pvalue: 0.0021



# Coefficient range plot

```
plot(reg, type = "coef.range", omit1step = TRUE)
```

yields

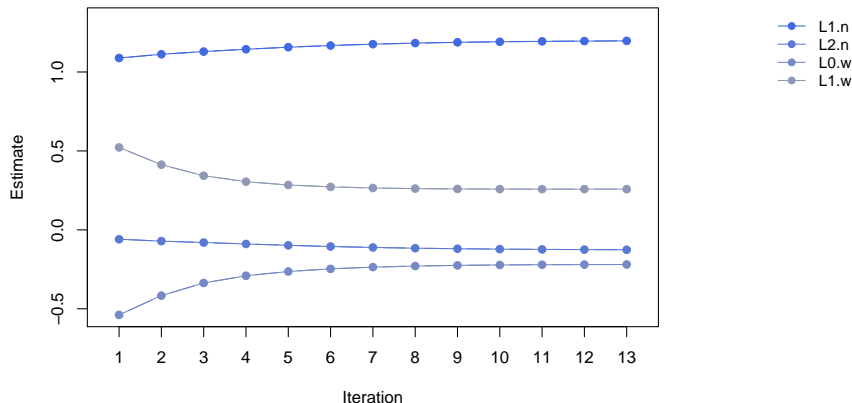


# Coefficient path plot (Hansen & Lee, 2020)

```
plot(reg, type = "coef.path", omit1step = TRUE,  
     co = c("L1.n", "L2.n", "L0.w", "L1.w")  
)
```

yields

Coefficient estimates over 13 iterations



# Arguments of function `pdynmc` (1)

R-command	Type of moment conditions
<b>use.mc.diff</b>	MYD/MFCD/MXD
<b>use.mc.lev</b>	MYL/MFCL/MXL
<b>use.mc.nonlin</b>	MN
<code>use.mc.nonlinAS</code>	MNAS

R-command	Estimate parameter(s)	Derive moment condition(s)
<b>include.y</b>	+	MYD/MYL
<code>fur.con/include.dum</code>	+	MFCD/MFCL
<code>include.x</code>	+	MXD/MXL
<code>include.x.instr</code>	-	MXD/MXL
<code>include.x.toInstr</code>	+	-

Note: Essential arguments are indicated in bold (**dat**, **varname.i**, **varname.t**).

## Arguments of function `pdynmc` (2)

- Relate to data set columns: `varname.reg.end`
- Restrict number of parameters: `lagTerms.reg.end`
- Restrict number of moment conditions: `maxLags.reg.end`

	<code>varname.</code>	<code>lagTerms.</code>	<code>maxLags.</code>
<code>.i</code>	+	-	-
<code>.t</code>	+	-	-
<code>.y</code>	+	+	+
<code>.reg.end</code>	+	+	+
<code>.reg.pre</code>	+	+	+
<code>.reg.ex</code>	+	+	+
<code>.reg.instr</code>	+	-	-
<code>.reg.toInstr</code>	+	-	-
<code>.reg.fur</code>	+	+	-
<code>.dum</code>	+	-	-

## Arguments of function `pdynmc` (3)

Context	R-command
Basic configuration	<code>w.mat</code> <code>std.err</code> <code>estimation</code>
Handle multicollinearity	<code>col_tol</code> <code>inst.thresh</code>
Stata-conformity	<code>inst.stata</code> <code>w.mat.stata</code>
Iterated estimation	<code>max.iter</code> <code>iter.tol</code>
Nonlinear optimization	<code>opt.method</code> <code>hessian</code> <code>optCtrl</code>
Starting values	<code>custom.start.val</code> <code>start.val</code> <code>start.val.lo</code> <code>start.val.hi</code> <code>seed.input</code>

# References

- Ahn, S. C. & P. Schmidt (1995), Efficient estimation of models for dynamic panel data. *Journal of Econometrics*, **68**(1), 5–27.
- M. Arellano (2003), *Panel Data Econometrics*, Oxford University Press.
- Arellano, M. & S. Bond (1991), Some Tests of Specification for Panel Data: Monte Carlo Evidence and an Application to Employment Equations. *Review of Economic Studies* **58**, 277–297.
- Bun, M. J. G. & V. Sarafidis (2015), Chapter 3 – Dynamic panel data models. In B. H. Baltagi, Editor, *The Oxford Handbook of Panel Data*, 76–110, Oxford University Press.
- Fritsch, M., Pua, A. & J. Schnurbus (2020), pdynmc - An package for estimating linear dynamic panel data models based on nonlinear moment conditions, *Working Paper*.
- Hansen, B. E. & S. Lee (2020), Inference for Iterated GMM Under Misspecification, *Econometrica*, *forthcoming*.
- J. F. Kiviet (2007), Chapter 11 – Judging contending estimators by simulation: Tournaments in dynamic panel data models. In Phillips, G. D. A. & E. Tzavalis, Editors, *The Refinement of Econometric Estimation and Test Procedures: Finite Sample and Asymptotic Analysis*, 282–318. Cambridge University Press.
- R Core Team (2020) R: A language and environment for statistical computing. *R Foundation for Statistical Computing*. Vienna, Austria.